Reinforcement Learning with Action-Free Pre-Training from Videos

Younggyo Seo [1 2] **Kimin Lee** [3 4] **Stephen James** [3] **Pieter Abbeel** [3]

Abstract

Recent unsupervised pre-training methods have shown to be effective on language and vision domains by learning useful representations for multiple downstream tasks. In this paper, we investigate if such unsupervised pre-training methods can also be effective for vision-based reinforcement learning (RL). To this end, we introduce a framework that learns representations useful for understanding the dynamics via generative pre-training on videos. Our framework consists of two phases: we pre-train an action-free latent video prediction model, and then utilize the pre-trained representations for efficiently learning action-conditional world models on unseen environments. To incorporate additional action inputs during fine-tuning, we introduce a new architecture that stacks an action-conditional latent prediction model on top of the pre-trained action-free prediction model. Moreover, for better exploration, we propose a video-based intrinsic bonus that leverages pre-trained representations. We demonstrate that our framework significantly improves both final performances and sample-efficiency of vision-based RL in a variety of manipulation and locomotion tasks. Code is available at `https://github.com/younggyoseo/apv`.

1. Introduction

Deep reinforcement learning (RL) has made significant advance in solving various sequential decision-making problems (Mnih et al., 2015; Levine et al., 2016; Silver et al., 2017; Vinyals et al., 2019; Berner et al., 2019; Akkaya et al., 2019; Kalashnikov et al., 2021). However, existing RL methods often start learning *tabula rasa* without any prior knowledge of the world, therefore requiring a large amount of environment interaction for learning meaningful behaviors. By contrast, within the computer vision (CV) and natural language processing (NLP) domains, recent unsupervised pre-training approaches have shown to be effective by leveraging the pre-trained representations for fine-tuning in downstream tasks (Mikolov et al., 2013; Pennington et al., 2014; Noroozi & Favaro, 2016; Gidaris et al., 2018; Devlin et al., 2019; Radford et al., 2018; He et al., 2020).

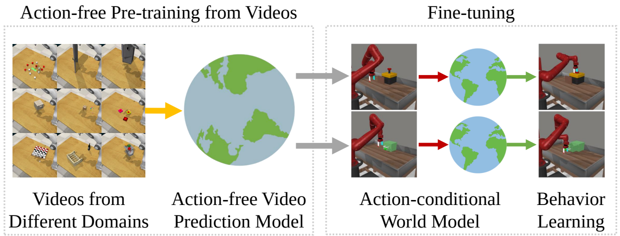

Figure 1. We pre-train an action-free latent video prediction model using videos from different domains (left), and then fine-tune the pre-trained model on target domains (right).

Recent works have shown promise in adopting such *pre-training and fine-tuning* paradigm to vision-based RL, by demonstrating that representations pre-trained with various unsupervised representation learning schemes can be effective in downstream tasks (Finn et al., 2016b; Dwibedi et al., 2018; Zhan et al., 2020; Laskin et al., 2020b; Stooke et al., 2021; Schwarzer et al., 2021b). Notably, Stooke et al. (2021) show that weight initialization with contrastively pre-trained representations leads to performance improvement. These works, however, mostly focus on the setup where pre-training datasets are collected in the target domains, or in different domains but with very similar visuals. Instead, we would like to leverage videos from diverse domains for pre-training, and transfer the pre-trained representations for solving newly encountered domains.

In this paper, we present APV: **A**ction-Free **P**re-training from **V**ideos, a novel framework that performs generative pre-training on videos for improving the sample-efficiency of vision-based RL. Since our goal is to learn the representations that can be transferred to various downstream tasks from readily available videos, our framework do not require the videos to be collected in the same domain of the downstream tasks, and also do not assume the datasets contain action information. Summarized in Figure 1, our framework comprises two phases: we first pre-train an action-free latent

---

[1] KAIST [2] Work done while visiting UC Berkeley [3] UC Berkeley [4] Now at Google Research. Correspondence to: Younggyo Seo <younggyo.seo@kaist.ac.kr>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

video prediction model to learn useful representations from videos, then fine-tune the pre-trained model for learning action-conditional world models on downstream tasks. To leverage the fine-tuned world models for behavior learning, we build APV on top of DreamerV2 (Hafner et al., 2021).

The key ingredients of APV are as follows:

- **Action-free pre-training from videos:** To capture rich dynamics information from diverse videos, we pre-train an action-free latent video prediction model. We find that the representations from the pre-trained model can be transferred to various downstream tasks.
- **Stacked latent prediction model** To incorporate additional action inputs during fine-tuning, we introduce a new architecture that stacks an action-conditional latent dynamics model on top of the action-free model.
- **Video-based intrinsic bonus** For better exploration, we propose an intrinsic bonus that utilizes video representations from the action-free model. Since the pre-trained representations contain information useful for understanding dynamics of environments, our intrinsic bonus effectively encourages agents to learn diverse behaviors.

In our experiments, we pre-train the action-free prediction model using 4950 videos collected on 99 manipulation tasks from RLBench (James et al., 2020) and fine-tune the pre-trained model on a range of manipulation tasks from Meta-world (Yu et al., 2020). Despite a big domain gap between RLBench and Meta-world, we demonstrate that APV significantly outperforms DreamerV2. For example, APV achieves the aggregate success rate of 95.4% on six manipulation tasks, while DreamerV2 achieves 67.9%. Moreover, we show that RLBench pre-trained representations can also be effective in learning locomotion tasks from DeepMind Control Suite (Tassa et al., 2020), where both the visuals and objectives significantly differ from RLBench videos.

2. Related Work

Unsupervised representation learning for CV and NLP. Recently, unsupervised representation learning methods have been actively studied in the domain of CV. Various representation learning methods, including reconstruction (He et al., 2021), rotation (Gidaris et al., 2018), solving zigsaw puzzles (Noroozi & Favaro, 2016), and contrastive learning (He et al., 2020; Chen et al., 2020), have reduced the gap with supervised pre-training with labels. In the domain of NLP, unsupervised representation learning has been successfully applied to training language models with generalization ability (Devlin et al., 2019; Radford et al., 2018; Yang et al., 2019). Notably, Devlin et al. (2019) and Radford et al. (2018) trained large transformer networks (Vaswani et al., 2017) with masked token prediction and generative pre-training, respectively, and showed that pre-trained models can be effectively fine-tuned on downstream tasks. In this work, we demonstrate that unsupervised pre-training can also be effective for vision-based RL.

Unsupervised representation learning for RL. Unsupervised representation learning for RL has also been studied to improve the sample-efficiency of RL algorithms. Notably, Jaderberg et al. (2017) showed that optimizing auxiliary unsupervised losses can improve the performance of RL agents. This has been followed by a series of works which demonstrated the effectiveness of various unsupervised learning objectives, including world-model learning (Hafner et al., 2019; 2021), reconstruction (Yarats et al., 2021c), future representation prediction (Gelada et al., 2019; Schwarzer et al., 2021a), bisimulation (Castro, 2020; Zhang et al., 2021), and contrastive learning (Oord et al., 2018; Anand et al., 2019; Mazoure et al., 2020; Srinivas et al., 2020). While these works optimize auxiliary unsupervised objectives to accelerate the training of RL agents, we instead aim to pre-train representations as in CV and NLP domains.

There have been several approaches to perform unsupervised pre-training for RL (Finn et al., 2016b; Dwibedi et al., 2018; Zhan et al., 2020; Srinivas et al., 2020; Stooke et al., 2021; Schwarzer et al., 2021b). In particular, Schwarzer et al. (2021b) proposed several self-supervised learning objectives that rely on actions, but assume access to action information of downstream tasks which may not be available in practice. The work closest to ours is Stooke et al. (2021), which demonstrated that the representations contrastively pre-trained without actions and rewards can be effective on unseen downstream tasks but with very similar visuals. In this work, we instead develop a framework that leverages action-free videos from diverse domains with different visuals and embodiments for pre-training. Concurrent to our work, Xiao et al. (2022) showed that pre-training a visual encoder on videos can be effective for downstream RL tasks. We instead pre-train a video prediction model instead of a visual encoder that operates on a single image.

Behavior learning with videos. Video datasets have also been utilized for behavior learning in various ways (Peng et al., 2018; Torabi et al., 2018; Aytar et al., 2018; Liu et al., 2018; Sermanet et al., 2018; Edwards et al., 2019; Schmeckpeper et al., 2020b;a; Chang et al., 2020; Chen et al., 2021; Zakka et al., 2022). Aytar et al. (2018) solved hard exploration tasks on Atari benchmark by designing an imitation reward based on YouTube videos, and Peng et al. (2018) proposed to learn physical skills from human demonstration videos by extracting reference motions and training an RL agent that imitates the extracted motions. Our work differs in that we utilize videos for pre-training representations, instead of learning behaviors from videos.

We provide more discussion on related fields in Appendix C.

3. Method

We formulate a vision-based control task as a partially observable Markov decision process (POMDP), which is defined as a tuple $(\mathcal{O}, \mathcal{A}, p, r, \gamma)$. Here, $\mathcal{O}$ is the high-dimensional observation space, $\mathcal{A}$ is the action space, $p(o_t | o_{<t}, a_{<t})$ is the transition dynamics, r is the reward function that maps previous observations and actions to a reward $r_t = r(o_{\leq t}, a_{<t})$, and $\gamma \in [0, 1)$ is the discount factor. The goal of RL is to learn an agent that behaves to maximize the expected sum of rewards $\mathbb{E}_p\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$.

3.1. Action-free Pre-training from Videos

In order to pre-train representations from action-free videos, we first learn a latent video prediction model, which is an action-free variant of a latent dynamics model (Hafner et al., 2019). Unlike autoregressive video prediction models that predict a next frame and utilize it as an input for the following prediction, the model instead operates on the latent space (Zhang et al., 2019; Hafner et al., 2019; Franceschi et al., 2020). Specifically, the model consists of three main components: (i) the representation model that encodes observations o_t to a model state z_t with Markovian transitions, (ii) the transition model that predicts future model states $\hat{z}_t$ without access to the observation, and (iii) the image decoder that reconstructs image observations $\hat{o}_t$. The model can be summarized as follow (see Figure 2):

$$\begin{aligned}
&\text{Representation model:} && z_t \sim q_\theta(z_t \mid z_{t-1}, o_t) \\
&\text{Transition model:} && \hat{z}_t \sim p_\theta(\hat{z}_t \mid z_{t-1}) \qquad (1) \\
&\text{Image decoder:} && \hat{o}_t \sim p_\theta(\hat{o}_t \mid z_t)
\end{aligned}$$

We train the model to reconstruct image observations, and to make the prediction from the representation model and transition model be close to each other. All model parameters θ are jointly optimized by minimizing the negative variational lower bound (ELBO; Kingma & Welling 2014):

$$\begin{aligned}
\mathcal{L}(\theta) \doteq \mathbb{E}_{q_\theta(z_{1:T} \mid o_{1:T})}\Big[\sum_{t=1}^{T} \Big(& \underbrace{-\ln p_\theta(o_t \mid z_t)}_{\text{image log loss}} \\
& + \underbrace{\beta_z \,\mathrm{KL}\left[q_\theta(z_t \mid z_{t-1}, o_t) \,\|\, p_\theta(\hat{z}_t \mid z_{t-1})\right]}_{\text{action-free KL loss}} \Big)\Big], \quad (2)
\end{aligned}$$

where β_z is a scale hyperparameter and T is the length of training sequences in a minibatch. Since the transition model does not condition on observations, it allows us to efficiently predict future states in the latent space without needing to predict future images using the image decoder at inference time. We implement the transition model as an action-free recurrent state-space model (RSSM; Hafner et al. 2019), which consists of both deterministic and stochastic components, and the representation model by combining the action-free RSSM with an image encoder. We refer to Appendix B for a more detailed formulation.

Figure 2. Illustration of action-free latent video prediction model. The model is trained to capture visual and dynamics information from action-free videos by reconstructing image observations. At inference time, the transition model is used to predict future states in the latent space without conditioning on predicted frames.

3.2. Stacked Latent Prediction Model

Once we pre-train the action-free prediction model, we fine-tune it into an action-conditional prediction model that can be used for solving various visual control tasks. Since actions and rewards, which provide more information about target tasks, are available during fine-tuning, it motivates incorporating them into the model. One naïve approach would be to initialize the action-conditional prediction model with the action-free model, and learn a reward predictor on top of it. But we find this fine-tuning scheme rapidly erases the useful knowledge in pre-trained models (see Figure 6(a) for supporting results). To effectively utilize the pre-trained representations, we introduce a new architecture that stacks an action-conditional prediction model on top of the action-free model as below (see Figure 3(a)):

$$\begin{aligned}
&\text{Action-free} \\
&\text{Representation model:} && z_t \sim q_\theta(z_t \mid z_{t-1}, o_t) \\
&\text{Transition model:} && \hat{z}_t \sim p_\theta(\hat{z}_t \mid z_{t-1}) \\
&\text{Action-conditional} \\
&\text{Representation model:} && s_t \sim q_\phi(s_t \mid s_{t-1}, a_{t-1}, z_t) \\
&\text{Transition model:} && \hat{s}_t \sim p_\phi(\hat{s}_t \mid s_{t-1}, a_{t-1}) \\
&\text{Image decoder:} && \hat{o}_t \sim p_\phi(\hat{o}_t \mid s_t) \\
&\text{Reward predictor:} && \hat{r}_t \sim p_\phi(\hat{r}_t \mid s_t), \qquad (3)
\end{aligned}$$

which is optimized by minimizing the following objective:

$$\begin{aligned}
\mathcal{L}(\phi, \theta) \doteq\ & \mathbb{E}_{q_\phi(s_{1:T} \mid a_{1:T}, z_{1:T}),\, q_\theta(z_{1:T} \mid o_{1:T})}\Big[\\
\sum_{t=1}^{T} \Big(& \underbrace{-\ln p_\phi(o_t \mid s_t)}_{\text{image log loss}} \underbrace{-\ln p_\phi(r_t \mid s_t)}_{\text{reward log loss}} \\
& + \underbrace{\beta_z \,\mathrm{KL}\left[q_\theta(z_t \mid z_{t-1}, o_t) \,\|\, p_\theta(\hat{z}_t \mid z_{t-1})\right]}_{\text{action-free KL loss}} \qquad (4) \\
& + \underbrace{\beta \,\mathrm{KL}\left[q_\phi(s_t \mid s_{t-1}, a_{t-1}, z_t) \,\|\, p_\phi(\hat{s}_t \mid s_{t-1}, a_{t-1})\right]}_{\text{action-conditional KL loss}} \Big)\Big],
\end{aligned}$$

(a) Stacked latent prediction model (b) Video-based intrinsic bonus

Figure 3. Illustration of our framework. (a) We stack an action-conditional prediction model on top of the pre-trained action-free prediction model. At inference time, the transition model in the action-conditional model is used to predict future states in the latent space conditioned on future potential actions. (b) To compute the intrinsic bonus, we first average pool a sequence of model states from the action-free prediction model, and apply random projection to reduce the dimension of representations while preserving distances. The intrinsic bonus for each observation is computed as the distance in the representation space to its k-nearest neighbor in samples from a replay buffer.

where β is a scale hyperparameter. Here, we note that we initialize the image decoder $p_\theta(o_t|s_t)$ with the pre-trained image decoder $p_\phi(o_t|z_t)$. We implement the transition model of action-conditional prediction model as RSSM, and the representation model as RSSM with dense layers that receive the model states of the action-free model as inputs. We refer to Appendix B for a more detailed formulation. In our experiments, we use $\beta_z = 0$ during fine-tuning, and only utilize the action-conditional RSSM for future imagination.

3.3. Video-based Intrinsic Bonus

It has been observed that good representations are crucial for efficient exploration in environments with high-dimensional observations (Laskin et al., 2021). To utilize useful information captured in the pre-trained representations for exploration, we propose a video-based intrinsic bonus. Our main idea is to increase the diversity of visited trajectories by utilizing it as an intrinsic bonus. Specifically, given a sequence of model states from the action-free prediction model $z_{t:t+\tau}$, we apply average pooling across the sequence dimension to obtain a trajectory representation $y_t = \text{Avg}(z_{t:t+\tau})$. Then, we utilize the distance of y_t to its k-nearest neighbor in samples from a replay buffer as a metric for measuring the diversity of trajectories. To summarize, our intrinsic bonus is defined as below (see Figure 3(b) for illustration):

$$r_t^{\text{int}} \doteq ||f(y_t) - f(y_t^k)||_2, \tag{5}$$

where f is a random projection (Bingham & Mannila, 2001) that maps the model state to a low-dimensional representation for compute-efficient distance computation, and y_t^k is a k-nearest neighbor of y_t in a minibatch. By explicitly encouraging the agents to visit more diverse trajectories instead of single states (Pathak et al., 2017; Burda et al., 2019; Pathak et al., 2019; Liu & Abbeel, 2021), it effectively encourages the agents to explore environments in a more long term manner, and thus learn more diverse behaviors. Then, the reward predictor is trained to predict the sum of r_t and λr_t^{int} as below:

$$\begin{aligned}\mathcal{L}^{\text{APV}}(\phi,\theta) \doteq \mathbb{E}_{q_\theta(s_{1:T}|a_{1:T},z_{1:T}),q_\phi(z_{1:T}|o_{1:T})}\Big[& \textstyle\sum_{t=1}^{T} \Big(\underbrace{-\ln p_\theta(o_t|s_t)}_{\text{image log loss}} \underbrace{-\ln p_\theta(r_t + \lambda r_t^{\text{int}}|h_t, z_t)}_{\text{APV reward log loss}} \\ & \underbrace{+\beta_z \text{KL}[q_\phi(z_t|z_{t-1}, o_t) \,\|\, p_\phi(\hat{z}_t|z_{t-1})]}_{\text{action-free KL loss}} \\ & \underbrace{+\beta \text{KL}[q_\theta(s_t|s_{t-1}, a_{t-1}, z_t) \,\|\, p_\theta(\hat{s}_t|s_{t-1}, a_{t-1})]}_{\text{action-conditional KL loss}} \Big) \Big],\end{aligned} \tag{6}$$

where λ is a hyperparameter that adjusts the tradeoff between exploitation and exploration. We find the intrinsic bonus provides large gains when combined with pre-training, as the pre-trained representations already contain useful representation from the beginning of the fine-tuning (see Figure 7(a) for supporting results). In our experiments, we utilize a sliding window of size τ for constructing a set of $\{y_t\}$ from trajectories in a minibatch, then compute the intrinsic bonus using them. For behavior learning, we utilize the actor-critic learning scheme of DreamerV2 (Hafner et al., 2021) that learns values with imagined rewards from future imaginary states and a policy that maximizes the values (see Appendix A for details). We also summarize the difference of APV to DreamerV2 in Appendix D.

4. Experiments

We designed our experiments to investigate the following:

- Can APV improve the sample-efficiency of vision-based RL in robotic manipulation tasks by performing action-free pre-training on videos from different domains?
- Can representations pre-trained on videos from manipulation tasks transfer to locomotion tasks?
- How does APV compare to a naïve fine-tuning scheme?
- What is the contribution of each of the proposed techniques in APV?
- How does pre-trained representations qualitatively differ from the randomly initialized representations?
- How does APV perform when additional in-domain videos or real-world natural videos are available?

Following Agarwal et al. (2021), we report the interquartile mean with bootstrap confidence interval (CI) and stratified bootstrap CI for results on individual tasks and aggregate results, respectively, across 8 runs for each task. Source codes and other resources are available at https://github.com/younggyoseo/apv.

4.1. Experimental Setup

Meta-world experiments. We first evaluate APV on various vision-based robotic manipulation tasks from Meta-world (Yu et al., 2020). In all manipulation tasks, the episode length is 500 steps without any action repeat, action dimension is 4, and reward ranges from 0 to 10. To evaluate the ability of APV to learn useful representations from different domains, we use videos collected in robotic manipulation tasks from RLBench (James et al., 2020) as pre-training data (see Figure 4)[1]. Specifically, we collect 10 demonstrations rendered with 5 camera views in 99 tasks from RLBench, giving a total of 4950 videos. We then train the action-free video prediction model by minimizing the objective in Equation 2 for 600K gradient steps. For downstream tasks, we fine-tune the model by minimizing the objective in Equation 6 for 250K environment steps, i.e., 500 episodes.

DeepMind Control Suite experiments. We also consider widely used robotic locomotion tasks from DeepMind Control Suite (Tassa et al., 2020). Following the common setup in this benchmark (Hafner et al., 2020), the episode length is 1000 steps with the action repeat of 2, and reward ranges from 0 to 1. For pre-training, we consider two datasets: (i) 1000 videos collected from Triped Walk (see Figure 10) and (ii) manipulation videos from RLBench. The former one is for evaluating the performance of APV on in-domain transfer setup similar to the setup in Stooke et al. (2021), while the latter one is for investigating whether the pre-trained representations can be transferred to extremely different domains, i.e., out-of-domain transfer. Specifically, we collect 1000 videos encountered during the training of DreamerV2 agent in Triped Walk and use these videos for pre-training. For downstream tasks, we fine-tune the model for 1M environment steps. See Appendix E for more details.

Figure 4. Illustration of experimental setups in our experiments with examples of image observations from environments. One can see that visuals in pre-training videos are notably different from the visuals in downstream manipulation and locomotion tasks.

Hyperparameters. For newly introduced hyperparameters, we use $_z = 1.0$ for pre-training, and $_z = 0, \ = 1.0$ for fine-tuning. We use $= 5$ for computing the intrinsic bonus. To make the scale of intrinsic bonus be 10% of extrinsic reward, we normalize the intrinsic reward and use $= 0.1, 1.0$ for manipulation and locomotion tasks, respectively. We find that increasing the hidden size of dense layers and the model state dimension from 200 to 1024 improves the performance of both APV and DreamerV2. We use $T = 25, 50$ for manipulation and locomotion tasks, respectively, during pre-training. Unless otherwise specified, we use the default hyperparameters of DreamerV2.

4.2. Meta-world Experiments

RLBench pre-training results. Figure 5 shows the learning curves of APV pre-trained using the RLBench videos on six robotic manipulation tasks from Meta-world. We find that APV consistently outperforms DreamerV2 in terms of sample-efficiency in all considered tasks. In particular, our framework achieves success rate above 60% on Lever Pull task while DreamerV2 completely fails to solve the task. These results show that APV can leverage action-free videos for learning useful representations that improve the sample-efficiency of vision-based RL.

---

[1] In this work, we do not consider a setup where we perform pre-training on Meta-world videos and fine-tuning for solving RLBench manipulation tasks, as existing RL algorithms struggle to solve challenging, sparsely-rewarded RLBench tasks.

Figure 5. Learning curves on manipulation tasks from Meta-world as measured on the success rate. APV with generative pre-training on videos collected in manipulation tasks from RLBench consistently outperforms DreamerV2 in terms of sample-efficiency. The solid line and shaded regions represent the interquartile mean and bootstrap confidence intervals, respectively, across eight runs.

Comparison with DrQ-v2. We also compare APV with a state-of-the-art model-free RL method DrQ-v2 (Yarats et al., 2021a) in Appendix F. We find that APV also significantly outperforms DrQ-v2 on most tasks, while DrQ-v2 struggles to achieve strong performance. While this aligns with the observation of Yarats et al. (2021a) where DreamerV2 outperformed DrQ-v2 on DeepMind Control Suite, investigating why DrQ-v2 fails is an interesting future direction.

Comparison with naïve fine-tuning. To verify the necessity of the proposed architecture for fine-tuning, we compare APV to a naïve fine-tuning scheme that initializes the action-conditional latent dynamics model with the pre-trained parameters of the action-free model (see Appendix E for the details). For a fair comparison, we do not utilize the intrinsic bonus for APV. Figure 6(a) shows that DreamerV2 with this naïve fine-tuning scheme (DreamerV2 w/ Naïve FT) does not provide large gains over DreamerV2, which implies that naïve fine-tuning quickly loses pre-trained representations. By contrast, we find that APV without intrinsic bonus consistently outperforms DreamerV2 by achieving $> 10\%$ higher success rate from the beginning of the fine-tuning, even though the same pre-trained model is used for fine-tuning. This shows the proposed architecture is crucial for effective fine-tuning.

Ablation study. To evaluate the contribution of the proposed techniques in APV, we report the performance of our framework with or without generative pre-training and intrinsic bonus in Figure 6(b). First, we observe that APV (Pre: X / Int: X), whose difference to DreamerV2 is the usage of stacked latent prediction model, achieves similar performance to DreamerV2. This implies that our performance gain is not from the architecture itself, but the way we utilize it for fine-tuning is important. We also find that our intrinsic bonus can improve the performance with or without pre-training, and generative pre-training can also improve the performance with or without intrinsic bonus. Importantly, one can see that the best performance is achieved when both components are combined. This implies that our proposed techniques synergistically contribute to the performance improvement.

Effects of video-based intrinsic bonus. We investigate the effect of considering multiple model states of length in Equation 5 instead of a single model state. Figure 6(c) shows that APV with $= 5$ achieves better performance than $\in \{1; 3\}$. We think this is because considering a sequence of observations, i.e., videos, enables us to utilize contextual information for encouraging agents to perform diverse behaviors. But we also find that APV with $= 10$ performs worse than $= 5$, which might be due to the complexity from average pooling over longer videos.

Qualitative analysis. We visually investigate why the pre-trained representations can be useful for unseen Meta-world

(a) Comparison with naïve fine-tuning (b) Effects of pre-training and intrinsic bonus (c) Length of future model states

Figure 6. Learning curves on manipulation tasks from Meta-world as measured on the success rate. We report the interquartile mean and stratified bootstrap confidence interval across total 48 runs over six tasks. (a) Comparison with a naïve fine-tuning scheme that initializes the action-conditional prediction model with the action-free prediction model. (b) Performance of APV with or without generative pre-training and intrinsic bonus. Here, *Pre* denotes generative pre-training, and *Int* denotes intrinsic bonus. (c) Performance of APV with varying the length of future model states used for computing the intrinsic bonus.

(a) Video representations (b) Importance of dynamics information (c) Effects of in-domain videos

Figure 7. (a) t-SNE visualization of average pooled model states from the action-free prediction model. (b) Learning curves of APV on six manipulation tasks when only the parameters of convolutional image encoder and decoder are transferred, i.e., without transferring dynamics information captured in recurrent models. (c) Learning curves of APV on four manipulation tasks when pre-trained on RLBench videos only (RLB), and on both of RLBench videos and additional in-domain Meta-world videos (RLB + MW). Dotted and bold lines indicate the performance when all parameters are fine-tuned, and the representation model of the action-free model is frozen, respectively.

tasks. Specifically, we sample video clips of length 25 from the 10 videos of six manipulation task, and visualize the averaged model states from the sampled videos using t-SNE (Van der Maaten & Hinton, 2008) in Figure 7(a), where colors indicate the tasks. We find that the pre-trained representations from each task are clustered, while randomly initialized ones are entangled. This shows that the pre-trained representations capture information about the tasks without access to Meta-world videos during pre-training.

Importance of dynamics information. To investigate whether the performance gain comes from utilizing the dynamics information captured in pre-trained representations or visual information captured in the image encoder and decoder, we report the performance of APV when only the pre-trained parameters of the convolutional image encoder and decoder are transferred. Figure 7(b) shows that APV (Encoder / Decoder Only) performs worse than APV, which demonstrates that utilizing the dynamics information learned from diverse videos is crucial for performance.

Pre-training with additional in-domain videos. We also consider an experimental setup where we have an access to additional videos with similar visuals collected on the target domain. Specifically, we collect 10 videos on each task from ML-10 training tasks in Meta-world, i.e., total of 100 videos, and utilize these videos for pre-training in conjunction with RLBench videos. Then, we fine-tune the pre-trained model for solving 4 manipulation tasks that were not seen during pre-training. To evaluate how additional videos affect representation learning, we report the performance of APV when the representation model of the action-free prediction model is frozen, and use it as a proxy for evaluating the quality of representations.

Figure 7(c) shows that pre-training with additional Meta-world videos (APV (RLB + MW)) achieves almost similar performance to pre-training with only RLBench videos (APV (RLB)). This shows that pre-trained representations with RLBench videos already learns useful representations so that they can be quickly fine-tuned for solving Meta-

(a) State regression (b) Reward regression (c) Effects of pre-training datasets

Figure 8. We report the prediction error on held-out test sets obtained while training a regression model to predict (a) proprioceptive states and (b) rewards. We observe that RLBench pre-trained model achieves a small prediction error throughout training from the beginning. (c) Learning curves on manipulation tasks from Meta-world when pre-trained with manipulation, locomotion, and real-world videos. We report the interquartile mean and stratified bootstrap confidence interval over 48 runs over six tasks.

Figure 9. Learning curves on locomotion tasks from DeepMind Control Suite as measured on the episode return. Interestingly, we find that APV pre-trained on manipulation videos from RLBench consistently outperforms DreamerV2. We also observe that utilizing in-domain videos from Triped Walk leads to further improved performance. The solid line and shaded regions represent the interquartile mean and bootstrap confidence intervals, respectively, across eight runs.

world tasks. However, we observe that pre-training with in-domain videos significantly improves the performance when the pre-trained representation model is frozen. This implies that additional in-domain videos help for addressing the domain gap between pre-training and fine-tuning.

State and reward regression analysis. Our hypothesis for how pre-training from videos is useful for RL is that pre-trained representations can capture useful information (e.g., rewards and proprioceptive states) for solving RL tasks. To test this hypothesis, we train a regression model that predicts proprioceptive states and rewards on the pre-collected Triped dataset. Figure 8(a) and Figure 8(b) show the regression performances with and without RLBench pre-trained representations. We find that the RLBench pre-trained model not only has a small prediction error at the beginning, but also quickly converges compared to the model trained from scratch. This shows that pre-training helps the agent to understand environment.

Effects of pre-training datasets To investigate the effect of pre-training domains, we evaluate the performance of APV on Meta-world when pre-trained with different datasets. To this end, we consider pre-training on manipulation videos from RLBench (i.e., APV w/ Manipulation Videos), locomotion videos from Triped dataset (i.e., APV w/ Locomotion Videos), and real-world natural videos of people performing diverse behaviors (i.e., APV w/ Real-world). Specifically, for real-world videos, we utilize Something-Something-V2 dataset (Goyal et al., 2017), which contains 159K videos of people performing actions. Because we find that our video prediction model suffers from severe underfitting to real-world video datasets (see Appendix G for the examples of blurry predicted future frames), we use subsampled 1.5K videos for pre-training. Moreover, it would be interesting to study whether pre-training on complex robotics video (e.g., RoboNet (Dasari et al., 2019)) can improve performance on more complex robotics tasks (e.g., RLBench).

In Figure 8(c), we observe that pre-training with videos from more similar domains can be helpful. For instance, APV w/ Manipulation outperforms APV w/ Locomotion, which shows that pre-training with manipulation videos can be more effective for manipulation tasks. However,

Figure 10. Illustration of additional experimental setup in DeepMind Control Suite experiments. We use videos collected in Triped Walk for pre-training, and then fine-tune the pre-trained model for solving downstream Quadruped and Hopper locomotion tasks.

APV w/ Real-world struggles to outperform the baseline without pre-training even though there is no underfitting issue, which might be due to the small number of training data and domain gap between egocentric real-world videos and third-person robotic videos. It would be an interesting direction to investigate how different aspects of pre-training datasets, e.g., size, domain, and point of view.

4.3. DeepMind Control Suite Experiments

Figure 9 shows the learning curves of APV and DreamerV2 on locomotion tasks. Interesingly, we find that APV pre-trained on manipulation videos from RLBench (pink curves) consistently achieves better performance than DreamerV2 (black curves). This demonstrates that representations pre-trained using manipulation videos, which have notably different visuals and objectives, effectively capture dynamics information useful for quickly learning the dynamics of locomotion environments. Also, by utilizing the videos from a similar domain, i.e., Triped environment, the performance of APV is further improved. We provide additional experimental results that evaluate the contribution of the proposed techniques on DeepMind Control Suite tasks in Appendix I.

5. Discussion

In this work, we introduce a vision-based RL framework that learns representations useful for understanding the dynamics of downstream domains via action-free pre-training on videos, and utilizes the pre-trained representations for fine-tuning. Our experimental results demonstrate that APV can improve both sample-efficiency and final performances of vision-based RL on various manipulation and locomotion tasks, by effectively transferring the pre-trained representations from unseen domains. However, one limitation of our work is that pre-training is conducted only on simulated robotic videos, which is because of the underfitting issue reported in our analysis. Given that, one interesting direction would be to scale up the architecture or utilize recently developed video prediction architectures (Babaeizadeh et al., 2021; Yan et al., 2021), and investigate how the prediction quality affects the performance. Another interesting direction is to incorporate generalization approaches in RL (Tobin et al., 2017; Higgins et al., 2017; Laskin et al., 2020a) to deal with the difference between pre-training and fine-tuning domains. Moreover, while our work focuses on representation learning via generative pre-training, another interesting future direction would be to investigate the performance of representation learning schemes such as masked prediction (He et al., 2021; Xiao et al., 2022; Yu et al., 2022), latent reconstruction (Yu et al., 2021; Schwarzer et al., 2021a), and contrastive learning (Oord et al., 2018). It would also be interesting to utilize our pre-trained representations for tasks that require more long-term reasoning, such as reward learning from preferences (Park et al., 2022) or videos (Chen et al., 2021). By presenting a generic framework that can leverage videos with diverse visuals and embodiments for pre-training, we hope this work would facilitate future research on unsupervised pre-training for RL.

Acknowledgements

We would like to thank Danijar Hafner, Junsu Kim, Fangchen Liu, Jongjin Park, Jihoon Tack, Wilson Yan, and Sihyun Yu for helpful discussions. We also thank Cirrascale Cloud Services[2] for providing compute resources. This work was partially supported by Hong Kong Centre for Logistics Robotics, Center for Human Compatible AI (CHAI), Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00166, Self-directed AI Agents with Problem-solving Capability).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

Aigner, S. and Körner, M. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans. *arXiv preprint arXiv:1810.01325*, 2018.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell,

---

[2] https://cirrascale.com

G., Ribas, R., et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.-A., and Hjelm, R. D. Unsupervised state representation learning in atari. In *Advances in Neural Information Processing Systems*, 2019.

Aytar, Y., Pfaff, T., Budden, D., Paine, T. L., Wang, Z., and de Freitas, N. Playing hard exploration games by watching youtube. In *Advances in Neural Information Processing Systems*, 2018.

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. Stochastic variational video prediction. In *International Conference on Learning Representations*, 2018.

Babaeizadeh, M., Saffar, M. T., Nair, S., Levine, S., Finn, C., and Erhan, D. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2016.

Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Bingham, E. and Mannila, H. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Castro, P. S. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Chang, M., Gupta, A., and Gupta, S. Semantic visual navigation by watching youtube videos. In *Advances in Neural Information Processing Systems*, 2020.

Chen, A. S., Nair, S., and Finn, C. Learning generalizable robotic reward functions from" in-the-wild" human videos. In *Proceedings of Robotics: Science and Systems*, 2021.

Chen, C., Wu, Y.-F., Yoon, J., and Ahn, S. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Clark, A., Donahue, J., and Simonyan, K. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C. Robonet: Large-scale multi-robot learning. *Conference on Robot Learning*, 2019.

Denton, E. and Fergus, R. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Dwibedi, D., Tompson, J., Lynch, C., and Sermanet, P. Learning actionable representations from visual observations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

Edwards, A., Sahni, H., Schroecker, Y., and Isbell, C. Imitating latent policies from observation. In *International Conference on Machine Learning*, 2019.

Finn, C., Goodfellow, I., and Levine, S. Unsupervised learning for physical interaction through video prediction. In Advances in neural information processing systems, 2016a.

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016b.

Franceschi, J.-Y., Delasalles, E., Chen, M., Lamprier, S., and Gallinari, P. Stochastic latent residual video prediction. In International Conference on Machine Learning, 2020.

Gelada, C., Kumar, S., Buckman, J., Nachum, O., and Bellemare, M. G. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, 2019.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In International Conference on Learning Representations, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Advances in Neural Information Processing Systems, 2014.

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al. The" something something" video database for learning and evaluating visual common sense. In Proceedings of the IEEE international conference on computer vision, 2017.

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, 2018.

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, 2019.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In International Conference on Learning Representations, 2020.

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models. In International Conference on Learning Representations, 2021.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In International Conference on Machine Learning, 2019.

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, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377, 2021.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. Darla: Improving zero-shot transfer in reinforcement learning. In International Conference on Machine Learning, 2017.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. In Advances in Neural Information Processing Systems, 2016.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. In International Conference on Learning Representations, 2017.

James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. Rlbench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters, 2020.

Jang, Y., Kim, G., and Song, Y. Video prediction with appearance and motion conditions. In International Conference on Machine Learning, 2018.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. Model-based reinforcement learning for atari. In International Conference on Learning Representations, 2019.

Kalashnikov, D., Varley, J., Chebotar, Y., Swanson, B., Jonschkowski, R., Finn, C., Levine, S., and Hausman, K. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. arXiv preprint arXiv:2104.08212, 2021.

Kalchbrenner, N., Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. Video pixel networks. In International Conference on Machine Learning, 2017.

Kim, S. W., Zhou, Y., Philion, J., Torralba, A., and Fidler, S. Learning to simulate dynamic environments with gamegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

Kim, S. W., Philion, J., Torralba, A., and Fidler, S. Drivegan: Towards a controllable high-quality neural simulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In International Conference on Learning Representations, 2014.

Kwon, Y.-H. and Park, M.-G. Predicting future frames using retrospective cycle gan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. In Advances in Neural Information Processing Systems, 2020a.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In International Conference on Machine Learning, 2020b.

Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L., and Abbeel, P. Urlb: Unsupervised reinforcement learning benchmark. In Advances in Neural Information Processing Systems Datasets and Benchmarks Track, 2021.

Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., and Levine, S. Stochastic adversarial video prediction. arXiv preprint arXiv:1804.01523, 2018.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. arXiv preprint arXiv:1906.05274, 2019.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research, 2016.

Liu, H. and Abbeel, P. Behavior from the void: Unsupervised active pre-training. arXiv preprint arXiv:2103.04551, 2021.

Liu, Y., Gupta, A., Abbeel, P., and Levine, S. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018.

Lotter, W., Kreiman, G., and Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. In International Conference on Learning Representations, 2017.

Luc, P., Clark, A., Dieleman, S., Casas, D. d. L., Doron, Y., Cassirer, A., and Simonyan, K. Transformation-based adversarial video prediction on large-scale data. arXiv preprint arXiv:2003.04035, 2020.

Mazoure, B., Combes, R. T. d., Doan, T., Bachman, P., and Hjelm, R. D. Deep reinforcement and infomax learning. In Advances in Neural Information Processing Systems, 2020.

Michalski, V., Memisevic, R., and Konda, K. Modeling deep temporal dependencies with recurrent grammar cells. In Advances in Neural Information Processing Systems, 2014.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. Nature, 2015.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In European conference on computer vision, 2016.

Oh, J., Guo, X., Lee, H., Lewis, R., and Singh, S. Action-conditional video prediction using deep networks in atari games. In Advances in Neural Information Processing Systems, 2015.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. In Advances in Neural Information Processing Systems, 2018.

Ostrovski, G., Bellemare, M. G., Oord, A. v. d., and Munos, R. Count-based exploration with neural density models. In International Conference on Machine Learning, 2017.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., and Lee, K. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In International Conference on Learning Representations, 2022.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In International Conference on Machine Learning, 2017.

Pathak, D., Gandhi, D., and Gupta, A. Self-supervised exploration via disagreement. In International Conference on Machine Learning, 2019.

Peng, X. B., Kanazawa, A., Malik, J., Abbeel, P., and Levine, S. Sfv: Reinforcement learning of physical skills from videos. ACM Transactions On Graphics (TOG), 2018.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. Technical report, 2018.

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604, 2014.

Reed, S., Oord, A., Kalchbrenner, N., Colmenarejo, S. G., Wang, Z., Chen, Y., Belov, D., and Freitas, N. Parallel multiscale autoregressive density estimation. In International Conference on Machine Learning, 2017.

Rybkin, O., Zhu, C., Nagabandi, A., Daniilidis, K., Mordatch, I., and Levine, S. Model-based reinforcement learning via latent-space collocation. In International Conference on Machine Learning, 2021.

Schmeckpeper, K., Rybkin, O., Daniilidis, K., Levine, S., and Finn, C. Reinforcement learning with videos: Combining offline observations with interaction. In Conference on Robot Learning, 2020a.

Schmeckpeper, K., Xie, A., Rybkin, O., Tian, S., Daniilidis, K., Levine, S., and Finn, C. Learning predictive models from observation and interaction. In European Conference on Computer Vision, 2020b.

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

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. In International Conference on Learning Representations, 2021a.

Schwarzer, M., Rajkumar, N., Noukhovitch, M., Anand, A., Charlin, L., Hjelm, D., Bachman, P., and Courville, A. Pretraining representations for data-efficient reinforcement learning. In Advances in Neural Information Processing Systems, 2021b.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. In International Conference on Machine Learning, 2020.

Seo, Y., Chen, L., Shin, J., Lee, H., Abbeel, P., and Lee, K. State entropy maximization with random encoders for efficient exploration. In International Conference on Machine Learning, 2021.

Seo, Y., Lee, K., Liu, F., James, S., and Abbeel, P. Autoregressive latent video prediction with high-fidelity image generator, 2022. URL https://openreview.net/forum?id=K-hiHQXEQog.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. Time-contrastive networks: Self-supervised learning from video. In 2018 IEEE international conference on robotics and automation (ICRA), 2018.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. nature, 2017.

Srinivas, A., Laskin, M., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In Internatial Conference on Machine Learning, 2020.

Srivastava, N., Mansimov, E., and Salakhudinov, R. Unsupervised learning of video representations using lstms. In International Conference on Machine Learning, 2015.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. In International Conference on Machine Learning, 2021.

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

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In Advances in Neural Information Processing Systems, 2017.

Tassa, Y., Tunyasuvunakool, S., Muldal, A., Doron, Y., Liu, S., Bohez, S., Merel, J., Erez, T., Lillicrap, T., and Heess, N. dm_control: Software and tasks for continuous control. arXiv preprint arXiv:2006.12983, 2020.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), 2017.

Torabi, F., Warnell, G., and Stone, P. Generative adversarial imitation from observation. arXiv preprint arXiv:1807.06158, 2018.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. Journal of machine learning research, 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. Attention is all you need. In Advances in Neural Information Processing Systems, 2017.

Villegas, R., Pathak, A., Kannan, H., Erhan, D., Le, Q. V., and Lee, H. High fidelity video prediction with large stochastic recurrent neural networks. Advances in Neural Information Processing Systems, 2019.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature, 2019.

Vondrick, C., Pirsiavash, H., and Torralba, A. Generating videos with scene dynamics. In Advances in Neural Information Processing Systems, 2016.

Weissenborn, D., Tackstrom, O., and Uszkoreit, J. Scaling autoregressive video models. In International Conference on Learning Representations, 2020.

Wu, C., Huang, L., Zhang, Q., Li, B., Ji, L., Yang, F., Sapiro, G., and Duan, N. Godiva: Generating open-domain videos from natural descriptions. arXiv preprint arXiv:2104.14806, 2021.

Xiao, T., Radosavovic, I., Darrell, T., and Malik, J. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Yan, W., Zhang, Y., Abbeel, P., and Srinivas, A. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Yan, W., Okumura, R., James, S., and Abbeel, P. Patch-based object-centric transformers for efficient video generation. *arXiv preprint arXiv:2206.04003*, 2022.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 2019.

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., Fergus, R., Lazaric, A., and Pinto, L. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, 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*, 2021c.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2020.

Yu, T., Lan, C., Zeng, W., Feng, M., Zhang, Z., and Chen, Z. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Yu, T., Zhang, Z., Lan, C., Chen, Z., and Lu, Y. Mask-based latent reconstruction for reinforcement learning. *arXiv preprint arXiv:2201.12096*, 2022.

Zakka, K., Zeng, A., Florence, P., Tompson, J., Bohg, J., and Dwibedi, D. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, 2022.

Zhan, A., Zhao, P., Pinto, L., Abbeel, P., and Laskin, M. A framework for efficient robotic manipulation. *arXiv preprint arXiv:2012.07975*, 2020.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., and Levine, S. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, 2019.

Ziebart, B. D. Modeling purposeful adaptive behavior with the principle of maximum causal entropy, 2010.

A. Behavior Learning

For behavior learning, we utilize actor-critic learning scheme of Hafner et al. (2021) where the agent maximizes the values of imagined future states by propagating the analytic gradients back through the world model separately learned with Equation 3. Specifically, given a stochastic actor and a deterministic critic as below:

$$\begin{aligned}&\text{Actor:} \quad \hat{a}_t \quad p\,(\hat{a}_t | \hat{s}_t)\\ &\text{Critic:} \quad v\,(\hat{s}_t) \quad \mathrm{E}_{p\,;p}\Big[\textstyle\sum_{i\,t}\ {}^{i\,t}\hat{r}_i\Big]\ ;\end{aligned} \tag{7}$$

a sequence of H future states $\hat{s}_{1:H}$ is recursively predicted conditioned on initial states $\hat{s}_0$, which are model states encountered during world model training, using the stochastic actor and the transition predictor of world model. Then a deterministic critic is learned to regress the target (Schulman et al., 2015; Sutton & Barto, 2018) as follows:

$$L(\ ) \doteq \mathrm{E}_{p\,;p}\left[\sum_{t=1}^{H\ 1} \frac{1}{2}\ v\,(\hat{s}_t)\quad \mathrm{sg}(V_t\,)^{2}\right]\ ; \tag{8}$$

where sg is a stop gradient function, and -return V_t is defined using the future states as follows:

$$V_t \doteq \hat{r}_t + \begin{cases}(1\quad)v\,(\hat{s}_{t+1}) + \ V_{t+1} & \text{if } t < H\\ v\,(\hat{s}_H) & \text{if } t = H\end{cases} \tag{9}$$

Then, the actor is trained to maximize the -return in Equation 8 by leveraging the straight-through estimator (Bengio et al., 2013) for backpropagating the value gradients through the discrete world model as follows:

$$L(\ ) \doteq \mathrm{E}_{p\,;p}\quad V_t \qquad H[a_t|\hat{s}_t]\ ; \tag{10}$$

where the entropy of actor $H[a_t|\hat{s}_t]$ is maximized to encourage exploration, and is a hyperparameter that adjusts the strength of entropy regularization.

B. Formulation with Recurrent State-Space Model

While we follow the formulation of Hafner et al. (2020) for our main draft, we additionally provide a detailed formulation of the action-free latent video prediction model and stacked latent prediction model implemented with a recurrent state-space model (RSSM; Hafner et al. 2019) that consists of deterministic and stochastic components, for better understanding of our implementation.

B.1. Action-free Latent Video Prediction Model

The main component of our model is an action-free RSSM that consists of (i) a recurrent model that computes deterministic states h_t^{AF} conditioned on previous states for each time step t, (ii) a representation model that computes posterior stochastic states z_t^{AF} conditioned on determinsitic states h_t^{AF} and image observations o_t, (iii) a transition predictor that computes prior stochastic states $\hat{z}_t^{AF}$ without access to image observations. We define the model state as the concatenation of h_t^{AF} and z_t^{AF}, which is used as an input to the decoder. The model can be summarized as:

$$\begin{aligned}&\text{Recurrent model:} && h_t^{AF} = f\ \ (h_{t\ 1}^{AF}; z_{t\ 1}^{AF})\\ &\text{Representation model:} && z_t^{AF} \quad q\ \ (z_t^{AF}|\, h_t^{AF}; o_t)\\ &\text{Transition predictor:} && \hat{z}_t^{AF} \quad p\ \ (\hat{z}_t^{AF}|\, h_t^{AF})\\ &\text{Image decoder:} && \hat{o}_t \quad p\ \ (\hat{o}_t\,|\, h_t^{AF}; z_t^{AF})\end{aligned} \tag{11}$$

At inference time, the recurrent model and the transition predictor (i.e., learned prior) are used for predicting future model states, from which the image decoder reconstruct future frames. All the components of the model parameterized by are jointly optimized by minimizing the following loss:

$$L(\ ) \doteq \mathrm{E}_{q\ (z_{1:T}^{AF}\,|\,o_{1:T})}\Big[\sum_{t=1}^{T}\ \underbrace{\ln p\ \ (o_t\,|\,h_t^{AF}; z_t^{AF})}_{\text{image log loss}} + \underbrace{{}^{AF}\mathrm{KL}\ \ q\ \ (z_t^{AF}|\,h_t^{AF}; o_t)\,||\,q\ \ (\hat{z}_t^{AF}|\,h_t^{AF})}_{\text{action-free RSSM KL loss}}\Big]\ ; \tag{12}$$

which is a negative variational lower bound (ELBO; Kingma & Welling 2014) objective where ^{AF} is a scale hyperparameter and T is the length of training sequences in a minibatch.

B.2. Stacked Latent Prediction Model

Our stacked latent prediction model consists of the action-free RSSM followed by the action-conditional RSSM as below:

$$
\begin{array}{lll}
 & \textbf{Action-free RSSM} & \\
\left\{ \begin{array}{l} \\ \\ \\ \end{array} \right. & \text{Recurrent model:} & h_t^{\text{AF}} = f\ (h_{t\ 1}^{\text{AF}}; z_{t\ 1}^{\text{AF}}) \\
 & \text{Representation model:} & z_t^{\text{AF}} \quad q\ (z_t^{\text{AF}} | h_t^{\text{AF}}; o_t) \\
 & \text{Transition predictor:} & \hat{z}_t^{\text{AF}} \quad p\ (\hat{z}_t^{\text{AF}} | h_t^{\text{AF}}) \\
 & \textbf{Action-conditional RSSM} & \\
\left\{ \begin{array}{l} \\ \\ \\ \end{array} \right. & \text{Recurrent model:} & h_t^{\text{AC}} = f\ (h_{t\ 1}^{\text{AC}}; z_{t\ 1}^{\text{AC}}; a_{t\ 1}) \\
 & \text{Representation model:} & z_t^{\text{AC}} \quad q\ (z_t^{\text{AC}} | h_t^{\text{AC}}; a_{t\ 1}; h_t^{\text{AF}}; z_t^{\text{AF}}) \\
 & \text{Transition predictor:} & \hat{z}_t^{\text{AC}} \quad p\ (\hat{z}_t^{\text{AC}} | h_t^{\text{AC}}) \\
 & \text{Image decoder:} & \hat{o}_t \quad p\ (\hat{o}_t | h_t^{\text{AC}}; z_t^{\text{AC}}) \\
 & \text{Reward predictor:} & \hat{r}_t \quad p\ (\hat{r}_t | h_t^{\text{AC}}; z_t^{\text{AC}})
\end{array} \tag{13}
$$

All components parameterized by and are optimized jointly by minimizing the following:

$$
\begin{aligned}
L(\ ;\ ) \doteq \mathbb{E}_{q\ (z^{\text{AC}}_{1:T} | a_{1:T}; h^{\text{AF}}_{1:T}; z^{\text{AF}}_{1:T}); q\ (z^{\text{AF}}_{1:T} | o_{1:T})} \Big[\sum_{t=1}^{T} & \underbrace{\ln p\ (o_t | h_t^{\text{AC}}; z_t^{\text{AC}})}_{\text{image log loss}} \underbrace{\ln p\ (r_t | h_t^{\text{AC}}; z_t^{\text{AC}})}_{\text{reward log loss}} \\
& \underbrace{+\ {}^{\text{AF}}\mathrm{KL}\ q\ (z_t^{\text{AF}} | h_t^{\text{AF}}; o_t) \| q\ (\hat{z}_t^{\text{AF}} | h_t^{\text{AF}})}_{\text{action-free RSSM KL loss}} \underbrace{+\ {}^{\text{AC}}\mathrm{KL}\ q\ (z_t^{\text{AC}} | h_t^{\text{AC}}; h_t^{\text{AF}}; z_t^{\text{AF}}) \| q\ (\hat{z}_t^{\text{AC}} | h_t^{\text{AC}})}_{\text{action-conditional RSSM KL loss}} \Big] ;
\end{aligned} \tag{14}
$$

where ${}^{\text{AC}}$ is a scale hyperparameter.

C. Extended Related Work

Video prediction. A line of works close to our work is video prediction methods, that aims to predict the future frames conditioned on images (Michalski et al., 2014; Ranzato et al., 2014; Srivastava et al., 2015; Vondrick et al., 2016; Lotter et al., 2017), texts (Wu et al., 2021), and actions (Oh et al., 2015; Finn et al., 2016a), which would be useful for various applications, e.g., control with world models (Hafner et al., 2019; Kaiser et al., 2019; Rybkin et al., 2021), and simulator development (Kim et al., 2020; 2021). Recent successful approaches include generative adversarial networks (GANs; Goodfellow et al. 2014) known to generative a sequence of frames by introducing adversarial discriminators that makes prediction based on temporal information (Aigner & Körner, 2018; Jang et al., 2018; Kwon & Park, 2019; Clark et al., 2019; Luc et al., 2020), and autoregressive video prediction models (Babaeizadeh et al., 2018; Kalchbrenner et al., 2017; Reed et al., 2017; Denton & Fergus, 2018; Lee et al., 2018; Villegas et al., 2019; Weissenborn et al., 2020; Wu et al., 2021; Babaeizadeh et al., 2021; Yan et al., 2021; Seo et al., 2022; Yan et al., 2022). Our work builds on prior latent video prediction methods that utilize a state-space model to the domain of video prediction (Hafner et al., 2019; Franceschi et al., 2020), which enables us to predict future states without conditioning on predicted frames in an efficient manner. It would be an interesting direction to scale up current architecture or develop a new high-fidelity latent prediction model and utilize it for pre-training for RL.

Exploration in RL. Exploration in RL has been studied for encouraging the agents to visit more diverse states by maximizing the entropy of the action space (Ziebart, 2010; Haarnoja et al., 2018) and the various form of intrinsic bonus, including prediction errors (Houthooft et al., 2016; Pathak et al., 2017; Burda et al., 2019; Sekar et al., 2020), count-based state novelty (Bellemare et al., 2016; Tang et al., 2017; Ostrovski et al., 2017), and state entropy (Hazan et al., 2019; Lee et al., 2019; Liu & Abbeel, 2021; Seo et al., 2021; Yarats et al., 2021b). Our framework differs in that we explicitly consider a sequence of future model states for computing the intrinsic bonus, while previous works only consider a single state. In this work, we show that this encourages the agents to explore the environments in a more long-term manner, thus perform more diverse behaviors.

D. Difference to DreamerV2

Architecture. The main difference lies in the architecture of world models used for imagining future model states. APV utilizes a stacked latent prediction model where the action-free prediction model first processes o_t into representations z_t and then the action-conditional prediction model processes them into s_t conditioned on additional actions. In contrast, DreamerV2 utilizes a RSSM where the action-conditional prediction model directly processes o_t into model states s_t conditioned on actions. Namely, in terms of architecture, APV can be seen as a DreamerV2 that takes representations from the action-free prediction model as inputs instead of raw observations.

Behavior learning. The behavior learning scheme of APV is same as DreamerV2 except that APV learns a reward predictor to predict the sum of both extrinsic reward and intrinsic reward. In terms of behavior learning, DreamerV2 can be seen as a specialized case of APV with a scale hyperparameter $= 0$.

E. Experimental Details

Implementation. We build our framework on top of the official implementation of DreamerV2[3], which is based on TensorFlow (Abadi et al., 2016). We use a single Nvidia RTX3090 GPU and 10 CPU cores for each training run. The training time required for pre-training of APV is 24 hours, and fine-tuning of APV requires 4.75 hours for Meta-world experiments, and 6.25 hours for DeepMind Control Suite experiments, when used with XLA optimization. This takes longer than training of vanilla DreamerV2, which requires 3.5 hours for Meta-world experiments, and 4.25 hours for DeepMind Control Suite experiments.

Dataset details. For data collection in Meta-world environment, we use the `corner` viewpoint for rendering. We use scripted policy availble in the official implementation[4]. For RLBench environment, we use 5 camera viewpoints consisting of `front_rgb`, `left_shoulder_rgb`, `overhead_rgb`, `right_shoulder_rgb`, and `wrist_rgb`. We also utilize the scripted policy available in the official implementation[5]. For data collection in Triped Walk, we build the tasks by modifying the Quadruped Walk available in DeepMind Control Suite.

Model details. For DeepMind Control Suite experiments, we find that utilizing the concatenation of the model state z_t and the representation from the image encoder as inputs to the action-conditional prediction model improves the performance, but in Meta-world experiments, we observe that there is no difference to only utilizing z_t. Therefore, we only use the concatenation as inputs for DeepMind Control Suite experiments. For Meta-world experiments, during fine-tuning, we find that not updating the pre-trained video prediction model at the initial phase of DreamerV2 (which is called *pretrain* in the original source code) leads to slightly better performance, so we use this for all Meta-world experiments. We also find that increasing the hidden size of linear layers and the dimension of deterministic states in RSSMs from 200 to 1024 improves the performance of both our framework and DreamerV2. In all experiments, we use increased model size for all experiments. Specifically, in our experiments, DreamerV2 has 31M parameters and APV has 45M parameters, while DreamerV2 with default hyperparameters has 15M parameters (see Figure 6(b) for the experimental results demonstrating that the performance gain from APV is not from the increased number of parameters). Unless otherwise specified, we follow the architectures and hyperparameters used in Hafner et al. (2021).

Video-based intrinsic bonus details. We use $k = 16$ for all experiments for computing intrinsic bonus by measuring the distance to k-NN representation. For computing the intrinsic bonus, we construct a queue of size 4096 that contains the recent representations from the action-free model, and use the representations in queue for finding k-NN state. Our preliminary internal results show that this technique of introducing the queue is not critical to the performance, but we leave the technique here, as it is used in our reported experimental results.

Naïve fine-tuning details. We implement the naïve fine-tuning scheme in Figure 6(a) by (i) zero-masking the action inputs during pre-training, and (ii) removing the masks at fine-tuning phase and re-initializing the parameters whose inputs are actions. We also applied gradient clipping and warm up scheme to both this baseline and DreamerV2, and find that these techniques does not resolve the difficulty of dealing with additional action inputs.

---

[3] `https://github.com/danijar/dreamerv2`

[4] `https://github.com/rlworkgroup/metaworld/tree/master/metaworld`

[5] `https://github.com/stepjam/RLBench`

F. Meta-world Experiments with DrQ-v2

We report the performance of the state-of-the-art model-free RL method DrQ-v2 on vision-based robotic manipulation tasks from Meta-world. We use image observations of $64 \times 64 \times 3$, and conduct the hyperparameter search over action repeat of $\{1, 2\}$ and the frame stacking of $\{3, 6\}$. We find that action repeat of 1 with frame stacking of 6 performs the best. As shown in Figure 11, we find that DrQ-v2 struggles to achieve competitive performance on most of the considered tasks, which necessitates more analysis to understand the reason of a failure.

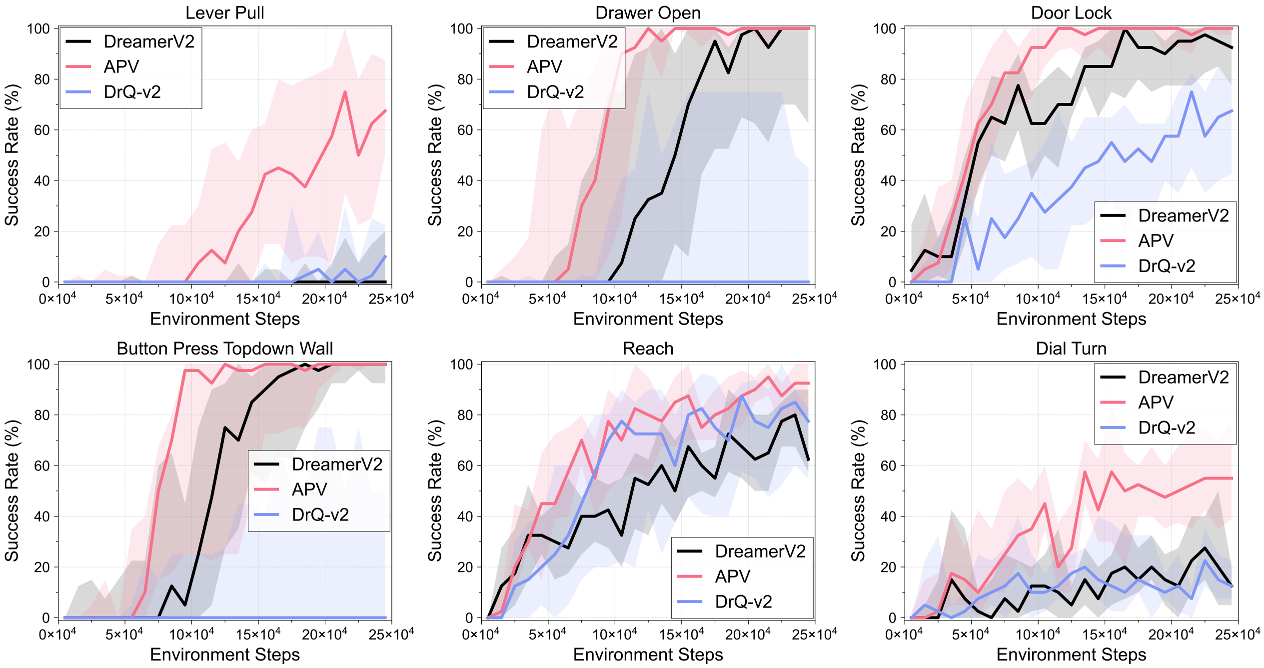

Figure 11. Learning curves on manipulation tasks from Meta-world as measured on the success rate. The solid line and shaded regions represent the interquartile mean and bootstrap confidence intervals, respectively, across eight runs.

G. Real-World Video Prediction on Something-Something-V2

We report the future frames predicted by our action-free video prediction model trained on real-world videos from Something-Something-V2 (Goyal et al., 2017). One can see that the model severely suffers from underfitting, and generates blurry frames. Developing a lightweight, high-fidelity video prediction model for RL would be an interesting future direction. For instance, we can consider adopting RSSM architecture based on Transformers (Chen et al., 2022).

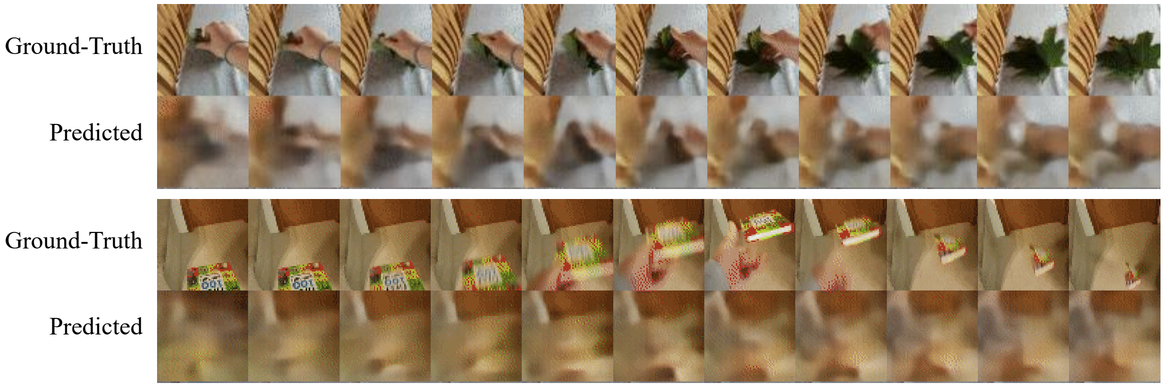

Figure 12. Future frames predicted by our action-free video prediction model on Something-Something-V2 dataset. We observe that our model severely suffers from underfitting, generating only blurry frames.

H. Video Prediction on RLBench and Meta-world

We report the future frames predicted by our action-free video prediction model trained on videos from RLBench (James et al., 2020) and Meta-world (Yu et al., 2020). One can see that predicted frames on both datasets capture dynamics information of robots (e.g., how robots are moving towards objects), in contrast to the prediction on Something-Something-V2 where predicted frames are so blurry that it is difficult to see which behavior is performed in the videos. We also observe that prediction quality in Meta-world is much better than that of RLBench, which is because Meta-world videos have more simple visuals and are collected in smaller number of tasks.

Figure 13. Future frames predicted by our action-free video prediction model on RLBench (top) and Meta-world (bottom). We observe that predicted frames on both datasets capture dynamics information of robots (e.g., how robots are moving towards objects), while prediction in RLBench is not high-quality as in Meta-world which is a more simple domain.

I. Ablation Study on DeepMind Control Suite

We provide the results from ablation studies on locomotion tasks from DeepMind Control Suite in Figure 14. Interestingly, we find that pre-training can significantly improve the performance without intrinsic bonus on Quadruped tasks, but APV with only intrinsic bonus does not make significant difference over vanilla DreamerV2. This might be because the visual observations in Quadruped tasks are very complex, so intrinsic bonus based on randomly initialized representations becomes not particularly useful in the tasks. But APV with both pre-training and intrinsic bonus performs best, which shows that pre-training representations can provide useful information from the beginning of the fine-tuning. On the other hand, on Hopper Hop, we find that APV without intrinsic reward struggles to outperform DreamerV2, which is because Hopper Hop is more difficult in terms of exploration. We also observe that APV without pre-training cannot also outperform DreamerV2 before 300K environment steps, since the model needs a large amount of samples to learn representations that are useful for capturing the dynamics information of the environment. By utilizing the dynamics information captured in the pre-trained representations for exploration, APV with both pre-training and intrinsic bonus performs best from the initial phase of fine-tuning.

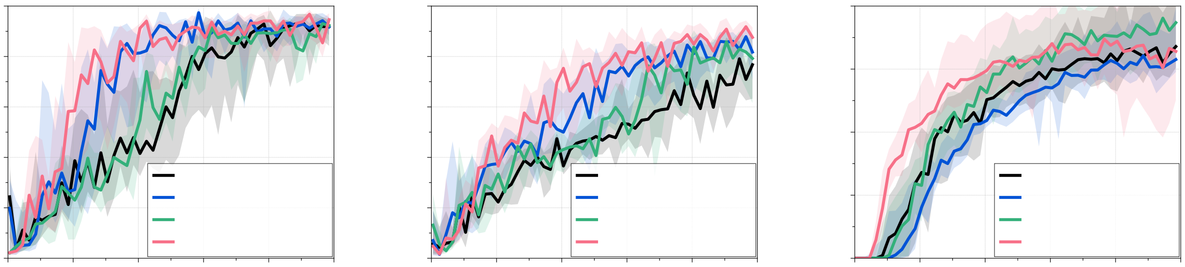

Figure 14. Learning curves of APV on locomotion tasks from DeepMind Control Suite as measured on the episode return. The solid line and shaded regions represent the interquartile mean and bootstrap confidence intervals, respectively, across eight runs.