Article

Visual Pretraining via Contrastive Predictive Model for Pixel-Based Reinforcement Learning

Tung M. Luu, Thang Vu, Thanh Nguyen and Chang D. Yoo *

School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea

* Correspondence: cd_yoo@kaist.ac.kr

Abstract: In an attempt to overcome the limitations of reward-driven representation learning in vision-based reinforcement learning (RL), an unsupervised learning framework referred to as the visual pretraining via contrastive predictive model (VPCPM) is proposed to learn the representations detached from the policy learning. Our method enables the convolutional encoder to perceive the underlying dynamics through a pair of forward and inverse models under the supervision of the contrastive loss, thus resulting in better representations. In experiments with a diverse set of vision control tasks, by initializing the encoders with VPCPM, the performance of state-of-the-art vision-based RL algorithms is significantly boosted, with 44% and 10% improvement for RAD and DrQ at 100 steps, respectively. In comparison to the prior unsupervised methods, the performance of VPCPM matches or outperforms all the baselines. We further demonstrate that the learned representations successfully generalize to the new tasks that share a similar observation and action space.

Keywords: representation learning; vision-based deep reinforcement learning; deep reinforcement learning; sample efficiency

1. Introduction

Recent advances in deep reinforcement learning (RL) have allowed agents to perform complex control tasks directly from raw sensory observations. Remarkable successes have been achieved, ranging from learning to play video games from raw pixels [1], solving complex tasks from first-person-view observations [2,3], to autonomously performing robotic tasks [4–7].

As a standard practice, deep RL methods jointly learn a visual encoder and a policy in an end-to-end manner. In this paradigm, the visual representations are learned under the strong supervision of task-specific rewards. While the simplicity of end-to-end methods is appealing, learning the representations relying on the rewards can have several limits. First, the representations are acquired harder under sparse rewards, thus requiring more data for convergence. Additionally, in practice, the reward function is commonly designed and retested until a suitable function is selected, resulting in the representation learning process being repeated, thus it is inefficient. Furthermore, whenever the agent encounters the new tasks, the representation learning is performed again even if the environment appearance is identical across tasks.

In this work, we pursue an alternative paradigm as depicted in Figure 1, wherein the visual encoder is first pretrained without any reward supervision, detached from the policy learning, and then the learned task-agnostic representations are transferred for learning policy on a specific task, under the reward supervision. This two-stage training enables the representations learned in an environment to be reused for other tasks that use the same environment, such as the cheetah environment, containing moving forward and backward tasks. Furthermore, the policy learning stage also requires fewer samples for training, and is thus more sample efficient.
In the first stage, a natural choice is pretraining the visual encoder on a large dataset such as ImageNet. However, previous works show that naively pretraining in such a dataset does not lead to a significant impact [8–10]. This ineffectiveness might stem from the domain discrepancy, and more importantly, the pretraining data is lacking in reflecting natural dynamics relation, i.e., the Markov decision process (MDP) property, which is inherent in RL tasks. As we show later in our experiments, pretraining the visual encoder with even in-domain data without considering dynamics is still underperformed. Early works [11–13] commonly learn the compact representation by reconstructing the pixels in the current or subsequent frame, which is very challenging with high-dimensional observations. Recently, ref. [14] introduced the contrastive prediction to bypass the use of reconstruction-based prediction. However, this method is lacking in capturing the environment dynamics, thus resulting in low performance in complex tasks such as Cheetah or Walker. To mitigate this shortcoming, ref. [15] introduced augmented temporal contrast (ATC), which additionally trains a marginal forward dynamic model as an auxiliary task for learning representation. Despite effectiveness, the proposed marginal forward model does not condition actions while learning, thus making ATC focus on temporal coherence between observations rather than modeling environment dynamics. To address these issues, this paper introduces a method referred to as the visual pretraining via contrastive predictive model (VPCPM). Specifically, VPCPM utilizes the reward-free data from the environment to learn the visual encoder by jointly optimizing the forward and inverse dynamic models through the contrastive objective. The visual encoder firstly maps the observation into the latent state. Given a latent state and a latent action, the forward model predicts the next latent state. Meanwhile, the inverse model infers the executed action given the consecutive latent states. Learning the forward and inverse models serves as the constraint to enable the visual encoder to consistently follow the underlying dynamics of the environment.

The proposed method is evaluated on a diverse set of image-based continuous control tasks from DeepMind Control Suite [16]. The experiment results show that VPCPM consistently improves the performance and sample efficiency of the state-of-the-art vision-based RL algorithms. Specifically, at 100k environment step, VPCPM improves the mean returns 44% over reinforcement learning with augmented data (RAD) [17], 10% over data-regularized Q (DrQ) [18]. Compared to prior unsupervised pretraining methods, VPCPM matches or outperforms all the baselines as tested across all environments. Moreover, the investigation of unseen tasks shows that the VPCPM-initialized encoder successfully generalizes to unseen tasks that operate in the same environment.

The rest of the paper is organized as follows: Section 2 discusses related works, while Section 3 describes setup in vision-based RL and the related base algorithm. Section 4 details our proposed method. Section 5 presents extensive experiments of the proposed methods, while the conclusion of our paper is presented in Section 6.
2. Related Work

**Vision-based Deep Learning.** Since the breakthrough results of convolutional neural network (CNN) [19] in the 2012 ILSVRC image classification task, the deep learning approaches have dominated in almost every recognition and detection task. Many works have already implemented the vision-based deep learning methods to detect the number and location of occupants [20–22], perform instance segmentation from RGB images [23–25] or point cloud [26–28], and reconstruct 3D scenes [29–31]. In robotics control, by leveraging the expressive CNN together with the advanced reinforcement learning methods, there are many works that successfully perform the complex manipulation tasks based on visual perception [6,7,32,33]. Despite these successes, there are still challenges lying ahead in achieving versatile, robust, and generalized representations.

**Representation Learning for RL.** Several methods have been proposed in the literature to learn better representations for vision-based RL. Ref. [2] proposed auxiliary tasks including next-observation and reward-based prediction in improving the performance on DMLab [34]. Representation learning using the reconstruction-based task is also a common method, where a beta variation autoencoder (β-VAE) [35] or a deterministic AE is applied for learning a low-dimensional representation, and subsequently, a policy was learned using this representation. Recent work [36] attempted to jointly learn VAE/AE together with RL objective, while [12] proposed to train the two objectives one after the other in an alternating fashion. Ref. [13] also used the VAE for learning the representation in the goal-conditioned setting. These methods only focus on learning the representation of an observation rather than modeling the environment dynamics. Moreover, the use of reconstruction-based tasks makes algorithms hard to scale in high-dimensional environments.

Leveraging the predictive model, which is the combination of the forward and inverse dynamic models, for learning representation is also commonly used. Ref. [37] constructed a joint inverse–forward model to learn representation, in which the forward model is used as a regularizer for training the inverse model features. Ref. [38] used only the inverse model to learn the representations, and then the learned representations were used for the forward model to compute intrinsic reward. Ref. [39] learned the representation by combining the forward and inverse models using cycle consistency losses. In this approach, the use of the forward model makes the representations more Markovian [40,41], i.e., the next latent state only depends on the current latent state and action. In the meanwhile, the inverse model encourages the encoder to capture the controllable features. However, this also makes the representation ignore the uncontrollable features that cannot be captured by the inverse model, which might be useful for solving tasks. Our method proposes the use of contrastive loss while learning the forward model to prevent this degeneracy.

Several methods leverage the recent advances in self-supervised learning for acquiring the representation [42–46] in order to improve the performance of the base RL algorithms. Ref. [42] applied a variant of the noise-contrastive estimation loss on future steps on top of A2C [47] to learn better representation and improve sample efficiency, although the results were limited. In the Atari [48] domain, ref. [49] introduced a new contrastive loss to learn better representation in a fully unsupervised setting, while in [45], the authors adapted the BYOL objective [50] for learning the visual encoder in parallel with the RL objective, which led to state-of-the-art results on the Atari benchmark. In the vision-based continuous control domain, ref. [14] introduced a general framework that is a combination of contrastive loss with image augmentation for learning the representation in a more sample efficient way. Ref. [46] proposed a contrastive version of the conditional entropy bottleneck objective [51] to learn a compressed representation of the predictive information of the environment dynamics. Furthermore, to learn the robustness representation, ref. [52,53] attempted to embed the bisimulation metric [54] into the latent space to encourage the representations invariant to the distractors and generalized to unseen environments. Recently, refs. [17,18] showed that a modest degree of image augmentation can significantly improve sample efficiency for learning directly from visual observation. Most of the existing methods learn the visual encoder by optimizing the unsupervised/self-supervised auxiliary and
the RL objective simultaneously, which can be considered as learning from scratch, in an end-to-end manner. By contrast, our framework considers training in two stages, and we further propose a new self-supervised auxiliary objective in the pretraining stage to provide a meaningful representation for directly using or further fine-tuning in the testing environment.

**Unsupervised Pretraining Representation for RL.** Pretraining the representations without supervision using an unsupervised or self-supervised framework is a common practice in other fields, such as natural language processing (NLP) [55–57] or computer vision [42–44,50,58,59]. These studies consider effective ways to learn the visual encoder from massive unlabeled data and reduce sample complexity when learning a new task. In deep RL, early studies attempted to pretrain the visual encoder by using pixel-reconstruction task [11,12] or object detection task [60]; the pretrained representations are then fine-tuned on a specific task. Some recent studies [8–10] showed that naively pretraining on ImageNet is not helpful for performing the downstream RL tasks. It hints that the visual encoder of the RL agent would be pretrained on the data closely related to its environments. Along this line of research, ref. [14] attempted to use their proposed objective, originally designated for online training, to learn the visual encoder detached from policy learning. However, due to the lack of encoding of the environment dynamics, the performance was limited in complex tasks such as Cheetah. The study closest to ours is [15], which also attempted to learn the encoder detached from the RL objective. However, this method implicitly estimates the next observation by marginalizing over actions. In contrast, our method relies on the forward and inverse dynamics models with contrastive loss, where the proposed forward model is conditioned on action such that in the latent space, the representation of states and action consistently follow the underlying Markov decision process of the environment.

In addition to pretraining the visual encoder, there is also another line of research that tries to pretrain the policy with self-supervised intrinsic rewards [38,61–65]. In this setting, the agent is first allowed to freely interact with the environment for a long period without access to extrinsic rewards, then it is exposed to task-specific rewards to learn on downstream tasks. The intrinsic reward is commonly formulated to encourage the RL agent to gain new knowledge about the environment [38,61,62], maximize diversity of collected data [9,66], or learn diverse skills [67–70]. For the vision-based RL tasks that are applicable in this setting, the visual encoder is concurrently trained with the policy during pretraining. Recently, ref. [9] proposed to use a particle-based estimator [71] to estimate entropy for observations, with its representations learned by using contrastive loss from SimCLR [44]. Alternatively, ref. [66] proposed a self-supervised pretraining scheme that allows detaching the representation learning from exploration (i.e., learning from intrinsic rewards) to enable the generalization of representations for unseen tasks. In this method, the representations are learned by a variant of clustering-based contrastive loss SwAV [72]. These works are promising to acquire the general policy as well as the generalized representations. However, they are still required to freely interact with the environment during pretraining, which is potentially unsafe in the real world. In contrast, our framework allows us to learn the encoder entirely from an offline dataset, thus resulting as safer and enabling the reuse of past data.

3. Background

In this section, the framework for vision-based reinforcement learning is presented together with a representative off-policy model-free algorithm, soft actor-critic.

3.1. Reinforcement Learning from Images

The problem of solving control task from high-dimensional observations is formulated as a partially observable Markov decision process (POMDP) [73,74], which can be 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, the transition dynamics \(p = \Pr(o_{t+1} | o_{\leq t}, a_t)\) represent the probability distribution over the next observation \(o_{t+1}\) given the history of previous observations \(o_{\leq t}\) and current
action $a_t$, the reward function $r : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ that maps the current observation and action to a reward $r_t = r(o_t, a_t)$, and $\gamma \in [0, 1]$ is a discount factor. Following common practice [1], the POMDP is reformulated as an MDP [73] by stacking consecutive observations into a state $s_t = \{o_t, o_{t-1}, o_{t-2}, \ldots\}$. For simplicity of notation, the transition dynamics and reward function are redefined as $p = Pr(s_{t+1}|s_t, a_t)$ and $r_t = r(s_t, a_t)$, respectively. The goal of RL is to find a policy $\pi(a_t|s_t)$ that maximizes the expected return defined as the total of accumulated reward $\mathbb{E}_p[\sum_{t=0}^{T} \gamma^t r_t|a_t \sim \pi(.)|s_t, s_{t+1} \sim p(., s_t, a_t), s_0 \sim p_0(.)]$, where $T$ is the length of episode and $p_0$ is the probability distribution of initial state.

### 3.2. Soft Actor-Critic

Soft actor-critic (SAC) [75] is an off-policy actor-critic method based on the maximum entropy RL framework [76], which encourages the exploration and robustness to noise by maximizing a weighted objective of the reward and the policy entropy. To update the parameters, SAC performs the soft policy evaluation and improvement steps. The soft policy evaluation step fits a parametric Q-function $Q(s_t, a_t)$ using transitions from the replay buffer $\mathcal{D}$ by minimizing the soft Bellman residual:

$$J(Q) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \left[ (Q(s_t, a_t) - r_t - \gamma \bar{V}(s_{t+1}))^2 \right].$$  \hspace{1cm} (1)

The target value function $\bar{V}$ is approximated via a Monte Carlo estimate of the following expectation:

$$\bar{V}(s_t) = \mathbb{E}_{a_t \sim \pi}[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t)],$$  \hspace{1cm} (2)

where $\bar{Q}$ is the target Q-function, with its parameters obtained from an exponentially moving average of the Q-function parameters for stabilizing training. The soft policy improvement step then updates the following stochastic policy $\pi$ by minimizing the following objective:

$$J(\pi) = \mathbb{E}_{s_t \sim \mathcal{D}}[\mathbb{E}_{a_t \sim \pi}[\alpha \log \pi(a_t|s_t) - Q(s_t, a_t)]].$$ \hspace{1cm} (3)

In this work, the learnable version of temperature $\alpha$ is used instead of the pre-fixed value, which is optimized by the following objective:

$$J(\alpha) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}}[-\alpha \log \pi(a_t|s_t) - \alpha \bar{H}],$$ \hspace{1cm} (4)

where $\bar{H} \in \mathbb{R}$ is the target entropy hyperparameter that policy attempts to match, which in practice is usually set equal to $-|\mathcal{A}|$.

SAC is one of the state-of-the-art RL algorithms for continuous control [75]. It is also widely used as a backbone for solving vision-based control tasks [5,14,17,18,36,46]. In this work, we adopt RAD [17] and DrQ [18], which are built on top of SAC, for policy learning.

### 4. Method

In this section, the proposed visual pretraining via contrastive predictive model (VPCPM) is described. The proposed method can be used to pretrain the visual encoder, which is then utilized for policy learning by common model-free vision-based RL algorithms.

#### 4.1. Network Architecture

The control policy network takes the input as the state $s_t$ and outputs the action $a_t$. It consists of the visual encoder $\pi_e$ parameterized by $\phi$ and the policy $\pi_a$ parameterized by $\theta_a$, as depicted in Figure 1. This design enables the encoder to be trained independently with the policy part, i.e., without requiring the RL objective for training. The goal of the proposed method is to learn useful representations from an amount of given data without rewards, such that $\pi_a$ can be efficiently trained on top of that to solve RL tasks.
4.2. Visual Pretraining via Contrastive Predictive Model

VPCPM introduces a useful prior for the vision-based RL training procedure by enforcing the representations not only representing the semantic information but also conforming to the basis of dynamics from the environment. During the pretraining stage, for a given environment, it is assumed that there is a pre-collected dataset $D$ consisting of $N$ transitions without task-specific rewards: $(s_{i}^{(l)}, a_{i}^{(l)}, s_{i+1}^{(l)})$ with the index $i = \{1, \ldots, N\}$. The visual encoder is desired to effectively encode the semantics and consistently follow dynamics only from the primitive elements, i.e., observations and actions.

An overview of the proposed method is shown in Figure 2. The visual encoder $\pi_e: \mathcal{O} \rightarrow \mathcal{Z}$ learns the mapping from the observation into the latent space. VPCPM alternates between learning the forward dynamic model (forward step) and inverse dynamics model (inverse step) while optimizing the underlying encoder $\pi_e$ (Algorithm 1). At the forward step, the forward model $F$ parameterized by $\psi$ takes the inputs as the current latent state and the latent action in predicting the next latent state. To optimize $F$ together with $\pi_e$, the InfoNCE loss [42] is employed, which contrasts between the predicted next latent state and the ground truth. Formally, let $f: \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$ be a similarity metric; the objective of forward model is described as:

$$J_F(\psi, \phi) = -\frac{1}{K} \sum_i \log \frac{\exp f(\hat{z}_{i+1}^{(l)}, \bar{z}_{i+1}^{(l)})}{\frac{1}{K} \sum_j \exp f(\hat{z}_{i+1}^{(l)}, \bar{z}_{j+1}^{(l)})},$$

where $\hat{z}_{i+1}^{(l)} = F(\pi_e(s_i^{(l)}), \bar{a}_i^{(l)}; \psi)$ is the predicted latent state, and $\bar{z}_{i+1}^{(l)} = \pi_e(s_{i+1}^{(l)})$ is the target latent state without encoder parameter update. The expectation is computed over $K$ samples of $(s_t, a_t, s_{t+1})$. Operating in the latent space bypasses the prediction of the forward model in pixel space, which would be extremely challenging given the large uncertainty in pixel prediction. The use of InfoNCE helps to learn the discriminative representation, where the dissimilar states are repelled and the similar states are pulled close. Additionally, this objective also prevents trivial collapsed solutions in which the constant features are obtained for every state.

![Figure 2. VPCPM for the encoder $\pi_e$: At the forward step, the forward model $F$ takes the inputs as the latent state $z_t$ and the latent action $\hat{a}_t$ for predicting next latent state $z_{t+1}$. In this step, both $F$ and $\pi_e$ are optimized together. At the inverse step, the inverse model $I$ takes the inputs as two latent states $z_t$ and $z_{t+1}$ for predicting the action $a_t$. In this step, both $I$ and $\pi_e$ are optimized together.](image)
Algorithm 1 Visual pretraining via contrastive predictive model (VPCPM)

1: **Input**: Dataset \( D = \{ (s_i^t, a_i^t, s_{i+1}^t) \}_{i=1}^{N} \), the encoder’s parameters \( \phi \), batch size \( K \)
2: **Output**: The encoder’s parameters \( \phi \)
3: **Initialize**: The parameters \( \phi, \psi, \rho \)
4: for \( k = 1 \) to \( \infty \) do
5: Sample a batch \( B = \{ (s_i^t, a_i^t, s_{i+1}^t) \}_{i=1}^{K} \)
6: Compute the latent states:
   \[
   z_i^t = \pi_e(AUG(s_i^t)) \quad z_{i+1}^t = \pi_e(AUG(s_{i+1}^t))
   \]
7: Train forward model: \( J_f(\psi, \phi) \) \( \triangleright \) Equation (5)
8: Train inverse model: \( J_i(\rho, \phi) \) \( \triangleright \) Equation (6)
9: end for

In the inverse step, the inverse model \( I \) parameterized by \( \rho \) takes the states before and after transition and predicts the action in between. In this work, the inverse model operates in the latent space extracted by the visual encoder \( \pi_e \). The encoder is jointly optimized with the inverse model by minimizing the following objective:

\[
J_i(\rho, \phi) = \frac{1}{K} \sum_i \ell(I(\pi_e(s_i^t), \pi_e(s_{i+1}^t); \rho), a_i^t).
\]  \( \triangleright \) Equation (6)

For continuous action, \( \ell \) can be defined as the mean squared error or mean absolute error between ground truth and the predicted action. When predicting the action, the inverse model pays attention to the controllable features and the temporal difference between consecutive states in latent space, which also encourages the encoder to capture the discriminative features. Pretraining the representations satisfying both forward and inverse dynamics models strengthens the relations of state and actions in latent space, establishing an initialization point of the encoder for further fine-tuning inside a region of parameter space in which the parameters are henceforth restricted.

5. Experimental Results
5.1. Experiment Setup

The proposed method is evaluated on a diverse set of the image-based control tasks from DeepMind Control suite [16], which was recently considered as a standard for benchmarking the sample efficiency of RL from images [14,17,77,78]. DMControl consists of different robot models (environments), where each model can be associated with a particular MDP representing a specific task. The selected benchmark includes six environments from the PlaNet benchmark [77], as shown in Figure 3, where the action repeat is treated as a hyperparameter (Table 1). The settings of visual observation follow [14,17,18,36], which consider a stack of three consecutive 84 \( \times \) 84 RGB renderings as a state.

Figure 3. VPCPM is benchmarked on six image-based control environments from the DeepMind Control Suite [16]. The order of environment from lowest to the highest dimension of action: Cartpole, Ball in cup, Reacher, Finger, Cheetah, and Walker. Each task offers a unique set of challenges, including complex dynamics, sparse rewards, and hard exploration.
Table 1. Hyperparameters for training the vision-based RL algorithms: RAD [17] and DrQ [18].

| Parameter                  | RAD [17] | DrQ [18] |
|----------------------------|----------|----------|
| Action repeat              | 2 Finger, spin; Walker, walk 8 Cartpole, swingup 4 otherwise |
| Batch size                 | 512      | 512      |
| Augmentation               | Crop [17] | Crop [18] |
| Replay buffer size         | $10^6$   | $10^5$   |
| Initial steps              | 1000     | 1000     |
| Stacked frames             | 3        | 3        |
| Discount $\gamma$          | 0.99     | 0.99     |
| Optimizer                  | Adam     | Adam     |
| Learning rate ($\pi_\text{e}$, $\pi_\text{a}$, $Q$) | $2 \times 10^{-4}$ cheetah $1 \times 10^{-3}$ (all) $1 \times 10^{-3}$ otherwise |
| Learning rate ($\alpha$)   | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Critic target update frequency | 2       | 2       |
| Critic soft-update rate $\tau$ | 0.01   | 0.01   |
| Actor update frequency     | 2        | 2        |
| Actor log stddev bounds    | $[-10, 2]$ | $[-10, 2]$ |
| Initial temperature        | 0.1      | 0.1      |

For the vision-based RL algorithm, we use two state-of-the-art methods including RAD [17] and DrQ [18], which are based on soft-actor critic (SAC) [75]. The network architecture used is identical to [17,18,36]. Unless stated otherwise, the configurations of the algorithm are as follows: the actor and critic neural network are trained using Adam optimizer [35] and a mini-batch size of 512. For SAC, the initial temperature is 0.1, the soft target update rate $\tau$ is 0.01, and the target network and the actor updates are performed every two critic updates similar [17,18,36]. The random cropping [14,17] is used as image augmentation during pretraining. The learning rate for actor, critic, and the parameter $\alpha$ of RAD and DrQ is followed by the setup from each method. In our CPM, the forward model is parameterized by four 50-d hidden layers followed by ReLU activation except the last, and the inverse model is parameterized by three 1024-d hidden layers followed by ReLUs, except the last layer, which uses tanh to normalize the actions. The action is encoded by an MLP consisting of two 50-d hidden layers followed by ReLUs except for the last. Input to the forward model MLP is a concatenation of the current latent state and the current encoded action. Input to the inverse model MLP is a concatenation of the current and next latent states. Both the forward and inverse model have separated Adam optimizer [79]. During the pretraining stage, the encoder, inverse, and forward model are trained with a learning rate of $1 \times 10^{-4}$ for Walker, $2 \times 10^{-4}$ for Cheetah, and $1 \times 10^{-3}$ otherwise, with the batch size of 512.

The performance of the agent is evaluated across five seeds; for every seed, the average returns of 10 episodes are computed every 10k environment steps. The figures plot the mean performance together with ±1 standard deviation shading. The performance is reported over the true environment steps as a common practice [14,17,18,36,78], thus are invariant to the action repeat hyperparameter. Throughout experiments, pretraining data are collected by a random policy. Specifically, for Cheetah and Walker domain, 50k transitions are collected, and 25k for the others. The encoder is pretrained within 50k iterations for Cheetah and Walker, and 25k otherwise, which corresponds to one update step per transition. The full set of parameters is shown in Table 1.

5.2. Effects of Pretrained Representation

In this section, the effectiveness of VPCPM in pretraining the visual encoder for different vision-based algorithms is investigated. Two state-of-the-art algorithms, including RAD [17] and DrQ [18], are evaluated, where the hyperparameters reported in each method are used. The random crop augmentation from each method is applied. Crop image
augmentation in RAD is formed by cropping $84 \times 84$ frames from an input frame of $100 \times 100$, while in DrQ, the $84 \times 84$ frames are padded each side by $\pm 4$, then cropped back to the $84 \times 84$ size. Moreover, in the RAD paper, there are some tasks using translation augmentation; we instead use crop augmentation across tasks, thus the results may be varied. The parameters of the pretrained encoder are fine-tuned by the base RL algorithm in a specific task.

Figure 4 and Table 2 compare these methods with and without pretraining. We provide the result at both 100k and 500k steps as common report for DMControl [14,17,18,46]. At 100k step, VPCPM enhances over RAD ranging from 13 to 118%, with the largest magnitude on Ball in cup-catch. For DrQ, the enhancements are in the range from 6 to 20%, with the largest magnitude on Reacher-easy. From Figure 4, the improvement is shown clearer at the early stage of training on the sparse reward tasks such as Ball in cup-catch and Finger-spin. The reason is that these tasks usually failed to complete the task at the beginning, thus observing less reward signal to learn the visual representation. With our VPCPM-initialized representations, the policy part can be quickly learned, resulting in significantly accelerating learning progress. For the Cartpole-swingup task, the action space is very small (with the dimension of one); thus, with the well-presented representation of states, the task can be quickly solved. The tasks including Reacher-easy, Walker-walk, and Cheetah-run are more challenging because of the exploration problem, thus requiring more samples to complete the tasks even with good representations, resulting in a lower magnitude of the improvement. Overall, our method improves 44% over RAD and 10% over DrQ at 100k steps. At 500k steps, the enhancement is smaller, with 3.4% over RAD and 2.9% over DrQ. This is because the base algorithms are almost converged around 500k; thus, the effect of pretraining is moderate.

**Figure 4.** The performance on six tasks from PlaNet benchmark [77]. Pretraining the visual encoder by VPCPM consistently improves performance and sample efficiency across all environments.
Table 2. Episodic return at 100k and 500k environment steps with mean and standard deviation for 5 seeds. The scores are bold, indicating the improvement compared to the corresponding base RL algorithm.

|                  | RAD [17] | DrQ [18] |
|------------------|----------|----------|
|                  | Scratch  | +VPCPM   | Scratch | +VPCPM   |
| Finger, spin     | 860 ± 29 | 974 ± 15 | 901 ± 104 | 958 ± 20 |
| Cartpole, swingup| 454 ± 155| 798 ± 42 | 759 ± 92  | 825 ± 40 |
| Reacher, easy    | 704 ± 212| 855 ± 117| 601 ± 213 | 720 ± 65 |
| Cheetah, run     | 365 ± 31 | 420 ± 81 | 344 ± 67  | 384 ± 45 |
| Walker, walk     | 493 ± 175| 614 ± 128| 612 ± 164 | 651 ± 115|
| Ball in cup, catch| 421 ± 247| 918 ± 25 | 913 ± 53  | 959 ± 7  |

500k steps

|                  | 100k Steps | 500k Steps |
|------------------|------------|------------|
| Finger, spin     | 982 ± 2    | 985 ± 15   | 938 ± 103  | 988 ± 5  |
| Cartpole, swingup| 867 ± 10   | 870 ± 13   | 868 ± 10   | 877 ± 11 |
| Reacher, easy    | 945 ± 34   | 965 ± 41   | 942 ± 71   | 945 ± 43 |
| Cheetah, run     | 573 ± 35   | 670 ± 7    | 660 ± 96   | 698 ± 47 |
| Walker, walk     | 948 ± 11   | 952 ± 15   | 921 ± 45   | 960 ± 10 |
| Ball in cup, catch| 962 ± 5   | 964 ± 13   | 963 ± 9    | 968 ± 4  |

5.3. Comparison with Prior Methods

In this section, the comparison of pretraining by using different unsupervised learning methods is conducted. VPCPM is compared against two representation learning approaches: non-model-based and model-based. The non-model-based representation learning approach includes (i) reconstruction loss as in VAE [35], and (ii) the contrastive loss from single observation as in CURL [14]. The model-based approach includes (iii) a simple predictive model (PM), where the forward and inverse model are learned by mean square error loss, (iv) augmented temporal contrast (ATC) [15], where the marginal forward model, i.e., without conditioning on actions, is parameterized by a residual network and learned using contrastive loss, and (v) predictive coding-consistency-curvature (PC3) [80], where the forward model is learned by the weighted sum of three losses: contrastive loss, mean square error loss, and low curvature loss. In PC3, the current latent state-action pair \((z_t, a_t)\) is used as the source of negative samples and used for the contrastive prediction of the next latent state \(z_{t+1}\). In contrast, we use the predicted next latent state \(\hat{z}_{t+1}\) as the source of negative samples. For the implementation of ATC, we use our implementation with the modification as follows, disable the inverse model, and remove the action input of the forward model. For PC3, we use the author’s provided code (https://github.com/VinAIResearch/PC3-pytorch.git, accessed on 17 March 2022). For a fair comparison, the same amount of samples is used during pretraining. The procedure for evaluation is similar to the previous section, but only the RAD algorithm is considered.

The performance in Figure 5 shows that RAD initialized by VPCPM outperforms across all environments. These improvements suggest the importance of imposing the dynamics to the visual encoder during pretraining, which is lacking in the methods that only focus on semantic information such as reconstruction and contrastive. In comparison to the simple PM, the proposed method benefits from the contrastive objective. Indeed, the inverse model is limited since it cannot capture the changes in the sensory stream beyond the agent’s control, and the use of contrastive is helpful to prevent this degeneracy. Moreover, learning in a contrastive manner represents states more discriminative in the latent space. Compared with ATC, VPCPM shows the importance of the action-condition forward model together with the inverse model in learning the controllable features. PC3 is originally designed to use for model-based planning algorithms such as iLQR, which requires the system to be locally linear. Thus, the features learned by PC3 might not be suitable for vision-based RL algorithms in the highly nonlinear system, as in our considered environments. Indeed, the results show that the representations from VPCPM are more...
useful for vision-based RL algorithms. Overall, the sample efficiency in deep RL should be attained from representations that are discriminative and follow the dynamics.

5.4. Effects of Components during Pretraining

Ablation tests were performed to determine the effects of individual components in VPCPM. The performances of RAD with the encoder pretrained by using the contrastive forward dynamic model (cFDM), inverse dynamic model (IDM), and both of them (VPCPM) are shown in Figure 6. Overall, the base RL agent benefits from pretraining by any type of dynamic model, but the cFDM shows more impacts. Together with the constraint of IDM, the proposed method significantly accelerates the sample efficiency of the base algorithm. Moreover, training the visual encoder together with FDM purely in latent space does not suffer from the collapsed problem, where the encoder outputs a constant across states.

5.5. Generalization over Unseen Tasks

In this section, the generalization of the pretrained encoder for unseen tasks is examined. Specifically, the encoder pretrained from a source task is used for the unseen target tasks. Subsequently, the RAD agent is trained on top of the pretrained encoder until converged. The considered tasks are shown in Table 3. The target tasks are different on the
reward function but share the same observation space, except for Reacher-hard, where the size of the visual indicator is different (see Figure 7). The performance of the base agent is evaluated in both “fine-tuning” and “frozen” settings, where the pretrained representation is frozen or fine-tuned. The results are averaged across five seeds and compared against the agent learning from scratch.

Table 3. The encoder pretrained with data from a source task is reused for new target tasks. Almost the encoders are successfully generalized to target tasks, except for Reacher-hard, needing to fine-tune.

| Source Task | Target Task                  |
|-------------|------------------------------|
| Cartpole-swingup | Cartpole-swingup_sparse      |
| Walker-walk | Cartpole-balance            |
| Reacher-easy | Cartpole-balance_sparse      |
|              | Walker-stand                |
|              | Reacher-hard                |

Figure 7. An example of observation from Reacher-easy (left) and Reacher-hard (right). The different size of the goal indicator causes the degraded performance when freezing the encoder. However, fine-tuning from this encoder still shows higher performance compared to training from scratch.

The results are shown in Figure 8. In almost all tasks, the frozen representation is sufficient in learning optimal policy. When fine-tuning from the pretrained initialization, the performance is slightly improved. The major exception is the Reacher-hard task, where the frozen encoder significantly underperforms. However, the fine-tuning encoder shows more sample efficiency than that learning from scratch. The difference in the observation space causes this downgraded performance, i.e., the different size of the target indicator. The enhancement in the performance of the base RL agent shows that VPCPM is successfully learning the abstract features without reward supervision.

Figure 8. Performance on unseen tasks comes from the same domain as the source task. The pretrained representation enables the RL agent to significantly improve sample efficiency, especially in sparse reward tasks.
5.6. Pretraining with Classification

To show the importance of imposing the dynamics in representation learning, we investigate the case where the visual encoder is trained to capture the semantic only, without knowing about the dynamics. To investigate that, we consider six-way classification which corresponds to six robotic models from DeepMind Control Suite [16], as indicated in Figure 3. The dataset is generated by an expert policy. In each class, the training and test sets contain 50k and 10k samples, respectively. In total, there are 300k samples for training and 60k samples for testing. The visual encoder and the classifier are trained using Adam optimizer [79] with the learning rate $3 \times 10^{-4}$, and $\beta = (0.9, 0.999)$. We use the data augmentation methods used in [43] and the random crop [17]. The pretrained encoder is then frozen and used for policy learning. The results are shown in Figure 9. In Ball in cup-catch and Cartpole-swingup, pretraining by classification slightly improves performance, while other tasks show no gain, or even degraded performance. The results indicate the importance of making the representation encode dynamic information for learning representation offline for RL tasks.

![Figure 9](image-url)  
Figure 9. The comparison of training from scratch, pretraining by classification, and VPCPM. The results show that the classification task does not lead to improved sample efficiency.

6. Conclusions

In this paper, a new self-supervised representation learning method is proposed to pretrain the visual encoder for the vision-based RL. By leveraging plenty of reward-free data, the proposed method successfully learns the meaningful initial representations that provide sufficient information and consistently follow underlying dynamics from an environment. Experimental results show that the state-of-the-art vision-based RL algorithms benefit from our method, with the gain of 44% over RAD and 10% over DrQ at 100k steps. Additionally, we benchmark several leading self-supervised methods for pretraining visual encoders. The results show that the performance of the policy learned on top of the VPCPM-trained encoder matches or outperforms all others. Furthermore, the independence of task-specific rewards during pretraining allows our learned representations to be reused for different tasks sharing similar observation and action space.

In this paper, we investigate the effectiveness of pretraining the visual encoder, in which the testing and training environments are similar. However, this condition is brittle in practice. Future works should improve the robustness of the pretrained representation such that it is invariant to the visual distractions from the environment such as variations in background, color, and camera pose.

Author Contributions: Conceptualization, T.M.L. and C.D.Y.; methodology, T.M.L. and C.D.Y.; software, T.M.L. and T.V.; validation, T.M.L., T.V. and T.N.; formal analysis, T.M.L., T.V., T.N. and C.D.Y.; investigation, T.M.L., T.V. and T.N.; resources, C.D.Y.; data curation, T.M.L.; writing—original
References

1. 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, 518, 529–533. [CrossRef] [PubMed]

2. Jaderberg, M.; Mnih, V.; Czarnecki, W.M.; Schaul, T.; Leibo, J.Z.; Silver, D.; Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.

3. Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, 10–15 July 2018; pp. 1407–1416.

4. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* 2016, 17, 1334–1373.

5. Lee, A.X.; Nagabandi, A.; Abbeel, P.; Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In Proceedings of the 34th Advances in Neural Information Processing Systems, NeurIPS 2020, Online, 6–12 December 2020; pp. 741–752.

6. Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In Proceedings of the 2nd Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29–31 October 2018; pp. 651–673.

7. Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; Ribas, R.; et al. Solving rubik’s cube with a robot hand. *arXiv* 2019, arXiv:1910.07113.

8. Julian, R.; Swanson, B.; Sukhatme, G.S.; Levine, S.; Finn, C.; Hausman, K. Never Stop Learning: The Effectiveness of Fine-Tuning in Robotic Reinforcement Learning. In Proceedings of the 4th Conference on Robot Learning, CoRL 2020, Online, 16–18 November 2020.

9. Liu, H.; Abbeel, P. Behavior from the void: Unsupervised active pre-training. In Proceedings of the 35th Advances in Neural Information Processing Systems, NeurIPS 2021, Online, 6–14 December 2021, pp. 18459–18473.

10. Shah, R.; Kumar, V. Rrl: Resnet as representation for reinforcement learning. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Online, 18–24 July 2021.

11. Lange, S.; Riedmiller, M. Deep auto-encoder neural networks in reinforcement learning. In Proceedings of the International Joint Conference on Neural Networks, IJCNN 2010, Barcelona, Spain, 18–23 July 2010; pp. 1–8.

12. Finn, C.; Tan, X.Y.; Duan, Y.; Darrell, T.; Levine, S.; Abbeel, P. Deep spatial autoencoders for visuomotor learning. In Proceedings of the International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, 16–21 May 2016.

13. Nair, A.V.; Pong, V.; Dalal, M.; Bahl, S.; Lin, S.; Levine, S. Visual reinforcement learning with imagined goals. In Proceedings of the 32th Advances in Neural Information Processing Systems, NeurIPS 2018, Montreal, QC, Canada, 3–8 December 2018.

14. Srinivas, A.; Laskin, M.; Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Online, 12–18 July 2020.

15. Stooke, A.; Lee, K.; Abbeel, P.; Laskin, M. Decoupling representation learning from reinforcement learning. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Online, 18–24 July 2021; pp. 9870–9879.

16. Tunyasuvunakool, S.; Muldal, A.; Doron, Y.; Liu, S.; Bohez, S.; Merel, J.; Erez, T.; Lillicrap, T.; Heess, N.; Tassa, Y. dm_control: Software and tasks for continuous control. *Softw. Impacts* 2020, 6, 100022. [CrossRef]

17. Laskin, M.; Lee, K.; Stooke, A.; Pinto, L.; Abbeel, P.; Srinivas, A. Reinforcement Learning with Augmented Data. In Proceedings of the 34th Advances in Neural Information Processing Systems, NeurIPS 2020, Online, 6–12 December 2020; pp. 19884–19895.

18. Yarats, D.; Kostrikov, I.; Fergus, R. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Online, 3–7 May 2021.

19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 26th Advances in Neural Information Processing Systems, NeurIPS 2012, Lake Tahoe, NV, USA, 3–8 December 2012.
46. Lee, K.H.; Fischer, I.; Liu, A.; Guo, Y.; Lee, H.; Canny, J.; Guadarrama, S. Predictive information accelerates learning in rl. In Proceedings of the 34th Advances in Neural Information Processing Systems, NeurIPS 2020, Online, 6–12 December 2020; pp. 11890–11901.

47. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.

48. Bellemare, M.G.; Naddaf, Y.; Veness, J.; Bowling, M. The arcade learning environment: An evaluation platform for general agents. J. Artif. Intell. Res. 2013, 47, 253–279. [CrossRef]  

49. Anand, A.; Racah, E.; Ozair, S.; Bengio, Y.; Côté, M.A.; Hjelm, R.D. Unsupervised state representation learning in atari. In Proceedings of the 33rd Advances in Neural Information Processing Systems, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019.

50. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Fires, B.A.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. In Proceedings of the 34th Advances in Neural Information Processing Systems, NeurIPS 2020, Online, 6–12 December 2020; pp. 21271–21284.

51. Fischer, I. The conditional entropy bottleneck. Entropy 2020, 22, 999. [CrossRef] [PubMed]

52. Zhang, A.; McAllister, R.; Calandra, R.; Gal, Y.; Levine, S. Learning invariant representations for reinforcement learning without reconstruction. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Online, 3–7 May 2021.

53. Agarwal, R.; Machado, M.C.; Castro, P.S.; Bellemare, M.G. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Online, 3–7 May 2021.

54. Ferns, N.; Precup, D. Bisimulation Metrics are Optimal Value Functions. In Proceedings of the 30th Association for Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, QC, Canada, 23–27 July 2014.

55. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.

56. Laskin, M.; Yarats, D.; Liu, H.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 17th North American Chapter of the Association for Computational Linguistics, NAACL 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.

57. Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Online, 18–24 July 2021.

58. Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Online, 18–24 July 2021.

59. Bardenet, R.; Hoci, M.; Goyal, S.; Bach, F. Dropout is not ensembling: It transpforms the data distribution. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.

60. Liu, H.; Abbeel, P.; Darrell, T.; Levine, S. Deep object-centric representations for generalizable robot learning. In Proceedings of the 34th International Conference on Machine Learning, ICML 2021, Online, 18–24 July 2021.

61. Pathak, D.; Exler, T.; Levine, S.; Allden-Hock, M.; Darrell, T. Efficient exploration via state marginal matching. In Proceedings of the 35th International Conference on Machine Learning, ICML 2022, Online, 29 July–4 August 2022.

62. Burda, Y.; Edwards, H.; Storkey, A.; Klimov, O. Exploration by random network distillation. In Proceedings of the 16th International Conference on Artificial Intelligence, UAI 2014, Quebec City, QC, Canada, 23–27 July 2014.

63. Aubret, A.; Matignon, L.; Hassas, S. A survey on intrinsic motivation in reinforcement learning. arXiv 2020, arXiv:1908.06976.

64. Nguyen, T.; Luu, T.M.; Yu, T.; Yoo, C.D. Sample-efficient reinforcement learning representation learning with curiosity contrastive forward dynamics model. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Brisbane, Australia, 21–25 May 2018.

65. Pathak, D.; Gandhi, D.; Gupta, A. Self-supervised exploration via disagreement. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 10–15 June 2019; pp. 5062–5071.

66. Burda, Y.; Edwards, H.; Storkey, A.; Klimov, O. Exploration by random network distillation. In Proceedings of the 16th International Conference on Artificial Intelligence, UAI 2014, Quebec City, QC, Canada, 23–27 July 2014.

67. Mnih, V.; Badia, A.P.; Lazaric, A.; Pinto, L.; Abbeel, P.; Barreto, A.; Van de Wiele, T.; Warde-Farley, D.; Mnih, V. Fast task inference with variational intrinsic successor features. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Online, 26 April–1 May 2020.

68. Liu, H.; Abbeel, P. Aps: Active pretraining with successor features. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Online, 18–24 July 2021; pp. 6736–6747.

69. Singh, H.; Misra, N.; Hriazdo, V.; Fedorowicz, A.; Demchuk, E. Nearest neighbor estimates of entropy. Am. J. Math. Manag. Sci. 2003, 23 301–321. [CrossRef]
72. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In Proceedings of the 34th Advances in Neural Information Processing Systems, NeurIPS 2020, Online, 6–12 December 2020; pp. 9912–9924.

73. Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction; MIT Press: Cambridge, MA, USA, 1998.

74. Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. Planning and acting in partially observable stochastic domains. *Artif. Intell.* **1998**, *101*, 99–134. [CrossRef]

75. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.

76. Ziebart, B.D. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*; Carnegie Mellon University: Pittsburgh, PA, USA, 2010; 216p.

77. Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; Davidson, J. Learning latent dynamics for planning from pixels. In Proceedings of the International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 10–15 June 2019; pp. 2555–2565.

78. Hafner, D.; Lillicrap, T.; Ba, J.; Norouzi, M. Dream to Control: Learning Behaviors by Latent Imagination. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Online, 26 April–1 May 2020.

79. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014; pp. 1–15.

80. Shu, R.; Nguyen, T.; Chow, Y.; Pham, T.; Than, K.; Ghavamzadeh, M.; Ermon, S.; Bui, H. Predictive coding for locally-linear control. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Online, 12–18 July 2020.