Contrastive Variational Reinforcement Learning for Complex Observations

Xiao Ma, Siwei Chen, David Hsu, Wee Sun Lee
National University of Singapore
{xiao-ma,siwei-15,leews,dyhsu}@comp.nus.edu.sg

Abstract: Deep reinforcement learning (DRL) has achieved significant success in various robot tasks: manipulation, navigation, *etc.* . However, complex visual observations in natural environments remains a major challenge. This paper presents *Contrastive Variational Reinforcement Learning* (CVRL), a model-based method that tackles complex visual observations in DRL. CVRL learns a contrastive variational world model discriminatively by maximizing the mutual information between latent states and observations, through *contrastive learning*. It avoids modeling the complex observation space unnecessarily, as the commonly used generative observation model often does, and is significantly more robust. We evaluated CVRL on challenging RL benchmark tasks that require continuous control. CVRL achieved comparable performance with state-of-the-art model-based DRL methods on standard Mujoco tasks. It significantly outperformed them on *Natural* Mujoco tasks and a robot box-pushing task with complex observations, *e.g.*, dynamic shadows. The CVRL code is available publicly at https://github.com/Yusufma03/CVRL.

Keywords: Model-Based RL, Contrastive Learning, Complex Observations

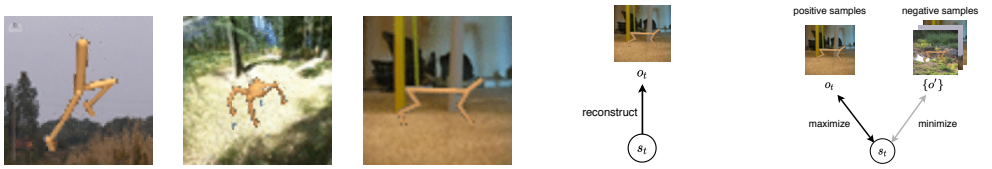
1 Introduction

Model-free reinforcement learning (MFRL) has achieved great success in game playing [1, 2], robot navigation [3, 4] and etc. However, extending existing RL methods to real-world environments remains challenging, because they require long-horizon reasoning with the low-dimensional useful features, *e.g.*, the position of a robot, embedded in high-dimensional complex observations, *e.g.*, visually rich images. Consider a four-legged mini-cheetah robot [5] navigating on the campus. To determine the traversable path, the robot must extract the relevant geometric features that coexist with irrelevant variable backgrounds, such as the moving pedestrians, paintings on the wall, etc.

Model-based RL (MBRL), in contrast to the model-free methods, reasons a world model trained by *generative* learning and greatly improves the sample efficiency of the model-free methods [6, 7, 8]. Recent MBRL methods learn compact latent world models from high-dimensional visual inputs with *Variational Autoencoders* (VAEs) [9] by optimizing the *evidence lower bound* (ELBO) of an observation sequence [10, 11]. However, learning a generative model under complex observations is challenging. VAEs learn the correspondence between observation o_t and latent state s_t by maximizing the conditional observation likelihood $p(o_t \mid s_t)$, *i.e.*, pixel-level reconstruction of observation o_t from agent state s_t. The generative parameterization unavoidably models the entire observation space, including the complex but irrelevant information to decision making. For example in robot navigation, a generative model will try to capture the pixel-level distribution of the paintings on walls, which is irrelevant to the task of navigating to the goal. As a result, standard MBRL based on generative models have a more difficult optimization landscape given complex observations and will be ineffective when applied to natural environments.

In this paper, we present *Contrastive Variational Reinforcement Learning* (CVRL) for robust MBRL under complex observations with high sample-efficiency and long-horizon planning ability. To be robust to complex observations, CVRL learns a *contrastive variational world model* by discriminative contrastive learning, which captures the environment dynamics without modeling the complex observations. Specifically, CVRL maximizes the mutual information between state s_t and observation o_t by scoring the real pair (s_t, o_t) against the fake pairs $\{(s_t, o')\}$ using a simple non-negative

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.



(a) Natural Mujoco Tasks (c) Generative Model (b) Contrastive Model

Figure 1: (a) CVRL addresses the tasks of sparse rewards, many degrees of freedom, and complex observations. (b) Standard generative observation model learns a observation likelihood function $p(o_t \mid s_t)$, *i.e.*, reconstructing observations o_t from s_t, which includes the irrelevant background features. (c) CVRL discovers the correspondence between state s_t and observation o_t by maximizing their mutual information $I(s_t, o_t)$ by scoring the real pair (s_t, o_t) against fake pairs $\{s_t, o_t'\}$, which avoids pixel-level reconstruction.

function. As a result, contrastive learning avoids directly modeling complex observations and is more robust than the generative models. For example, by contrasting observations from different places, the mini-cheetah can identify its current position s_t by simply understanding what observations $\{o'\}$ are unlikely to receive. Mathematically, we derive a *contrastive evidence lower bound* (CELBO), a new lower bound of $p(o_{1:T})$ from the mutual information perspective and it sidesteps the difficulty of learning a complex generative latent world model. CVRL solves the decision making problem combining online model predictive control (MPC) [12] with learned heuristics, *i.e.*, an efficiently and robustly trained actor-critic, for learning long-horizon behavior.

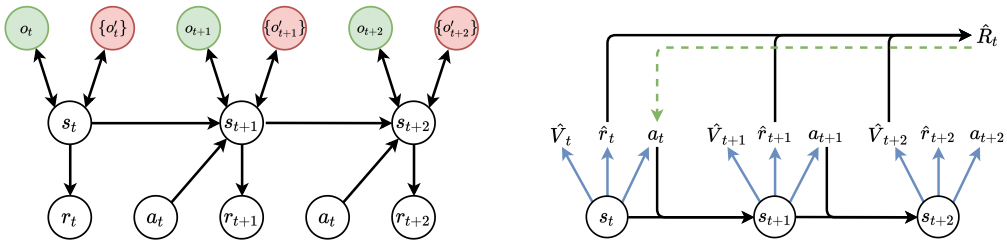
We evaluate CVRL on three challenging continuous control domains: Mujoco tasks designed in Deepmind Control Suite [13], Natural Mujoco tasks, a new domain with more complex observations that we introduce, and Box Pushing, a robot pushing experiment in PyBullet simulator [14]. CVRL outperforms the state-of-the-art model-based RL method in most cases. Results show that CVRL significantly improves the MBRL performance with contrastive representation learning.

2 Related Works

MBRL with World Models. Classic MBRL approaches have focused on planning in a predefined low-dimensional state space [15]. However, manually specifying a world model is difficult [16, 17]. Recently several works demonstrated that we could learn world models from raw pixel inputs. The majority rely on sequential variational autoencoders, which aims to minimize the reconstruction loss of the observations, to capture the stochastic dynamics of the environment [10, 11, 18]. Some other works in robotics learn to predict videos directly for planning [19, 20]. However, real-world observations are complex and noisy, building an accurate generative model over the entire observation space is challenging, which leads to an accumulated compositional error of the world model.

Contrastive Learning. Contrastive learning are widely used for learning word embeddings [21], image representation learning [22], graph representation learning [23] and etc. The main idea is to construct real and fake sample pairs and use a function to score them in different ways. Concurrent to our work, contrastive learning has been applied to learn latent world models [18, 24], motivated from different perspectives. Specifically, Hafner et al. [18] use contrastive learning as an alternative to image reconstruction, where the contrastive learned agent gives worse performance compared with the one learned by image reconstruction. On the contrary, we would like to emphasize the strength of contrastive learning in handling complex visual observations. CVRL significantly outperforms the SOTA model [18] on tasks with complex observations.

Reinforcement Learning under Complex Observations. Given complex observations, discriminative training is generally used to improve the robustness of the agent. Recent works suggest that learning task-oriented observation functions by end-to-end training improves the robustness of observation models [25, 17, 26, 27]. In particular, Ma et al. [26] introduced DPFRL which successfully addressed a challenging task with natural video in the background as well as robot navigation in a simulator constructed from real-world data. However, DPFRL relies on only the RL signal and is sample inefficient compared to model-based approaches. Besides, the generalization ability of DPFRL is also limited due to the model-free policy, and it failed on specific games. CVRL addresses the complex observation from a different perspective: we use contrastive learning to learn the latent world model, which avoids the modeling the complex observations. CVRL benefits both the sample efficiency of the model-based approaches and the robustness of the model-free approaches.



(a) Contrastive Latent World Model (b) Latent Guided MPC

Figure 2: (a) CVRL follows a contrastive latent world model, where the latent states are discovered by contrastive learning, *i.e.*, maximizing the correspondence between state s_t and the positive sample o_t (in green) and minimizing the correspondence between a set of negative samples $\{o'_t\}$ (in red). (b) CVRL chooses actions with a latent guided MPC using latent analytic gradients, which combines online planning with learned heuristics, *i.e.*, an efficiently and robustly learned actor-critic.

3 Contrastive Variational Reinforcement Learning

We introduce *contrastive variational reinforcement learning* (CVRL) for complex observations.

An overview of CVRL is given in Fig. 2. Since the visual observation reveals only part of the true state, we formulate the visual control problem as a partially observable Markov decision process (POMDP) with discrete time step $t \in [1:T]$, continuous actions a_t, complex visual observations o_t, and scalar rewards r_t. CVRL learns a *contrastive latent world model* of the environment, which consists of a transition function $p(s_t \mid s_{t-1}, a_t)$, a reward function $p(r_t \mid s_t)$, and a "discriminative" observation function $f(o_t, s_t)$ by contrastive learning. Contrastive learning scores positive state-observation pair (s_t, o_t) against a set of negative observations $\{o'_t\}$, *i.e.*, maximizing $f(s_t, o_t)$ while minimizing $f(s_t, o'_t)$. Contrastive learning is significantly more robust than generative learning by avoiding pixel-level reconstruction of complex observations. We introduce a new optimization objective, *Contrastive Evidence Lower Bound* which mathematically lower bounds the generative optimization objective. Moreover, CVRL performs decision making by *Latent Guided Model Predictive Control* (MPC) using analytic gradients with learned dynamics. The latent guided MPC combines online planning with learned heuristics, *i.e.*, an efficiently and robustly learned *Hybrid Actor-Critic* model.

3.1 Variational Latent World Models

CVRL adopts a variational latent model for discovering environment dynamics from pixel inputs. Variational latent world models are the sequential version of variational autoencoders (VAEs) [9]. For an observable variable x, VAEs learn a latent variable z that generates x by optimizing an *Evidence Lower Bound* (ELBO) of $\log p(x)$

$$\log p(x) = \log \int_z p(x \mid z) p(z) dz \geq \mathbb{E}_{q(z|x)} \left[p(x \mid z) \right] - KL \left[q(z \mid x) \parallel p(z) \right] \tag{1}$$

where $KL[q(z \mid x) \parallel p(z)]$ denotes the Kullback–Leibler divergence between the prior distribution $p(z)$ and a proposal distribution $q(z \mid x)$ that samples z conditioned on x.

In a sequential decision making task, CVRL applies a multi-step generalization to the single step ELBO by optimizing an ELBO of $p(o_{1:T}, r_{1:T} \mid a_{1:T})$ [10, 11]

$$\log p(o_{1:T}, r_{1:T} \mid a_{1:T}) = \log \int p_\theta(s_t \mid s_{t-1}, a_{t-1}) p_\theta(o_t \mid s_t) p_\theta(r_t \mid s_t) ds_{1:T}$$

$$\geq \sum_{t=1}^{T} \Bigg(\underbrace{\mathop{\mathbb{E}}_{q_\phi(s_t|o_{\leq t}, a_{\leq t})} \left[\log p_\theta(o_t \mid s_t) \right] + \mathop{\mathbb{E}}_{q_\phi(s_t|o_{\leq t}, a_{\leq t})} \left[\log p_\theta(r_t \mid s_t) \right]}_{\text{reconstruction}} - \underbrace{\mathop{\mathbb{E}}_{q_\phi(s_{t-1}|o_{\leq t-1}, a_{<t-1})} \left[KL[q_\phi(s_t \mid o_{\leq t}, a_{\leq t})] \parallel p_\theta(s_t \mid s_{t-1}, a_{t-1}) \right]}_{\text{dynamics}} \Bigg) \tag{2}$$

where θ and ϕ are model parameters. The first part encourages accurate reconstructions of the observation likelihood $p_\theta(o_t \mid s_t)$ and reward likelihood $p_\theta(r_t \mid s_t)$; the second part encourages learning self-consistent dynamics by KL-divergence. Specifically, the second part minimizes the KL divergence between the prior distribution $p_\theta(s_t \mid s_{t-1}, a_{t-1})$ and the posterior distribution $q_\phi(s_t \mid o_{\leq t}, a_{<t})$ conditioned on the observation sequences.

However, the pure stochastic transitions might have difficulties remembering the history and learning stability. Introducing a sequence of additional deterministic states $h_{1:T}$ tackles this issue [28, 11]. In this work, we use the recurrent state space model (RSSM) [11] that decomposes the original latent dynamic model into the following four components

Deterministic state model: $h_t = f_\theta(h_{t-1}, s_{t-1}, a_{t-1})$ Stochastic state model: $s_t \sim p_\theta(s_t \mid h_t)$
Observation model: $o_t \sim p_\theta(o_t \mid s_t)$ Reward model: $r_t \sim p_\theta(r_t \mid h_t, s_t)$

As a result, during training, RSSM approximate $q_\phi(s_t \mid o_{\leq t}, a_{\leq t})$ by $q_\phi(s_t \mid h_t, o_t)$.

3.2 Contrastive Evidence Lower Bound

One big issue of RSSM is that the pixel level generative observation model $p(o_t \mid s_t)$ has to capture the entire observation space, which is problematic given complex observations, *e.g.*, natural observations in autonomous driving or the natural Mujoco games. Given various videos, pixel-level reconstruction becomes difficult which leads to the inaccuracy in the learned latent world model. We introduce *Contrastive Evidence Lower Bound* (CELBO), a robust optimization objective that avoids reconstructing the observations and lower bounds the original ELBO (Eqn. 2).

Instead of maximizing the observation likelihood $p(x \mid z)$, we motivate the solution from a mutual information perspective. The mutual information between two variables x and y is defined as

$$I(x, y) = \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dxdy = \mathbb{E}_{p(x,y)} \left[\log \frac{p(x \mid y)}{p(x)} \right] \tag{3}$$

In Eqn. 2, the observation likelihood is computed for a specific trajectory $\tau = \{o_{1:T}, r_{1:T}, a_{1:T}\}$. In practice, during optimization, we consider the observation likelihood over a distribution of τ. We can rewrite the observation likelihood in Eqn. 2 as

$$\mathop{\mathbb{E}}_{q_\phi(s_t \mid o_{\leq t}, a_{\leq t}) p(o_{\leq t}, a_{\leq t})} \left[\log p_\theta(o_t \mid s_t) - \log p(o_t) + \log p(o_t) \right] = I(s_t, o_t) + E_{p(o \leq t)} \left[\log p(o_t) \right] \tag{4}$$

where the second term $E_{p(o \leq t)} \left[\log p(o_t) \right]$ could be treated as a constant that can be ignored during optimization. Eqn. 4 suggests that maximizing the observation likelihood is equivalent to maximizing the mutual information of the state-observation pairs. The benefit of such a formulation is that mutual information could be estimated without reconstructing the observations, *e.g.*, using energy models [29] or the "compatibility function" [25, 26]. When the observations are complex, mutual information formulation is more robust than the generative parameterization.

To efficiently optimize the mutual information, we use the InfoNCE, which is a contrastive learning method that optimizes a lower bound of the mutual information [30] and is proven to be powerful in a set of self-supervised learning tasks [30, 31]. Using the result in InfoNCE, the mutual information $I(s_t, o_t)$ could be lower bounded by

$$I(s_t, o_t) \geq \mathbb{E}_{q_\phi(s_t \mid o_{\leq t}, a_{\leq t}) p(o_{\leq t}, a_{\leq t})} \left[\log f_\theta(s_t, o_t) - \log \sum_{o'_t \in \mathcal{O}_t} f_\theta(s_t, o'_t) \right] \tag{5}$$

where function $f_\theta(s_t, o_t)$ is a non-negative function that measures the compatibility between state s_t and observation o_t, and $\mathcal{O}_t$ is a set of irrelevant observations sampled from a replay buffer. An intuition for Eqn. 5 is that we want to maximize the compatibility between the state s_t and the real observation o_t (positive sample), while minimizing its compatibility between a set of irrelevant observations (negative samples). In our case, we follow the setup of the original InfoNCE loss and use a simple bi-linear model for $f_\theta(s_t, o_t) = \exp(z_t^{\mathrm{T}} W_\theta s_t)$, where z_t is an embedding vector for observation o_t and W_θ is a learnable weight matrix parameterized by θ.

Substituting Eqn. 4 and Eqn. 5 into Eqn. 2, we derive the CELBO of $p(o_{1:T}, r_{1:T} \mid a_{1:T})$ as

$$\log p(o_{1:T}, r_{1:T} \mid a_{1:T}) \geq$$

$$\sum_{t=1}^{T} \left(\underbrace{\mathbb{E}_{q_\phi(s_t \mid o_{\leq t}, a_{\leq t})} \left[\log \frac{f_\theta(o_t, s_t)}{\sum_{o'_t \in \mathcal{O}_t} f_\theta(s_t, o'_t)} \right]}_{\text{contrastive learning}} + \underbrace{\mathbb{E}_{q_\phi(s_t \mid o_{\leq t}, a_{\leq t})} \left[\log p_\theta(r_t \mid s_t) \right]}_{\text{reconstruction}} - \underbrace{\mathbb{E}_{q_\phi(s_{t-1} \mid o_{\leq t-1}, a_{<t-1})} \left[KL[q_\phi(s_t \mid o_{\leq t}, a_{\leq t})] \parallel p_\theta(s_t \mid s_{t-1}, a_{t-1}) \right]}_{\text{dynamics}} \right) \tag{6}$$

The CELBO objective is similar to the Deep Variational Information Bottleneck [32] in the sense of mutual information maximization. The difference is that we take a mixed approach: we use contrastive learning to optimize the mutual information for only the state-observation pairs, and maximize the reward likelihood $p(r_t \mid s_t)$. Compared to the complex observations, the scalar reward is easy to reconstruct. The quality of contrastive learning highly depends on the choice of negative samples. Reward reconstruction is easier to optimize compared to contrastive learning.

3.3 Hybrid Actor-Critic

CVRL trains an actor-critic using a hybrid-approach, benefiting from the sample-efficiency of the model-based learning and the task-oriented feature learning from the model-free RL.

Actor-Critic from Latent Imagination. First, CVRL uses latent imagination to train the actor-critic, *i.e.*, reasoning the latent world model, which reduces the amount of the interactions needed with the non-differentiable environment. In particular, since the predicted reward and latent dynamics are differentiable, the analytic gradients can back-propagate through the dynamics. As a result, the actor-critic can potentially approximate long-horizon planning behaviors [18].

We adopt the same strategy with Dreamer [18]. We parameterize the actor model $a_t \sim q_\eta(a_t \mid s_t)$ as a tanh-transformed Gaussian, *i.e.*, $a_t = \tanh(\mu_\eta(s_t) + \sigma_\eta(s_t)\epsilon)$, where $\epsilon \sim \mathcal{N}(0, \mathbb{I})$. For value model, we use a feed-forward network $v_\psi(s_t)$ with a scalar output. To compute the analytic gradient, we first estimate the state values of the imagined trajectory $\{\tilde{s}_\tau, \tilde{a}_\tau, \tilde{r}_\tau\}_{\tau=t}^{t+H}$, where the actions are sampled from the actor network. We denote the value estimate of s_τ as a function $\tilde{V}(s_\tau)$. Detailed descriptions of the value estimation and imagined trajectory generation are in the appendix. The Dreamer learning objective is thus given by

$$L_{\text{Dreamer}} = \underbrace{-\mathbb{E}_{p_\theta, q_\eta} \left[\sum_{\tau=t}^{t+H} \tilde{V}(s_\tau) \right]}_{\text{actor loss}} + \underbrace{\mathbb{E}_{p_\theta, q_\eta} \left[\sum_{\tau=t}^{t+H} \frac{1}{2} \parallel v_\psi(s_\tau) - \tilde{V}(s_\tau) \parallel^2 \right]}_{\text{critic loss}} \tag{7}$$

Hybrid Actor-Critic. The performance of latent imagination highly relies on the accuracy of the learned latent world model. Given complex observations, learning an accurate world model is difficult, even with CELBO. We introduce a simple yet effective hybrid training scheme to address this issue. CVRL combines the Dreamer objective with a secondary training signal from off-policy RL, using the ground truth trajectories. Discriminative RL objective can improve the robustness of the actor-critic, while sacrificing the sample-efficiency [26]. Thus, CVRL benefits from both the sample-efficiency of the latent analytic gradient and the robustness of discriminative RL gradient.

In our experiment, we use the Soft Actor-Critic (SAC) [33] to perform off-policy RL. During each optimization step, we use the ground truth trajectory $\{s_t, a_t, r_t\}_{t=1}^{T}$, and use the imagined trajectories $\{\tilde{s}_\tau, \tilde{a}_\tau, \tilde{r}_\tau\}_{\tau=t}^{t+H}$. We have the final objective as $L_{\text{CVRL}} = L_{\text{Dreamer}} + \alpha * L_{\text{SAC}}$.

3.4 Latent Guided Model Predictive Control

Although the learned actor-critic maximizes the accumulated rewards, a model-free policy, without explicit reasoning with world models, might be stuck in local optimum [16, 34]. Model predictive

	Standard				Natural		
	CVRL	Dreamer† [18]	SAC	D4PG† [18]	CVRL	Dreamer	SAC
walker-walk	**980.3**	961.7	355.7	968.3	**941.5**	206.6	44.1
walker-run	377.7	**824.6**	153.0	567.2	**382.1**	82.7	78.1
cheetah-run	528.1	**894.5**	181.8	523.8	**248.7**	100.7	35.8
finger-spin	**717.8**	498.8	258.5	985.7	**850.4**	13.6	23.8
cartpole-balance	**997.1**	979.6	355.5	992.8	**911.9**	163.7	206.0
catpole-swingup	**863.4**	833.6	252.5	862.0	**413.8**	117.6	150.5
cup-catch	**964.9**	962.5	421.4	980.5	**894.2**	131.1	202.2
reacher-easy	**968.2**	935.1	239.2	967.4	**909.1**	133.7	137.7
quadruped-walk	**950.3**	931.6	337.1	-	**878.7**	153.2	204.3
pendulum-swingup	**912.1**	833.0	28.6	680.9	**842.9**	12.4	14.8

Table 1: CVRL achieves comparable performance with the SOTA method, Dreamer [18], on standard Mujoco tasks and significantly outperforms Dreamer on Natural Mujoco tasks. CVRL, Dreamer and SAC are trained for 5×10^6 steps, while the best model-free baseline D4PG is trained for 1×10^8 steps, which we use as an indicator for the performance in standard Mujoco tasks. †Results are taken directly from Dreamer paper.

control (MPC) is widely used to address the continuous control problems, where multiple iterations are required for the policy to converge to the optimal solution [35].

We introduce latent guided model predictive control. Specifically, we use the shooting method in trajectory optimization to address the MPC task. For state s_t, we perform a forward search using the latent world model guided by the learned actor-critic, and generate latent imagination trajectory $\{\tilde{s}_\tau, \tilde{a}_\tau, \tilde{r}_\tau\}_{\tau=t}^{t+H}$. We compute the value estimate for the sampled trajectory using $\tilde{V}(s_t)$, compute the analytic gradient by maximizing $\tilde{V}(s_t)$ and update the action sequences with analytic gradients. In practice, the combination of the offline training with online planning gives a better performance. The detailed description of the algorithm can be found in the appendix.

4 Experiments

We first evaluate CVRL on 10 continuous Mujoco control tasks in Deepmind Control Suite [13]. We then introduce a new, more challenging benchmark, Natural Mujoco. Finally, we apply CVRL to a robot pushing task with RGB images and randomized lighting sources in PyBullet simulator [14].

We compare CVRL with the SOTA generative MBRL method, Dreamer [18], and a model-free baseline, Soft Actor-Critic [33][1]. We also include the result of D4PG [36] trained for sufficient time on standard Mujoco tasks, as a baseline for Mujoco tasks. All results are averaged over 3 random seeds. A detailed description to the experiment setup can be found in the appendix. We show that: 1) CVRL significantly outperforms SAC in all cases, with much fewer training iterations; 2) CVRL significantly outperforms Dreamer on natural Mujoco tasks because of the robust contrastive learning, and achieves comparable performance on standard Mujoco; 3) the proposed hybrid actor-critic training scheme and guided model predictive control further improves the performance of CVRL.

4.1 Mujoco Tasks

The Mujoco tasks are difficult for reinforcement learning methods due to the sparse reward, 3D scenes and contact dynamics. Specifically, Natural Mujoco tasks are significantly more challenging, where they bridge the gap between simulated environment and real robots by replacing the simple backgrounds with natural videos sampled from ILSVRC dataset [37]. We present the results in Table 1 and analyze the quantitative results as follows.

Specifically, Natural Mujoco tasks brides the gap between simulated environment and real robots by replacing the simple backgrounds with natural videos sampled from ILSVRC dataset [37].

Model-based CVRL outperforms the model-free baseline. We observe that both CVRL reaches the best achievable performance, indicated by D4PG, the state-of-the-art model-free baseline trained for 20 times more steps (5×10^6 steps for CVRL and Dreamer, and 1×10^8 steps for D4PG). The learned latent world model successfully captures the real environment dynamics from pixel-level input, so that the trained actor-critic achieves comparable performance with the SOTA D4PG trained

[1] we use the official implementation of Dreamer and the SAC implementation from OpenAI baselines

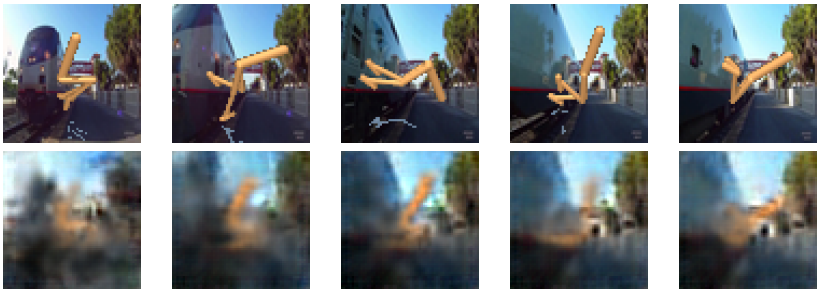
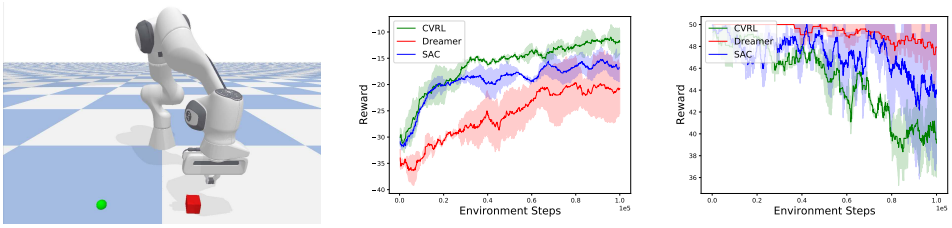


Figure 3: Generative models learn a latent world model by pixel level reconstruction, which is difficult when the observations are complex and variable. The first row shows the complex observations of natural Walker with varying video backgrounds, and the second row shows the reconstruction of generative models.

by ground truth trajectories. In contrast, given the same number of training steps, CVRL and Dreamer significantly outperform SAC on all tasks. This also suggests that the benefit of CVRL comes from the overall framework design, rather than the SAC.

CVRL is more robust to the natural observations. In Natural Mujoco tasks where the observations are more complex and variable, CVRL significantly outperforms the generative Dreamer in all cases. Although Dreamer achieves SOTA performance on the standard Mujoco tasks with relatively simple observations, its performance drops dramatically on natural Mujoco given complex observations introduced by the video background (*e.g.*, on walker-walk, 961.7 V.S. 206.6). CVRL, however, achieves comparable performance on 8 out of 10 tasks with or without the video background (*e.g.*, on walker-walk, 980.3 V.S. 941.5). This suggests that the contrastive learning, which avoids the pixel-level reconstruction, helps to learn a more robust latent world model than the generative models. Even with the variable complex video background, the learned latent world still successfully captures the underlying dynamics and achieves comparable performance with the simple observations. Besides, we visualize the reconstruction of generative models in Fig. 3. The reconstructions are blurry and lose information about the agent, which explains the failure of the generative Dreamer.



(a) Box Pushing Environment (b) Accumulated Reward (c) Execution Length

Figure 4: Box Pushing environment is challenging due to the contact physics, 3D scene, and the changing shadow introduced by the robot arm, which commonly exists in robot manipulation scenarios. CVRL achieves the highest performance with the shortest execution length.

4.2 Box Pushing

Robot pushing poses great challenges to reinforcement learning [38]. Specifically, the shadow and the occlusion by the robot arm often introduce confusing information to the perception module. We evaluate CVRL for robot pushing in PyBullet which gives an efficient and high-quality physics simulation. The task is to push the red box to the goal indicated by a green ball with an action of $a = (dx, dy)$ for the positional displacement. The positions of the box, the goal, and the lightning source are randomized at every episode. The negative Euclidean distance from the object to the goal is used as the reward, and the robot receives a reward of +1 when the box reaches the goal.

The results are presented in Fig. 4. We observe that CVRL achieves the highest reward with the shortest execution length, and it learns faster than Dreamer and SAC. Specifically, Dreamer struggles because reconstructing the shadow and the moving robot arm is challenging for generative learning. In contrast, although model-free SAC has failed on most of the Mujoco tasks with complex control

	CVRL	CVRL-generative	CVRL-no-MPC	CVRL-no-SAC	CVRL-reward-only
walker-walk	**941.5**	297.7	904.8	915.2	197.9
walker-run	**382.1**	71.4	343.2	378.3	115.4
cheetah-run	248.7	113.3	**430.1**	301.0	284.8
finger-spin	**850.4**	13.9	753.3	668.8	68.7
cartpole-balance	911.9	188.4	**996.3**	962.3	431.6
catpole-swingup	**413.8**	160.5	353.0	465.9	176.3
ball_in_cup-catch	894.2	254.8	881.4	**930.4**	368.7
reacher-easy	**909.1**	235.8	858.9	880.5	167.2
quadruped-walk	**878.7**	157.3	595.2	213.5	188.7
pendulum-swingup	**842.9**	19.7	831.5	813.3	20.8

Table 2: Ablation Studies on natural Mujoco tasks. CVRL generally outperforms all other variants.

dynamics, it outperforms Dreamer on Box Pushing by avoiding generative modeling, given the relatively simple action space. CVRL benefits from the sample efficiency of model-based learning, and it maintains the robustness to complex observations with contrastive learning.

4.3 Ablation Studies

We conduct a comprehensive ablation study on the Natural Mujoco tasks to better understand each proposed component. The results are presented in Table 2.

Contrastive variational latent world model is more robust to complex observations. CVRL-generative replaces the contrastive learning with a generative model that performs image-level reconstruction. Unlike Dreamer, CVRL-generative only differs from the CVRL in the parameterization of the representation learning method, and still has the rest of the proposed components. However, its performance degrades on all cases compared to CVRL. This aligns with our previous observation that contrastive learning is more robust given complex observations.

Latent guided MPC improves the reasoning ability for long-horizon behaviors. CVRL-no-MPC uses only the actor-critic for decision making. We observe it performs worse than CVRL on 8 out of 10 tasks, especially on some of the challenging tasks, *e.g.*, cartpole-swingup and quadruped-walk, where multi-step reasoning is required. The latent guided MPC improves the overall performance of CVRL.

The hybrid actor-critic is robust given complex observations. CVRL-no-SAC removes the SAC during actor-critic learning. Its performance drops on certain cases, compared to CVRL (on cheetah-run, 497.3 V.S. 301.0 and finger-spin, 987.1 V.S. 668.8). This is because when the useful features are highly coupled with variable and complex background, learning an accurate latent world model becomes difficult, even for CELBO. With ground-truth trajectories, SAC can provide accurate training signals to compensate for the compositional error of the latent world model.

Reward signal alone is not enough for learning the latent world model. CVRL-reward-only uses only reward prediction for representation learning. Its performance drops in all cases. This suggests that the robustness of CVRL comes from the contrastive learning, rather than only the reward learning.

5 Conclusions

We introduce CVRL, a framework for robust MBRL under natural complex observations. CVRL learns a contrastive variational world model with CELBO objective, a contrastive learning alternative to the ELBO, which avoids reconstructing the complex observations. CVRL lerans a robust hybrid actor-critic and uses guided MPC for decision making. It achieves comparable performance with the SOTA methods on 10 challenging Mujoco control tasks, and significantly outperforms SOTA methods on more challenging Natural Mujoco tasks and Box Pushing tasks.

However, CVRL does not perform as well as Dreamer on some tasks on standard Mujcoo tasks (walker-run and cheetah-run), where the observation is simple. While contrastive learning is robust to complex observations, its quality highly depends on the sampling strategy of negative samples. Currently we use a very simple strategy. Further work may consider smarter sampling strategies, *e.g.*, learning to sample using meta-learning.

References

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

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

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

[4] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[5] W. Bosworth, S. Kim, and N. Hogan. The mit super mini cheetah: A small, low-cost quadrupedal robot for dynamic locomotion. In *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–8. IEEE, 2015.

[6] J. F. Allen and J. A. Koomen. Planning using a temporal world model. In *Proceedings of the Eighth international joint conference on Artificial intelligence-Volume 2*, pages 741–747, 1983.

[7] K. Basye, T. Dean, J. Kirman, and M. Lejter. A decision-theoretic approach to planning, perception, and control. *IEEE Expert*, 7(4):58–65, 1992.

[8] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pages 2450–2462, 2018.

[9] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.

[10] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for POMDPs. In *Proceedings of the International Conference on Machine Learning*, pages 2117–2126, 2018.

[11] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

[12] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.

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

[14] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2019.

[15] K. Doya, K. Samejima, K.-i. Katagiri, and M. Kawato. Multiple model-based reinforcement learning. *Neural computation*, 14(6):1347–1369, 2002.

[16] P. Karkus, D. Hsu, and W. S. Lee. QMDP-net: Deep learning for planning under partial observability. In *Advances in Neural Information Processing Systems*, pages 4694–4704, 2017.

[17] X. Ma, P. Karkus, D. Hsu, and W. S. Lee. Particle filter recurrent neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI, 2020*, pages 5101–5108.

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

[19] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in neural information processing systems*, pages 5074–5082, 2016.

[20] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[21] A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012.

[22] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3318–3325, 2013.

[23] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[24] T. Kipf, E. van der Pol, and M. Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.

[25] P. Karkus, D. Hsu, and W. S. Lee. Particle filter networks with application to visual localization. In *Proceedings of the Conference on Robot Learning*, pages 169–178, 2018.

[26] X. Ma, P. Karkus, D. Hsu, W. S. Lee, and N. Ye. Discriminative particle filter reinforcement learning for complex partial observations. In *International Conference on Learning Representations*, 2020.

[27] P. Karkus, A. Angelova, V. Vanhoucke, and R. Jonschkowski. Differentiable mapping networks: Learning structured map representations for sparse visual localization. *arXiv preprint arXiv:2005.09530*, 2020.

[28] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pages 2980–2988, 2015.

[29] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

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

[31] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

[32] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[34] P. Karkus, X. Ma, D. Hsu, L. P. Kaelbling, W. S. Lee, and T. Lozano-Pérez. Differentiable algorithm networks for composable robot learning. *arXiv preprint arXiv:1905.11602*, 2019.

[35] R. Tedrake. Underactuated robotics: Learning, planning, and control for efficient and agile machines course notes for mit 6.832. *Working draft edition*, 3, 2009.

[36] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. Tb, A. Muldal, N. Heess, and T. Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[38] J. K. Li, W. S. Lee, and D. Hsu. Push-net: Deep planar pushing for objects with unknown physical properties. In *Robotics: Science and Systems*, volume 14, pages 1–9, 2018.

[39] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*, 2018.

A Algorithm Details

A.1 Latent Imagination

CVRL first generates the imagined trajectories using the learned world model parameterized by θ. Specifically, given a state $\tilde{s}_{\tau-1}$, we sample the next imagined state by $\tilde{s}_\tau \sim p_\theta(\tilde{s}_\tau \mid \tilde{s}_{\tau-1}, \tilde{a}_{\tau-1})$, which further generates a reward $\tilde{r}_\tau \sim p_\theta(\tilde{r}_\tau \mid \tilde{s}_\tau)$ and the next action $\tilde{a}_\tau \sim q_\eta(\tilde{a}_\tau \mid \tilde{s}_\tau)$. We repeat this process until we have an imagined trajectory $\{\tilde{s}_\tau, \tilde{a}_\tau, \tilde{r}_\tau\}_{\tau=t}^{t+H}$.

A.2 Value Estimation of Dreamer

Dreamer estimates the value $\tilde{V}(s_\tau)$ of imagined trajectories using the following equations:

$$\tilde{V}_N^k(\tilde{s}_\tau) = \mathbb{E}_{p_\theta, q_\eta}\left(\sum_{n=\tau}^{h-1} \gamma^{n-\tau}\tilde{r}_n + \gamma^{h-\tau} v_\psi(\tilde{s}_h)\right), \quad \text{where } h = \min(\tau + k, t + H)$$

$$\tilde{V}_\lambda(\tilde{s}_\tau) = (1-\lambda)\sum_{n=1}^{H-1} \lambda^{n-1}\tilde{V}_N^n(\tilde{s}_\tau) + \lambda^{H-1}\tilde{V}_N^H(\tilde{s}_\tau)$$

$\tilde{V}_N^k(\tilde{s}_\tau)$ estimates the value of $\tilde{s}_\tau$ using the rewards of k steps of rollouts and the value function estimate v_ψ of the last state. Dreamer ues $\tilde{V}_\lambda(\tilde{s}_\tau)$ as the final value estimation, which is an exponentially-weighted average of different k-step rollouts to tradeoff the bias and variance.

A.3 Latent Guided MPC

Originally, for each state s_t, the actor network $q_\eta(a_t \mid s_t)$ generates the action which maximizes the long-horizon accumulated reward. However, the approximation highly depends on the quality of the learned world model and might have difficulties approximating complex policies. Most importantly, it lacks the reasoning ability to adapt to variable environments.

We use the shooting method for MPC with differentiable world model. Specifically, we use stochastic gradient ascent to optimize the action sequences to output high accumulated reward. During execution, for each obsevation o_t, previous state s_{t-1} and action a_{t-1}, we encode / propose the current state by $q_\phi(s_t \mid o_{\leq t}, a_{\leq t})$. Next, we perform latent imagination and sample the imagined trajectories $\{\tilde{s}_\tau, \tilde{a}_\tau, \tilde{r}_\tau\}_{\tau=t}^{t+H}$ and estimate $\tilde{V}_\lambda(s_t)$. As $\tilde{V}_\lambda(s_t)$ is computed using predicted rewards and value estimations, which are conditioned on the action sequences, we can backpropagate the gradients from $\tilde{V}_\lambda(s_t)$ to the actions. We update the actions by

$$\tilde{a}'_\tau = \tilde{a}_\tau + \nabla_{\tilde{a}_\tau}\tilde{V}_\lambda(s_t)$$

We repeat this for all actions and return the first action after update.

Our latent guided MPC is similar to the planning algorithm used in DPI-Net [39]. The difference is that DPI-Net requires a pre-defined observation of the goal to compute the loss, whereas CVRL directly maximizes the accumulated reward and alleviate this assumption.

B Implementation Details

B.1 Negative Sample Selection

We adopt a simple strategy to generate negative samples. We sample a batch of sequences $\{o_{1:T}^{(i)}, a_{1:T}^{(i)}, r_{1:T}^{(i)}\}_{i=1}^B$ from a replay buffer, where T is the sequence length and B is the batch size. For each state-observation pair (s_t, o_t), we treat the other $B * T - 1$ observations $\{o'\}$ in the same batch as negative samples. An intuition of this choice is that: 1) by contrasting $(s_t^{(i)}, o_t^{(i)})$ with $(s_t^{(i)}, o_{t'}^{(j)})$ where $j \neq i$ and $t' \in [1, T]$, CELBO learns to identify invariant features of the task given variable visual features; 2) by contrasting $(s_t^{(i)}, o_t^{(i)})$ with $(s_t^{(i)}, o_{t'}^{(i)})$ where $t' \neq t$, CELBO learns to model the temporal dynamics of the task. We found this simple strategy works well in practice.

B.2 Hardware and Software.

We train all models on single NVidia RTX 2080Ti GPUs with Intel Xeon Gold 5220 CPU @ 2.20GHz. We implement all models with Tensorflow 2.2.0 and Tensorflow Probability 0.10.0. Specifically, our code is developed based on the official Tensorflow implementation of Dreamer, but heavily modified. We use the official implementation of Dreamer as our baseline, and we use the SAC implementation of OpenAI baselines. For all methods, we share certain structure, including the encoder, RSSM model and the actor-critic networks to make it a fair comparison.

B.3 Observation Encoder.

We use an encoder of 4 convolutional layers for image observations, which have a fixed kernel size of 4 with increasing channel numbers: 32, 65, 128, 256. We do not encode the actions again and directly concatenate it with the states.

B.4 RSSM.

We use a stochastic state s_t with size 30 and a deterministic state with size 200. The deterministic update function is parameterized using a GRU and the for the stochastic part, we learn the mean and variance of s_t using two fully connected layers with size 200 and 30.

B.5 Contrastive Learning.

In contrastive learning, we learn the compatibility between state s_t and observation o_t with a function $f_\theta(s_t, o_t)$. In our implementation, we first encode both o_t and s_t by two separate fully connected layers with size 200, then we compute the value of $f_\theta(s_t, o_t) = \exp(z_t^{\mathrm{T}} W_\theta s'_t)$, where z_t and s'_t are the embeddings of the observation and the state, and w_θ is a 200×200 matrix.

B.6 Actor-Critic.

For the actor network, we use 4 fully connected layer which takes in the concatenation of s_t and h_t as input, with intermediate hidden dimension of 400, and output the corresponding action, with tanh as the activation function. Specifically, a transformed distribution is used to achieve differentiable sampling. For the value network, 3 fully connected layers are used with hidden dimension of 400 and output dimension of 1. In addition, SAC needs additional Q-value network during training. For models needs SAC, we use 2 Q-value network with similar structure, except that the input is a concatenation of s_t, h_t and a_t.

B.7 Model Learning.

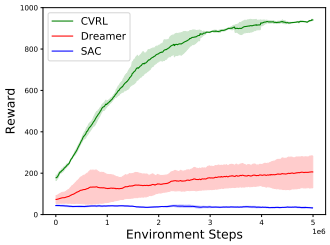
We train CVRL by 4 separate optimizers for different part of the network: model optimizer, value optimizer, actor optimizer and SAC optimizer. For all optimizers, we use Adam optimizer in our implementation with different learning rate. Model optimizer updates all contrastive variational world model dynamics by representation learning defined in Eqn. 6 with learning rate 6×10^{-4}; value optimizer updates only value network parameters with learning rate 8×10^{-5}; actor optimizer updates the actor parameters with learning rate 8×10^{-5}; SAC optimizer updates the actor parameters and the two Q-value network parameters with learning rate 8×10^{-5}.

B.8 Latent Guided MPC.

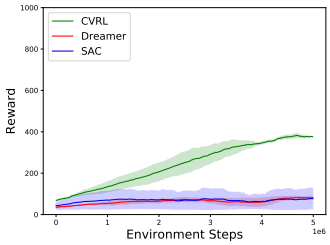
In latent guided MPC, we unroll for 15 steps and update the actions by standard SGD with learning rate 0.003.

C Additional Visualizations

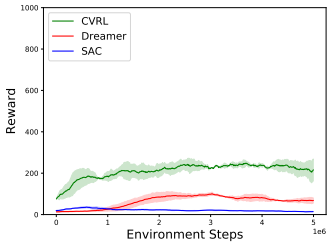
C.1 Natural Mujoco Tasks



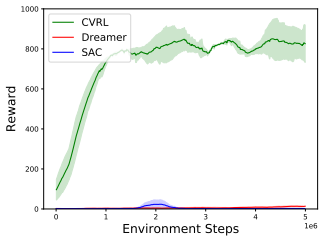
(a) Natural Walker Walk



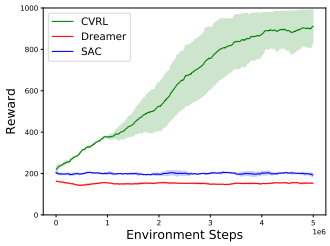
(b) Natural Walker Run



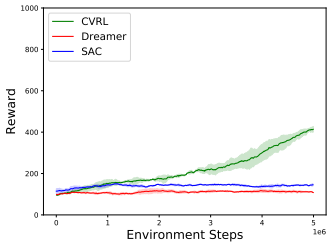
(c) Natural Cheetah Run



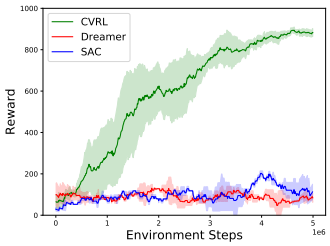
(d) Natural Finger Spin



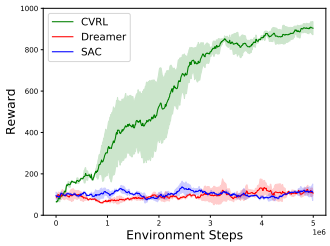
(e) Natural Cartpole Balance



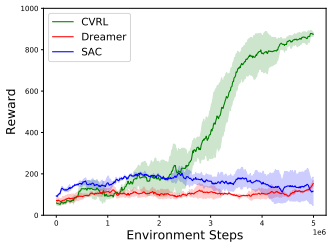
(f) Natural Cartpole Swingup



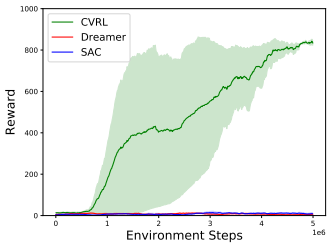
(g) Natural Cup Catch



(h) Natural Reacher Easy

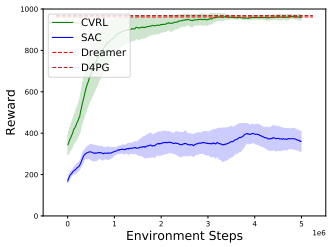


(i) Natural Quadruped Walk

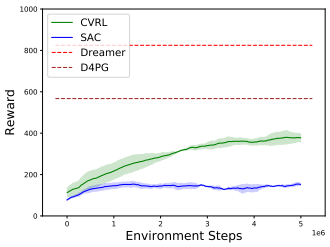


(j) Natural Pendulum Swingup

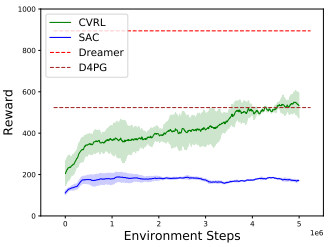
C.2 Standard Mujoco Tasks



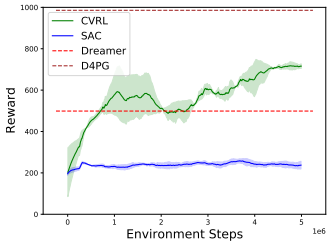
(a) Standard Walker Walk



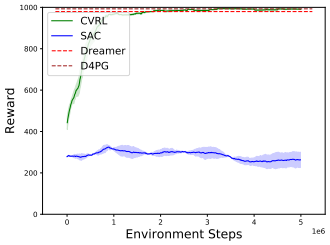
(b) Standard Walker Run



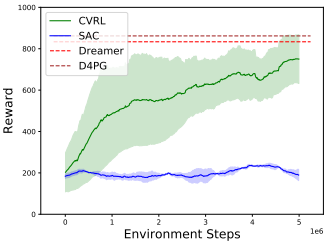
(c) Standard Cheetah Run



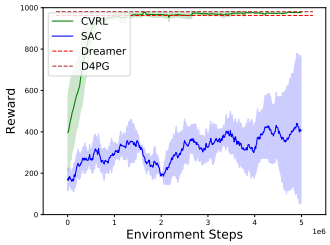
(d) Standard Finger Spin



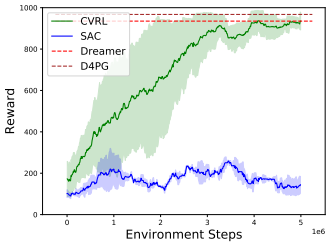
(e) Standard Cartpole Balance



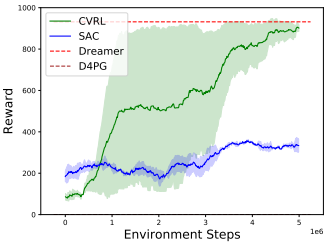
(f) Standard Cartpole Swingup



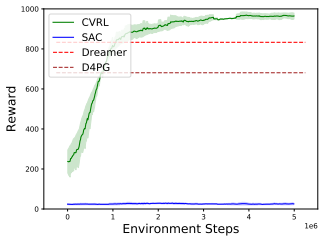
(g) Standard Cup Catch



(h) Standard Reacher Easy



(i) Standard Quadruped Walk



(j) Standard Pendulum Swingup