REINFORCEMENT LEARNING WITH PREDICTIVE CONSISTENT REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning informative representations from image-based observations is a fundamental problem in deep Reinforcement Learning (RL). However, data inefficiency remains a significant barrier. To this end, we investigate **P**redictive **C**onsistent **R**epresentations (**PCR**) that enforces predictive consistency on a learned dynamic model. Unlike previous algorithms that simply exploit a forward dynamics model, the PCR agent is trained to predict the future state and retain consistency across the predicted state of observation and its multiple views, which is demonstrated through careful ablation experiments. We empirically show that PCR outperforms the current state-of-the-art baselines in terms of data efficiency on a series of pixel-based control tasks in the DeepMind control suite. Notably, on challenging tasks like Cheetah-run, PCR reaches a 47.4% improvement when environmental steps are limited to 100k steps.

1 INTRODUCTION

Deep Reinforcement Learning (RL) harnesses the expressive power of deep neural networks and the long-term reasoning ability of RL to solve sequential decision-making problems (Mnih et al., 2015). Recent years have witnessed the sensational progress of it in various complex control tasks, such as playing video games (Berner et al., 2019), robotic control (Kalashnikov et al., 2018) and autonomous driving (Shalev-Shwartz et al., 2016).

Despite the notable success of deep RL, recent studies have observed that data/sample inefficiency severely impedes its performance when learning from high dimensional observations (Lake et al., 2016). This remains a significant barrier to the real-world applicability of deep RL, where collecting experiences is often costly and time-consuming (Dulac-Arnold et al., 2019). For instance, a successful RL agent requires several months to develop a decent grasping skill, standing sharply contrast to the human-level efficiency (Kalashnikov et al., 2018). Accordingly, elevating data efficiency is of paramount importance for the broader progress of deep RL.

A number of existing works approach this goal by augmenting deep RL with self-supervised tasks. The motivation of that is two-fold: (i) solely using potentially sparse reward signals is data-inefficient to fit a high-capability encoder (Yarats et al., 2021; 2020); (ii) Self-Supervised Learning (SSL) unleashes the potential of massive unsupervised signals for representation learning, achieving remarkable performance in downstream vision and language tasks, particularly in low data regimes (Devlin et al., 2019; Grill et al., 2020). Beyond that, there are proliferative paradigms of designing SSL tasks in RL due to its interactive and temporal-correlated training mechanism, such as maximally preserving predictive information (van den Oord et al., 2018; Lee et al., 2020b), modeling dynamics (Schwarzer et al., 2020; Shelhamer et al., 2016) and discriminating instances at a spatial or temporal level (Laskin et al., 2020a; Stooke et al., 2021).

In this work, we first revisit two of hypotheses made in the aforementioned methods: good state representations are the ones that (i) encode temporally predictive information and (ii) is consistent across augmented observations (views). Combining these two hypotheses naturally raises a new one: (iii) *a powerful state representation is capable of predicting the future (by modeling dynamics), and the prediction itself is consistent across multiple views, on top of which deep RL algorithms should be significantly more data-efficient.* Though considerable effort has been dedicated to testing hypotheses (i) and (ii), the explorations for hypothesis (iii), to our best knowledge, are still rare. We consider that it meets the Markovian assumption on the latent dynamics model (Hafner et al., 2019b; Lee et al., 2020b), and is thus worthy of study.

To investigate it, we propose Predictive Consistent Representations (PCR) that enforces predictive consistency on a learned dynamics model. The term 'Predictive' refers to the hypothesis (i) as we model the environmental dynamics, and the term 'Consistent' refers to the hypothesis (ii) as the dynamics model itself is forced to be consistent across multiple views of its input. Composing them with the RL objective teases out the final objective in the whole course of policy learning. We demonstrate our framework in Figure 2.1.

We evaluate PCR on a series of pixel-based control tasks from the DeepMind control suite, a common benchmark for testing the data efficiency in deep RL from visual observations. The empirical results show that our PCR agents outperform prior state-of-the-art baselines on the majority of tasks. We have also done extensive experiments to show that: (i) the powerfulness of PCR comes from its predictive consistency objective and (ii) predictive consistency is superior to contrastive consistency in RL settings.

We highlight our main contributions below:

- We propose a new hypothesis on the state representation, and present a novel method, dubbed PCR, to validate its reasonableness w.r.t. data efficiency in RL.
- We demonstrate that PCR agents outperform prior state-of-the-art baselines on the widely adopted pixel-based DeepMind control benchmark in terms of both data-efficiency and asymptotic performance.
- With careful ablation studies, we verify the effectiveness of the predictive consistency itself and against other similar approaches like contrastive consistency.

2 Related Work

2.1 Data Efficient RL

A number of approaches have advanced the data efficiency in RL from high dimensional observations such as images. Broadly, we classify the existing methods into three groups. The first group of works improves data efficiency by explicitly building the world models from environments. Representative works include PlaNet (Hafner et al., 2019b), Dreamer (Hafner et al., 2019a) and SLAC (Lee et al., 2020a), which use world models to perform planning or rollout-sampling in the latent state space. Among these, another vine of research devotes to shaping the state representations from the learned forward dynamics model without planning (Gelada et al., 2019; Kipf et al., 2019).

The second group of works effectively deploy the power of data augmentation. For instance, CURL (Laskin et al., 2020a) learns contrastive representation in RL from pixels and achieves remarkable data efficiency in DeepMind Control Suite. Based on that, RAD (Laskin et al., 2020b) and DrQ (Yarats et al., 2020) directly incorporate data augmentation with visual observations to regularize the model-free RL algorithms. In SPR (Schwarzer et al., 2020), data augmentation is injected into a forward dynamics model, enforcing temporal consistency across the state representation.

Recently, unsupervised/self-supervised representation learning has achieved glaring success in vision and language tasks by exploiting the internal structure of unlabeled data (Grill et al., 2020; Devlin et al., 2019). It, therefore, arouses the widespread interest of the third group. Works in this group have investigated various auxiliary tasks for representation learning in RL and have shown huge advances in data efficiency. One common rule is to leverage the temporal structure of the environment. From this, CPC (van den Oord et al., 2018), DRIML (Mazoure et al., 2020), PI-SAC (Lee et al., 2020b) maximally preserve the predictive information in state representation; SPR (Schwarzer et al., 2020) and Shelhamer et al. (2017) achieve this in a similar way by encoding the representation with environmental information by predicting future states; ST-DIM (Anand et al., 2019) and ATC (Stooke et al., 2021) introduce contrastive losses that operates on both spatial and temporal levels. In contrast, another designing paradigm focuses on the instances level. Representative examples are CURL and SAC-AE (Yarats et al., 2021) that do discrimination or reconstruction on the pixel-based observations.

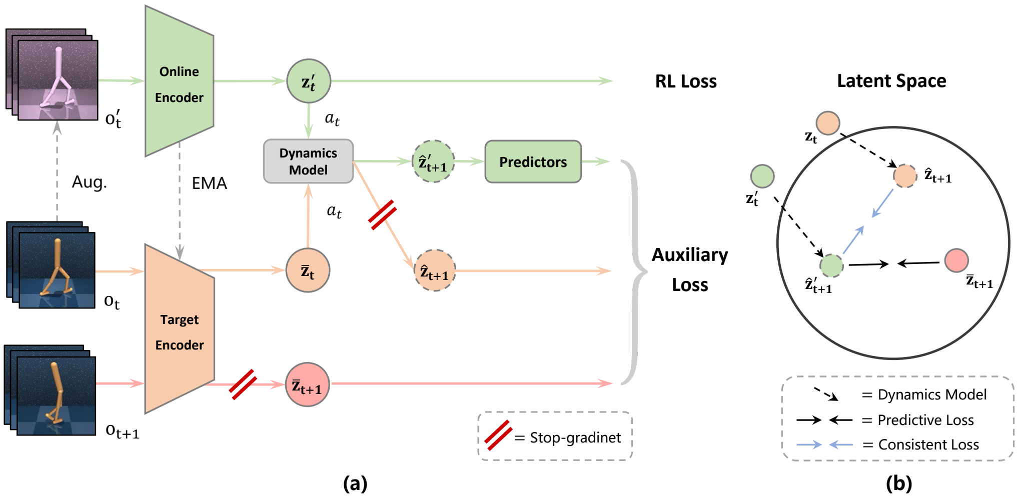

Figure 1: **(a) Overview of Predictive Consistent Representations.** A view of observation o'_t (o_t) is encoded into a representation z'_t (z_t) in the online branch in green (target branch in orange and red). The dynamics model takes as input z'_t (z_t) and action a_t, teasing out a prediction $\hat{z}'_{t+1}$ ($\hat{z}_{t+1}$) of future state representation $\bar{z}_{t+1}$. Representation z'_t is used in RL tasks, while $(\hat{z}'_{t+1}, \hat{z}_{t+1}, \bar{z}_{t+1})$ are used in auxiliary task with two predictors. **(b) Illustration of auxiliary loss in the latent space.** Auxiliary loss is composed of (i) *predictive loss* that is optimized to learn a dynamics model for predicting future and (ii) *consistent loss* that enforces predictive consistency on the dynamics model.

2.2 Predictive and Consistent Representation for RL

Two common hypotheses for state representations are often investigated in the literature. The first one is hypothesis (i) that temporally predictive information is vital for state representations. They realize it through different approaches, such as predicting the future state (Guo et al., 2020; Gelada et al., 2019; Kipf et al., 2019). The second one is hypothesis (ii) that a good state representation should be consistent across different views of observations. Typical methods include contrastive learning (Laskin et al., 2020a; Stooke et al., 2021) and data-regularization (Laskin et al., 2020b; Yarats et al., 2020).

Our method PCR is in line with the hypothesis (iii) that a powerful state representation is capable of predicting the future (akin to hypothesis (i)), and the prediction itself is consistent between different views of observations (akin to hypothesis (ii)). The key insight here is that we enforce the consistency on the learned dynamics model rather than directly on the state representation as in CURL and SPR. We consider that it meets the Markovian assumption that the transition is conditional independence of the current observations given current state (Lee et al., 2020a; Hafner et al., 2019b), where different views of observations are assumed to be emitted from the same latent state.

In this sense, PCR bears some resemblance to SPR. Both integrate the data augmentation with the dynamics model to learn predictive and consistent representations. We illustrate the difference between them in r.h.s of Figure 2.1: PCR takes both predictive loss and consistent loss into consideration, while SPR only considers the former. Essentially speaking, PCR forces the dynamics model itself to be consistent across different views of observations. Such consistency is not explicitly exploited by SPR. We ablate the consistent loss in Section 5.3 and show it indeed improves our performance on the majority of tasks from DMC.

We also relate PCR to CPC|Action (Guo et al., 2018) and CMC (Tian et al., 2020), which extent contrastive learning in the dimension of future-predicting and multiview, respectively. PCR combines their key ideas by optimizing an associative auxiliary objective. Specifically, we introduce predictive loss to learn a dynamics model and consistent loss to enforces consistency on the dynamics model. Composing these two forms a ‘multiview’ consistency in the latent space, where we extend the concept of ‘views’ in the temporal level, as illustrated in the r.h.s of Figure 2.1. Moreover, in our ablations, we show that prediction loss (Grill et al., 2020) is superior to the contrastive losses, which are adopted in their models.

3 Background

Reinforcement Learning. We train an RL agent within a Markov Decision Process defined by a tuple $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, where $\mathcal{S} = \mathbb{R}^k$ is the state space, $\mathcal{A}$ is the action space, the transition dynamics $P = Pr(s_{t+1}|s_t, a_t)$ determines the transition distribution over next state s_{t+1} given the state s_t and action a_t, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ represents the reward function whose element $r(s_t, a_t)$ is the expected reward collected by taking action a_t in state s_t, $\gamma \in [0, 1)$ is the discount factor that determines the present value of future reward. The objective of the agent is to find a policy $\pi(a_t|s_t)$ that maximizes the cumulative discounted return $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$. Crucially, in pixel-based control tasks, the agent perceives image-based observations $o_t \in \mathcal{O}(s_t) = \mathbb{R}^n$.

Soft Actor-Critic with Augmented Data. SAC (Haarnoja et al., 2018) is an off-policy RL algorithm for continuous control tasks. It optimizes a stochastic policy by maximizing a γ-discounted maximum-entropy objective (Ziebart et al., 2008). RAD-SAC (Laskin et al., 2020b), which we abbreviate as RAD, further enhances SAC with augmented data to elevate the data efficiency. The RAD agent trains the critic Q_ϕ by minimizing the Bellman residual:

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{\upsilon \sim \mathcal{D}} \left[\left(Q_\phi(o'_t, a_t) - \left(r_t + \gamma V(o'_{t+1}) \right) \right)^2 \right], \tag{1}$$

where $\upsilon = (o'_t, a_t, r_t, o'_{t+1})$ is sampled from the replay buffer $\mathcal{D}$ with augmentation on observations o_t and o_{t+1}. We practically learn two critic Q^1_ϕ, Q^2_ϕ and their targets for double Q-learning in practice (refer to Haarnoja et al. (2018) for details), and we omit it for expression simplicity. The target Q-value is estimated by

$$V(o'_{t+1}) = \mathbb{E}_{a' \sim \pi} \left[Q_{\bar{\phi}}(o'_{t+1}, a') - \kappa \log \pi_\psi(a'|o'_{t+1}) \right], \tag{2}$$

where the parameters of the target critic $Q_{\bar{\phi}}$ are updated in a exponential moving average (EMA) fashion with coefficient τ, κ is a positive entropy coefficient that balances the reward maximization and the behavioral stochasticity. For training the policy, one can apply the reparameterization trick to minimize the objective:

$$\mathcal{L}_\pi(\psi) = -\mathbb{E}_{a \sim \pi} \left[Q_\phi(o'_t, a) - \kappa \log \pi_\psi(a|o'_t) \right]. \tag{3}$$

4 Method

We propose a novel method named PCR that learns predictive consistent state representations for data-efficient RL. The main innovation of PCR is to introduce predictive consistency (consistent loss) on the dynamics model. Inspired by the success of He et al. (2020); Grill et al. (2020), we design a two-stream network architecture. One stream is trained in an online manner, while the other one serves a self-supervised target updated in EMA fashion. An overview of the PCR architecture is provided in Figure 2.1. We divide it into three main components which we describe as follows.

4.1 Online and Target Encoders

We consider a one-step transition $((o_t, o'_t), a_t, r_t, (o_{t+1}))$ in MDPs. Following Grill et al. (2020); Chen et al. (2020), we train two encoders: the *online encoder* f_θ and *target encoder* f_ξ, parameterized by a set of parameters θ and ξ, respectively.[1] The online encoder compresses the augmented observation (or view) o'_t into representation $z'_t \triangleq f_\theta(o'_t)$, while the target encoder is applied on the original observation: $\bar{z}_t \triangleq f_\xi(o_t)$. Two encoder share a same structure but different sets of parameters. The target parameters ξ are an EMA of the online parameters θ, which are updated via the rule:

$$\xi \leftarrow \tau'\theta + (1 - \tau')\xi, \tag{4}$$

where $\tau' \in [0, 1]$ is the EMA coefficient. Only the online parameters θ are updated via back-propagation.

[1] We include the projectors (Grill et al., 2020) in the conjunct encoders without ambiguity, which share the same parameters with Q-learning head. More details are presented in Section 5.

Algorithm 1 PCR: Predictive **C**onsistent **R**epresentation applied to RAD-SAC

```
$\phi, \bar{\phi}, \psi, \theta, \xi, \alpha, \beta$: randomly initialized network parameters    ▷ Copy weights $\bar{\phi} \leftarrow \phi, \xi \leftarrow \theta$
$\{\lambda_{pred}, \lambda_{cons}\}, \{\tau, \tau'\}$: loss and EMA coefficients    ▷ Default: $\lambda_{pred} = \lambda_{cons} = 1$
$\{\eta_\phi, \eta_\psi, \eta_\theta\}, N$: learning rates, mini-batch size    ▷ $h, q_\alpha, q_\beta$ share $\eta_\theta$ as learning rates
while training do
    for each environment step do
        $a_t \sim \pi_\psi(\cdot|f_\theta(o_t))$    ▷ Sample action from current policy
        $s_{t+1} \sim P(\cdot|s_t, a_t), o_{t+1} \sim O(s_{t+1})$    ▷ Transit from the underlying state $s_t$
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_t, o'_t, a_t, r_t, o_{t+1})\}$    ▷ Collect experiences with augmentation on $o_t$
    end for
    for each gradient step do
        $\{(o_t, o'_t, a_t, r(s_t, a_t), o_{t+1})\}_{i=1}^N \sim \mathcal{D}$    ▷ Sample a mini batch
        $\psi \leftarrow \psi - \eta_\psi \hat{\nabla}_\psi \mathcal{L}_(\psi)$    ▷ gradient step on actor
        $\{\phi, \theta\} \leftarrow \{\phi, \theta\} - \eta_\phi \hat{\nabla}_{\{\phi,\theta\}} \mathcal{L}_Q(\phi; \theta)$    ▷ gradient step on critic(s)
        $\{\theta, \alpha, \beta\} \leftarrow \{\theta, \alpha, \beta\} - \eta_\theta \hat{\nabla}_{\{\theta,\alpha,\beta\}} (\lambda_{pred} \mathcal{L}_{pred}(\theta; \alpha, \beta) + \lambda_{cons} \mathcal{L}_{cons}(\theta; \alpha, \beta)$
            ▷ gradient step on online encoder, predictors and dynamics model
        $\bar{\phi} \leftarrow \tau\phi + (1 - \tau)\bar{\phi},\ \xi \leftarrow \tau\theta + (1 - \tau)\xi$    ▷ EMA update of target critic and encoder
    end for
end while
```

4.2 Dynamics Model for Predicting Future

Learning a dynamics model has been well investigated in RL from visual observations. The agents learn informative state representations by predicting the future state in the latent space, avoiding the reconstruction of pixel-level objectives in the observation space. To this end, we introduce an action-conditioned *dynamics model* h that acts on state representations to augment the policy learning.

The dynamics model h takes the representation z'_t and action a_t as input, and outputs a prediction $\hat{z}'_{t+1} = h(z'_t, a_t)$ of the future representation $\bar{z}_{t+1}$. As suggested by Grill et al. (2020), we apply a *predictor* q_α to the online branch and minimize the normalized ℓ_2 loss between the prediction and target representation:

$$\mathcal{L}_{pred}(\theta) = 2 - 2 \cdot \frac{\langle q_\alpha(\hat{z}'_{t+1}), \bar{z}_{t+1} \rangle}{\|q_\alpha(\hat{z}'_{t+1})\|_2 \cdot \|\bar{z}_{t+1}\|_2} \tag{5}$$

4.3 Dynamics Model for Predictive Consistency

Though many existing works have explored the temporal consistency (Stooke et al., 2021; Yu et al., 2021) and instance consistency (Laskin et al., 2020a; Schwarzer et al., 2020) of state representations, the related works on the predictive consistency of dynamics model are still rare. To begin with, we give a formal definition of the predictive consistent dynamics model as follows.

Definition 1 *A dynamics model h is predictive consistent with encoder g if, for any action $a_t \in \mathcal{A}$ and any view $(o_{t,1}, o_{t,2})$ of a observation $o_t \in O(s_t)$, the following condition holds:*

$$d_\mathcal{Z}\left(\hat{z}_{t+1,1}, \hat{z}_{t+1,2}\right) \leq \epsilon$$

where $\hat{z}_{t+1,i} \triangleq h(g(o_{t,i}), a_t)$ for $i \in \{1, 2\}$, $d_\mathcal{Z}$ is a distance metric on the space $\mathcal{Z}$, and ϵ is the acceptable error bound.

Remark 1 We posit that one of the characteristics that a good dynamics model should possess is to consistently predict the future given the current representations extracted from the multiple views of observation. This characteristic is in line with the basic assumption on the latent dynamics model that transition is conditional independent of observations given the current (latent) state. Based on that, we enforce the predictive consistency on the dynamics model to accelerate the representation learning.

Then comes a crucial question: how to choose a proper distance metric on the latent state space? The following proposition suggests that ℓ_2 loss is possibly a reasonable choice.

Proposition 1 *Minimizing the ℓ_2 loss $\|\hat{z}_{t+1}-q_\beta(z'_{t+1})\|_2$ is equivalent to maximizing a lower bound of conditional mutual information $I(z'_t;\hat{z}_{t+1}|a_t)$, where q_β is a predictor distinguished from q_α.*

Proposition 1 indicates that minimizing $\|\hat{z}_{t+1}-q_\beta(z'_{t+1})\|_2$ is essentially to maximally maintain the information shared between multiple views of o_t, which excludes action information. This essentially meets the property of predictive consistency. Based on that, we give a practical objective in the course of learning: given the action a_t and state representations $(\hat{z}'_{t+1}, \hat{z}_{t+1} \triangleq h(\bar{z}_t, a_t))$, the consistent loss (for predictive consistency) of the dynamics model is calculate as:

$$\mathcal{L}_{cons}(\theta) = 2 - 2 \cdot \frac{\langle q_\beta(\hat{z}'_{t+1}), \hat{z}_{t+1}\rangle}{\|q_\beta(\hat{z}'_{t+1})\|_2 \cdot \|\hat{z}_{t+1}\|_2}, \tag{6}$$

Although the predictive consistency loss does not circumvent the trivial solution on its own, i.e., $h(\cdot,\cdot) \equiv 0$, such a behavior is not observed in our experiments. We attribute it to the updates of target encoder f_ξ, which is jointly optimized by the RL and auxiliary objectives. It serves as the target future prediction in L_{pred}, and constant equilibria are in conflict with optimizing L_{pred}.

Composing Training Objectives. Composing the RL and auxiliary objectives gives the overall training objective:

$$\mathcal{L}_{total}(\phi,\psi,\theta) = \underbrace{\mathcal{L}_Q(\phi) + \mathcal{L}_\pi(\psi)}_{\text{RL Loss}} + \underbrace{\lambda_{pred}\mathcal{L}_{pred}(\theta) + \lambda_{cons}\mathcal{L}_{cons}(\theta)}_{\text{Auxiliary Loss}}, \tag{7}$$

where λ_{pred} and λ_{cons} are the hyperparameters that steer the weights of the conjunct objectives. We summarize PCR in Algorithm 1.

5 Experiments

In this section, we first introduce the experiment setup including the environments, baselines, and implementation details of PCR. Furthermore, we conduct two ablation studies to explore the effective of Predictive Consistency and compare it with Contrastive Consistency, which is widely adopted by previous works (Laskin et al., 2020a; Chen et al., 2020; He et al., 2020).

5.1 Setup

Environments. We evaluate PCR on the DeepMind Control Suit (DMControl) (Tassa et al., 2018), a standard benchmark for measuring the data efficiency of RL algorithms with vision-input in continuous action space. Following previous works, we measure the performance of PCR at 100k (DMControl100k) and 500k (DMControl500k) environment steps during training, where the environment steps are defined as the number of steps the underlying simulators take. DMControl100k has been widely accepted as a benchmark for measuring data efficiency, while DMControl500k evaluates the asymptotic long-horizon performance of an RL algorithm.

Implementation Details. We build PCR on top of RAD. We first build four-layer CNN online and target encoders to encode the original observations. The representations output by the online (target) encoders are feed into a single-layer online (target) projector with BatchNorm being applied. This follows the previous works (Laskin et al., 2020a;b). Note that our online projector shares same architecture and parameter with Q-learning head. To keep our modules simple, the dynamics model is set as a two-layer MLP with LayernNorm on the last layer. Meanwhile, we adopt **crop** and **random translation**, powerful augmentations suggested by Laskin et al. (2020b), to obtain the multiple views of the original observation. The weights of λ_{pred} and λ_{cons} in Eq. 7 are set to 1 for simplicity. We list more details, including the augmentation mechanism and model hyperparameters, in Appendix B.

5.2 Performance

Baselines. For a fair comparison, we take as baselines the previous state-of-the-art algorithms that also focus on data-efficiency: SLAC (Lee et al., 2020a) learns compact latent representation through a forward model; CURL(Laskin et al., 2020a) learns a contrastive representation of the observations; SAC+AE (Yarats et al., 2021) introduces an auxiliary task of observation reconstruction; DrQ

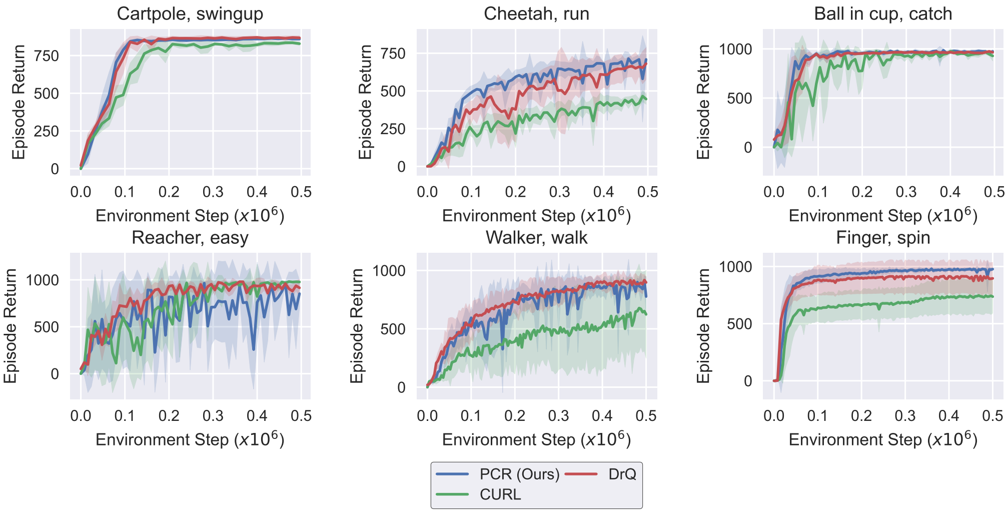

Figure 2: Learning of PCR, as well as two other SOTA methods (DrQ (Yarats et al., 2020) and CURL (Laskin et al., 2020a)). The results is averaged cross five different training seeds.The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

(Yarats et al., 2020) uses both data augmentation and weighted Q-objectives. All the above four methods are all built upon SAC (Haarnoja et al., 2018), a simple yet efficient algorithm for learning with state-based inputs. Unlike those methods, PlaNet (Hafner et al., 2019b) and Dreamer (Hafner et al., 2019a) takes a different approach: they explicitly learns a world model for generating fictitious trajectories.

Analysis. We evaluate PCR and baselines on six commonly-adopted environments. The scores measure the their performances at 100k and 500 environment steps, which is dubbed DMControl100k and DMControl500k, respectively. We report the scores achieved by PCR and baselines in Table 1, and plot the training curve in Figure 5.2. We train PCR five times with random seeds. The scores are averaged over five runs with ten times of evaluation on each runs. The results show that PCR reaches the state-of-the-art performance on the majority of (4 out of 6) DMControl100k and DMControl500k.[2] Particularly, in the most challenging task *Cheetah-Run*, PCR outperforms all previous state-of-the-art algorithms by **47.4%** on data limited DMControl100k benchmark and **13.6%** on asymptotic optimal DMControl500k benchmark.

We point out that simply incorporating a forward dynamics model or applying augmentations to visual observations, as suggested by the baselines, is not enough to achieve higher data efficiency. To this end, PCR takes one step further through consistent loss that enforces predictive consistency on the learned dynamics model. Such consistency makes our dynamics model more robust against multiple views of the original observations during the transition in latent space. The results show that this property significantly elevates data efficiency in complex control tasks with visual observations. To take a deeper investigation, we conduct ablation studies in Section 5.3 to verify the impact of predictively consistency, followed a comparison between contrastive representations and predictive representations in Section 5.3.2.

5.3 Ablations

5.3.1 Evaluation of Predictive Consistency

As described in Section 4, PCR explicitly learns a predictive consistent dynamics model and implicitly encode such consistency into the state representation. A crucial part is a consistent loss that forces the predicted representation from the online encoder to be predictive of the ones in the target

[2] Data listed in the table referred to Laskin et al. (2020b); Yarats et al. (2020).

Table 1: Scores of baselines and PCR (Ours) on DMControl100k and DMControl500k. We select 6 representative environments for benchmarking. The results show the mean and standard deviation averaged over five runs and the best results are indicated in bold. PCR (Ours) reaches state-of-the-art performance on 4 out of 6 tasks on both benchmarks and just below DrQ on *Reacher easy* and *Walker walk*. Particularly, PCR significantly outperforms DrQ on the most challenging task *Cheetah-run* with **47.4%** improvement on data limited DMControl100k bechmark and **13.6%** on asymptotic optimal DMControl500k benchmark.

100k Step Scores	**SLAC**	**PlaNet**	**Dreamer**	**SAC+AE**	**CURL**	**DrQ**	**Ours**
Finger, spin	693 ±141	136 ±216	341 ±70	740 ±64	767 ±56	901 ±104	**933 ±60**
Cartpole, swingup	-	297 ±39	326 ±27	311 ±11	582 ±146	759 ±92	**839 ±20**
Reacher, easy	-	20 ±50	314 ±155	274 ±14	538 ±233	**601 ±213**	443 ±96
Cheetah, run	319 ±56	138 ±88	235 ±137	267 ±24	299 ±48	344 ±67	**507 ±52**
Walker, walk	361 ±73	224 ±48	277 ±12	394 ±22	403 ±24	**612 ±164**	540 ±59
Ball in cup, catch	512 ±110	0 ±0	246 ±174	391 ±82	769 ±43	913 ±53	**936 ±14**
500k Step Scores							
Finger, spin	673 ±92	561 ±284	796 ±183	884 ±128	926 ±45	938 ±103	**985 ±3**
Cartpole, swingup	-	475 ±71	762 ±27	735 ±63	841 ±45	868 ±10	**875 ±4**
Reacher, easy	-	210 ±390	793 ±164	627 ±58	929 ±44	**942 ±71**	842 ±94
Cheetah, run	640 ±19	305 ±131	570 ±253	550 ±34	518 ±28	660 ±96	**750 ±35**
Walker, walk	842 ±51	351 ±58	897 ±49	847 ±48	902 ±43	**921 ±45**	878 ±18
Ball in cup, catch	852 ±71	460 ±380	879 ±98	794 ±58	959 ±27	963 ±9	**968 ±3**

stream. Hence, to validate the effectiveness of learning predictively consistency, we compare the performances of PCR and its variant that excludes the consistent loss.

Table 2 shows the performance of PCR (in the first line) and PCR without Consistent Loss (in the second line) of 6 tasks on DMControl100k benchmark. Each result is averaged over five different seeds. The significant gains of PCR over its counterpart demonstrate the effectiveness of predictive consistency loss. A takeaway from this observation is that a dynamics model in model-based RL methods would be more informative if invariant (consistent) against different views of observations.

Table 2: Scores of PCR and its variants: (i) PCR without Predictive Consistency and (ii) PCR with Contrastive Loss on DMControl100k (Data Limited Regime). The results show the mean and standard deviation averaged over five runs, and the best results are indicated in bold. All hyperparameters are identical except the corresponding loss module. Neither abandoning the Predictive Consistency module nor replacing Predictive Loss with Contrastive Loss yields better results.

100k Step Scores	**Finger, spin**	**Cartpole, swingup**	**Reacher, easy**	**Cheetah, run**	**Walker, walk**	**Ball in cup, catch**
PCR (Predictive Loss)	**980.3**	**839.3**	455.0	**503.6**	**524.5**	**936.0**
PCR (w/o Consistent Loss)	751.5	817.3	**688.1**	489.4	506.7	914.5
PCR (Contrastive Loss)	971.5	772.4	578.8	469.5	425.4	931.1

5.3.2 Contrastive Representation v.s. Predictive Representation

There have been two branches of methods for learning an informative representation in the visual domain: either with contrastive learning (Chen et al., 2020) or predictive learning (Grill et al., 2020). As concluded in Tian et al. (2020), learning with the contrastive Loss leads to a better representation than the predictive Loss in some vision tasks. The same scenario happened in RL domain: Laskin et al. (2020a); Stooke et al. (2021) use a contrastive objective for learning state representations, while Schwarzer et al. (2020); van den Oord et al. (2018) use a predictive objective. The key difference between these two schemes is that predictive learning only focuses on one specific instance, and the Loss is taken among multiple views (got through augmentation) of visual input. In contrast, contrastive learning takes a global perspective and treats samples except for the focusing one as negative samples.

Comparison of PCR with Contrastive Loss or Predictive Loss To explore the difference of applying either contrastive Loss or predictive Loss in RL with visual inputs, we implemented another version of PCR, the predictive consistency of which is replaced with contrastive Loss. The implementation of contrastive loss follows from Laskin et al. (2020a). In short, an InfoNCE loss is applied between one sample (anchor) and all other samples within the same batch (negatives). Please refer to Appendix B.3 for the detailed implementation of the contrastive Loss.

Table 2 compares the performance of PCR (the first line) and PCR with Contrastive Loss (the third line) on 6 tasks on DMControl100k benchmark. PCR with Predictive Loss (as stated in Algorithm 1) performs fat much better than PCR with Contrastive Loss, especially on challenging tasks like Cheetah-Run. We provide a possible explanation to justify the above distinction in the next.

Learning Representation in RL is another Story In short, the source of the performance gap comes from the interplay between the online nature of RL algorithms and the choices of negative samples in contrastive learning. In the traditional vision domain, samples within a dataset vary significantly since each of them is collected under different conditions. Accordingly, it makes sense to perform contrastion on instance level: i.e., image B should be a negative sample from image A's perspective, and they are supposed to be pulled away in the latent space. However, for RL algorithms like SAC, samples are constructed from online-collected trajectories. Therefore, samples selected from different episodes or different time steps within one episode may be highly similar. In other words, different samples may fail to form an good contrastive pair, and it is undesired to pull them away. This observation is in line with the previous work Schwarzer et al. (2020).

6 Conclusion

In this paper, we presented Predictive Consistency Representation (PCR), a self-supervised representation learning algorithm that significantly improves data efficiency for RL agents with visual inputs. On the one hand, PCR is capable of predicting future states through a forward dynamics model. On the other, this dynamics model is forced to be consistent across multiple views of the input observation. Empirically, PCR achieves state-of-the-art performance on both DMControl100k and DMControl500k benchmarks. With careful ablation studies, we verify the effectiveness of the Predictive Consistency itself and against other similar approaches like Contrastive Consistency. We hope this paper can lead researchers to rethink the fundamental assumptions made by self-supervised learning and bootstrapping more from the online decision nature of Reinforcement Learning.

REFERENCES

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in atari. *NeurIPS*, 2019.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub W. Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *ArXiv*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *ICML*, 2019.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *ICML*, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, 2020.

Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo Ávila Pires, Tobias Pohlen, and Rémi Munos. Neural predictive belief representations. *ArXiv*, 2018.

Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *ICML*, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *ArXiv*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2019b.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*. PMLR, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ArXiv*, 2013.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *ICML*, 2019.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 2016.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *ICML*, 2020a.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *NeurIPS*, 2020b.

Alex Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *NeurlIPS*, 2020a.

Kuang-Huei Lee, Ian Fischer, Anthony Liu, Yijie Guo, Honglak Lee, John Canny, and Sergio Guadarrama. Predictive information accelerates learning in rl. *NeurlIPS*, 2020b.

Bogdan Mazoure, Remi Tachet des Combes, Thang Long DOAN, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *NeurlIPS*, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *ICLR*, 2020.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *ArXiv*, 2016.

Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. Loss is its own reward: Self-supervision for reinforcement learning. *ArXiv*, 2016.

Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. Loss is its own reward: Self-supervision for reinforcement learning. *ArXiv*, 2017.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2021.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *ArXiv*, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*. Springer, 2020.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, 2018.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *ICLR*, 2020.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *AAAI*, number 12, 2021.

Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *ArXiv*, 2021.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.

A Proof of Proposition 1

For the input, let $I(A;B)$ denote the mutual information, $I(A;B|C)$ denote conditional mutual information, $H(A)$ denote entropy, and $H(A|B)$ denote conditional entropy for random variables $A/B/C$.

We show that minimizing objective L_{cons} is equivalent to maximizing a lower bound of the conditional mutual information $I(z'_t;\hat{z}_{t+1}|a_t)/I(\hat{z}'_{t+1};\hat{z}_{t+1}|a_t)$. By the chain rule,

$$I(z'_t;\hat{z}_{t+1}|a_t) = H(\hat{z}_{t+1}|a_t) - H(\hat{z}_{t+1}|a_t, z'_t),$$

where $H(\hat{z}_{t+1}|a_t)$ is irrelevant to z'_t due to the distinction between two encoders f_θ and f_ξ. Maximizing $I(z'_t; \hat{z}_{t+1}|a_t)$ is thus equivalent to maximizing $-H(\hat{z}_{t+1}|a_t, z'_t)$. We introduce a variational distribution $Q_\varphi(\hat{z}_{t+1}|a_t, z'_t)$ parameterized by φ to make computing such entropy tractable. Then we obtain a variational lower bound of $-H(\hat{z}_{t+1}|a_t, z'_t)$ as:

$$\begin{aligned}
-H(\hat{z}_{t+1}|a_t, z'_t) &= \mathbb{E}_{\hat{z}_{t+1},a_t,z'_t}\Big[\log P(\hat{z}_{t+1}|a_t, z'_t)\Big] \\
&= \max_{Q_\varphi} \mathbb{E}_{\hat{z}_{t+1},a_t,z'_t}\Big[\log Q_\varphi(\hat{z}_{t+1}|a_t, z'_t) + D_{KL}\big(P(\hat{z}_{t+1}|a_t, z'_t)||Q_\varphi(\hat{z}_{t+1}|a_t, z'_t)\big)\Big] \\
&\geq \max_{Q_\varphi} \mathbb{E}_{\hat{z}_{t+1},a_t,z'_t}\Big[\log Q_\varphi(\hat{z}_{t+1}|a_t, z'_t)\Big],
\end{aligned}$$

where $D_{KL}(\cdot||\cdot)$ denotes the Kullback–Leibler divergence.

Further assumption is made that $Q_\varphi(\hat{z}_{t+1}|a_t, z'_t)$ follows a Gaussian $\mathcal{N}\big(\hat{z}_{t+1}|q_\beta(h(z'_t, a_t))\big)$ with a diagonal identical covariance matrix $\sigma\boldsymbol{I}$ (Haarnoja et al., 2018; Kingma & Welling, 2013). Then it reaches the final expression of the lower bound:

$$\mathcal{L}'_{cons} \triangleq \max_{h,q_\beta} \mathbb{E}_{\hat{z}_{t+1},a_t,z'_t}\Big[-\|\hat{z}_{t+1} - q_\beta(h(z'_t, a_t)\|_2^2\Big],$$

which is equivalent to the consistent loss $\mathcal{L}_{cons}$ in Eq. 6. One can similar show that same results hold for $I(\hat{z}'_{t+1}; \hat{z}_{t+1}|a_t)$.

This nature renders us more insights into the predictive consistency of dynamic model. Specifically, one can rewrite $I(z'_t; \hat{z}_{t+1}|a_t)$ as $I\big(f_\theta(o'_t); h(f_\xi(o_t), a_t)|a_t\big)$. Since f_ξ is irrelevant to h and f_θ, maximizing such mutual information is essentially to maximally maintain the information shared between multiple views of o_t, which excludes action information. Accordingly, we jointly learn online encoder and dynamic model to achieve this purpose via gradient descent. The discussion of avoiding trivial solution is presented in Section 4.

B Implementation Details of PCR

We provide an implementation in the supplementary material. All the data and scripts for generating figures are included as well.

B.1 PCR Hyperparameters

Most of the hyperparameters follow previous best practice[rad]. Beyond that, we add the hyperparameters for predictive consistency.

Hyperparameter	Value
Augmentation	Crop - walker, walk; Translate - otherwise
Observation rendering	$(100, 100)$
Observation down/upsampling	$(84, 84)$(crop); $(108, 108)$ (translate)
Replay buffer size	100000
Initial steps	1000
Stacked frames	3
Action repeat	2 finger, spin; walker, walk 8 cartpole, swingup 4 otherwise
Optimizer	Adam
Learning rate $(f_\theta, \pi_\psi, Q_\phi)$	$2e-4$ cheetah, run $1e-3$ otherwise
Batch Size	512
Critic target update freq	2
Actor & Critic hidden dim	1024
Q function EMA τ	0.01
Reward discount γ	.99
Encoder CNN layers	4
Number of filters	32
Encoder & Predictor feature dim	50
Non-linearity	ReLU
Encoder EMA τ	0.05
λ_{pred}	1
λ_{cons}	1

B.2 Data Augmentation

We reuse the official open-source implementation of augmentation modules mentioned in Laskin et al. (2020b). The augmentations includes **crop** and **random translate**.

Crop: Extracts a random patch from the original frame. As our experiments will confirm, the intuition behind random cropping is primarily to imbue the agent with additional translation invariance. **Translate**: random translation renders the full image within a larger frame and translates the image randomly across the larger frame. In DMControl we render 100×100 pixel frames and crop randomly to 84×84 pixels.

The pseudo-code for the above augmentations are as follows:

```
def random_crop(imgs, size):
    n, c, h, w = imgs.shape
    w1 = torch.randint(0, w - size + 1, (n,))
    h1 = torch.randint(0, h - size + 1, (n,))
    cropped = torch.empty((n, c, size, size),
        dtype=imgs.dtype, device=imgs.device)
    for i, (img, w11, h11) in enumerate(zip(imgs, w1, h1)):
        cropped[i][:] = img[:, h11:h11 + size, w11:w11 + size]
    return cropped

def random_translate(imgs, size):
    n, c, h, w = imgs.shape
    outs = np.zeros((n, c, size, size))
    h1s = np.random.randint(0, size - h + 1, n)
    w1s = np.random.randint(0, size - w + 1, n)
    for out, img, h1, w1 in zip(outs, imgs, h1s, w1s):
        out[:, h1:h1 + h, w1:w1 + w] = img
    return outs
```

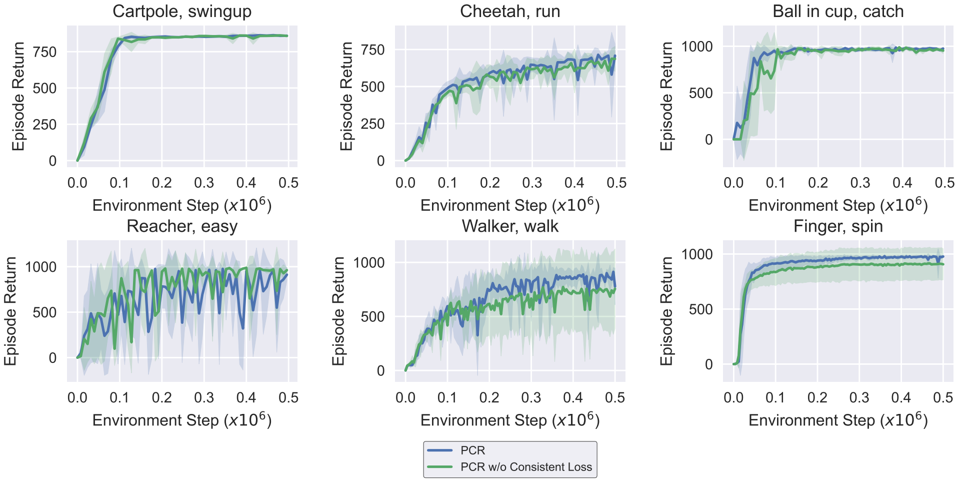

Figure 3: Scores of PCR and PCR without Predictive Consistency.The results is averaged cross five different training seeds.The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs. Without the Predictive Consistency, PCR performs worse and more unstable on complex tasks like *Walker-walk, Cheetah-run* and *Reacher-easy*, but all surpasses previous SOTA like DrQ (Yarats et al., 2020).

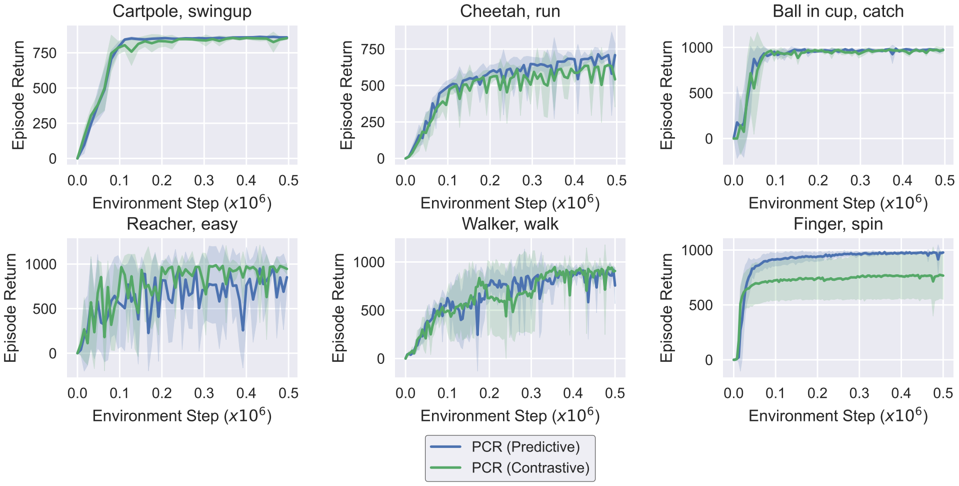

Figure 4: Comparison of the score between PCR and its variant with a Contrastive Loss. The results are averaged across five different training seeds. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs. Replacing Predictive loss with Contrastive loss leads to inferior performance on complex tasks like *Finger-spin, Cheetah-run* and *Reacher-easy*

.

B.3 PCR WITH CONTRASTIVE LOSS

The implementation of PCR with contrastive loss mentioned in ablation Studies is similar to Laskin et al. (2020a), which is the first to adopt contrastive learning objective in RL with visual observation. Contrastive learning can be understood as learning a differentiable dictionary look-up task. Given a query q and keys $\mathbb{K} = \{k_0, k_1, \dots\}$ and an explicitly known partition of $\mathbb{K}$ (with respect to q) $P(\mathbb{K}) = (\{k_+\}, \mathbb{K} \setminus \{k_+\})$, the goal of contrastive learning is to ensure that q matches with k_+ relatively more than any of the keys in $\mathbb{K} \setminus \{k_+\}$. Following Laskin et al. (2020a), we model the similarities between the anchor (q) and targets ($\mathbb{K}$) with bilinear products ($q^T W k$). An InfoNCE loss (van den Oord et al., 2018) is used here :

$$\mathcal{L}_q = \log \frac{\exp(q^T W k_+)}{\exp(q^T W k_+) + \sum_{i=0}^{K-1} \exp(q^T W k_i)} \tag{8}$$

For PCR with contrastive loss, we simply replace the predictive consistency loss with the above InfoNCE loss, and all other hyperparameters remain the same.

B.4 Loss Curve

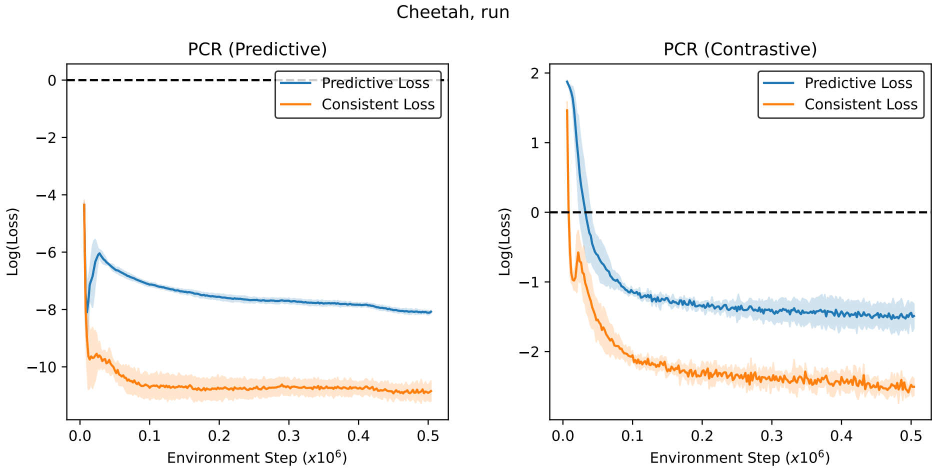

Figure 5: Comparison of Loss between contrastive PCR and Predictive PCR. The y-axis shows the log-loss for a fair comparison. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

Figure B.4 shows the training curve of original PCR (left, inherently adopts predictive loss) and PCR with contrastive loss (right). For the left figure, the definition of predictive loss and consistency loss follows Eq (7). Here point out an interesting observation that learning predictive loss is much more difficult than the consistent loss, as the absolute value of the consistent loss is much lower than the predictive loss.