Combining Reconstruction and Contrastive Methods for Multimodal Representations in RL

Philipp Becker
philipp.becker@kit.edu
Karlsruhe Institute of Technology
FZI Research Center for Information Technology

Sebastian Mossburger
Karlsruhe Institute of Technology

Fabian Otto
Bosch Center for Artificial Intelligence
University of Tübingen

Gerhard Neumann
Karlsruhe Institute of Technology
FZI Research Center for Information Technology

Abstract

Learning self-supervised representations using reconstruction or contrastive losses improves performance and sample complexity of image-based and multimodal reinforcement learning (RL). Here, different self-supervised loss functions have distinct advantages and limitations depending on the information density of the underlying sensor modality. Reconstruction provides strong learning signals but is susceptible to distractions and spurious information. While contrastive approaches can ignore those, they may fail to capture all relevant details and can lead to representation collapse. For multimodal RL, this suggests that different modalities should be treated differently based on the amount of distractions in the signal. We propose *Contrastive Reconstructive Aggregated representation Learning (CoRAL)*, a unified framework enabling us to choose the most appropriate self-supervised loss for each sensor modality and allowing the representation to better focus on relevant aspects. We evaluate *CoRAL's* benefits on a wide range of tasks with images containing distractions or occlusions, a new locomotion suite, and a challenging manipulation suite with visually realistic distractions. Our results show that learning a multimodal representation by combining contrastive and reconstruction-based losses can significantly improve performance and solve tasks that are out of reach for more naive representation learning approaches and other recent baselines.

1 Introduction

Most representation learning approaches for reinforcement learning (RL) (Hafner et al., 2020; 2021; 2023; Laskin et al., 2020; Lee et al., 2020; Yarats et al., 2021b; Zhang et al., 2020; Zhu et al., 2023; Deng et al., 2022) focus on images. Here, the challenge lies in compressing relevant information while not getting distracted by potentially irrelevant aspects. Yet, most agents in realistic scenarios can directly observe their internal states using sensors in the actuators, inertial measurement units,

Code: https://github.com/pbecker93/CoRAL/, Project Page: https://pbecker93.github.io/coral_pp/

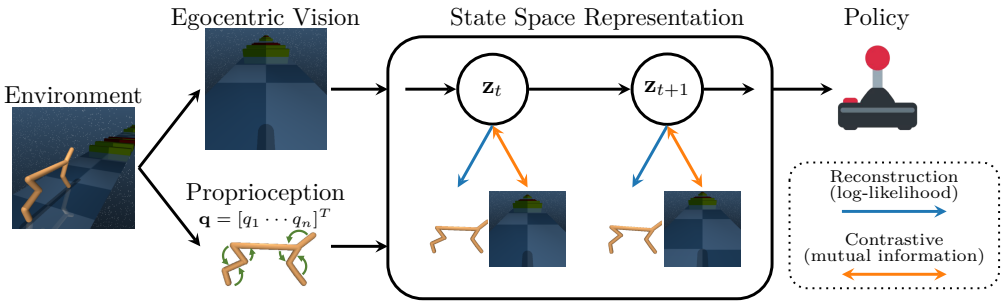

Figure 1: *Contrastive Reconstructive Aggregated representation Learning (CoRAL)* learns multimodal state space representations of all available sensors using a combination of reconstruction-based and contrastive objectives. Building on the insight that we can exchange likelihood-based reconstruction with contrastive approaches using mutual information, allows us to choose an appropriate loss function for each modality. Motivated by both a variational and predictive coding viewpoint, *CoRAL* helps model-free and model-based agents to excel in challenging tasks that require information fusion from sensors with different properties such as images and proprioception.

and force and torque sensors. Including this low-dimensional and concise proprioceptive sensing in representation learning can improve representation quality and downstream RL performance. For such multimodal representations, State Space Models (Murphy, 2012) are a natural choice as they lend themselves to accumulating information across multiple sensors and time. Previous works suggest using either reconstruction (Hafner et al., 2019; 2021) or contrastive methods (Hafner et al., 2020; Ma et al., 2020; Nguyen et al., 2021; Srivastava et al., 2021), both with their individual strengths and weaknesses. While reconstruction provides an informative learning signal, it may fail to learn good representations if observations are noisy or contain distracting elements (Zhang et al., 2020; Ma et al., 2020; Deng et al., 2022). In such cases, contrastive methods can ignore irrelevant parts of the observation and still learn valuable representations. However, they are prone to representation collapse and often struggle to learn accurate dynamics (Ma et al., 2020). We argue that the different properties of sensors, such as images and proprioception, suggest using different self-supervised loss functions for each modality.

We propose *Contrastive Reconstructive Aggregated representation Learning (CoRAL)* to combine contrastive and reconstruction-based approaches. *CoRAL* builds on state space representations and allows us to select the best-suited loss function for each modality, for example, reconstruction-based loss functions for concise, low-dimensional proprioception and contrastive losses for images with distractions. Learning such state space representations can be theoretically motivated using a variational inference (Hafner et al., 2019; Ma et al., 2020) or a predictive coding (Oord et al., 2018; Nguyen et al., 2021; Srivastava et al., 2021) viewpoint, which results in two instances of *CoRAL*. For both paradigms, *CoRAL* relies on the insight that we can replace likelihood-based reconstruction terms with contrastive losses based on mutual information, which allows for a principled combination of the two (Hafner et al., 2020; Ma et al., 2020). Fig. 1 provides an overview of the approach.

We integrate *CoRAL* into model-free and model-based RL to systematically assess the effects of learning multimodal representations by selecting appropriate losses. We evaluate on DeepMind Control (DMC) Suite (Tassa et al., 2018) tasks which we make more difficult by adding *Video Backgrounds* (Zhang et al., 2020; Nguyen et al., 2021) and *Occlusions*. Furthermore, we use a new *Locomotion* suite where agents must fuse proprioception and egocentric vision to move while navigating obstacles. Finally, we consider a novel challenging *Manipulation* suite consisting of static and mobile manipulation tasks with varying object geometries, built on ManiSkill2 (Gu et al., 2023). Here, the agents must combine proprioception and different visual modalities, such as color and depth, to move, navigate, and interact with varying objects in visually realistic environments. These experiments show that learning multimodal representations using the best-suited loss for each modality improves over other methods combining both modalities, such as representation learning

with a single loss and concatenating image representations with proprioception. *CoRAL* tends to work better than recent baselines on the *Video Background* and *Occlusion* tasks and allows significant performance gains in the challenging *Locomotion* and *Manipulation* tasks. Furthermore, *CoRAL* significantly improves model-based approaches with contrastive image representations, which are known to perform worse than reconstruction-based approaches (Hafner et al., 2020; Ma et al., 2020). Finally, we show the strengths of both instances of *CoRAL*. *Variational CoRAL* excels in tasks where the main challenge is filtering out irrelevant distractions from images, while *Predictive CoRAL* performs better in tasks that require propagating information over many timesteps.

To summarize our contributions: **(i)** We propose *CoRAL*, a general framework for multimodal representation learning for RL which allows using the best-suited self-supervised loss for each modality using the interchangeability of likelihood-based reconstruction and contrastive losses based on mutual information. **(ii)**We instantiate two versions of *CoRAL* using state space representations, namely *Variational-CoRAL* and *Predictive-CoRAL*, which are inspired by variational and contrastive predictive coding viewpoints, respectively. **(iii)** We systematically show their effectiveness on a diverse set of 26 tasks, across the *Video Backgrounds*, *Occlusions Locomotion*, and *Manipulation* suites.

2 Related Work

Representations for Reinforcement Learning. Many recent approaches use ideas from generative (Wahlström et al., 2015; Watter et al., 2015; Banijamali et al., 2018; Lee et al., 2020; Yarats et al., 2021b) and self-supervised representation learning (Zhang et al., 2020; Laskin et al., 2020; Yarats et al., 2021a; Stooke et al., 2021; You et al., 2022) to improve performance, sample efficiency, and generalization of RL from images. Those based on *Recurrent State Space Models (RSSMs)* (Hafner et al., 2019) are particularly relevant for this work. When proposing the *RSSM*, Hafner et al. (2019) used a generative approach. They formulated their objective as auto-encoding variational inference (Kingma & Welling, 2013), which trains the representation by reconstructing observations. Such reconstruction-based approaches have limitations with observations containing noise or many task-irrelevant details. As a remedy, Hafner et al. (2020) proposed a contrastive alternative based on mutual information and the InfoNCE estimator (Poole et al., 2019). Ma et al. (2020) refined this approach and improved results by modifying the policy learning mechanism. Using a different motivation, namely contrastive predictive coding (Oord et al., 2018), Okada & Taniguchi (2021); Nguyen et al. (2021); Srivastava et al. (2021); Okada & Taniguchi (2022) proposed alternative contrastive learning objectives for *RSSMs*. In this work, we leverage the variational and predictive coding paradigms and show that *CoRAL* improves performance for both. Fu et al. (2021); Wang et al. (2022) propose further factorizing the *RSSM*'s latent variable to disentangle task-relevant and task-irrelevant information. However, unlike contrastive approaches, they explicitly model the task-irrelevant parts instead of ignoring them, which can impede performance if the distracting elements become too complex to model. Zhu et al. (2023) propose a relaxed variational information bottleneck (Alemi et al., 2016) approach which trains *RSSMs* solely by predicting rewards and enforcing posterior predictability using a KL term. Other recent approaches for learning *RSSMs* include using prototypical representations (Deng et al., 2022) or masked reconstruction (Seo et al., 2022).

Sensor Fusion in Reinforcement Learning. Many application-driven approaches to visual RL for robots use proprioception to solve their specific tasks (Finn et al., 2016; Levine et al., 2016; Kalashnikov et al., 2018; Xiao et al., 2022; Fu et al., 2022). Yet, they usually do not use explicit representation learning or concatenate image representations and proprioception. Several notable exceptions use *RSSMs* with images and proprioception (Wu et al., 2022; Becker & Neumann, 2022; Hafner et al., 2022; 2023). Furthermore, Seo et al. (2023) learn world models using multiple images from different viewpoints. However, all these approaches focus on purely reconstruction-based representation learning. Srivastava et al. (2021) use images and proprioceptive information using contrastive predictive coding for both modalities. Opposed to all of these works, we propose combining contrastive approaches with reconstruction.

Multimodal Representation Learning. Representation learning from multiple modalities has widespread applications in general machine learning, where methods such as *CLIP* (Radford et al., 2021) combine language concepts with the semantic knowledge of images and allow language-based image generation (Ramesh et al., 2022). For robotics, Brohan et al. (2022); Mees et al. (2022); Driess et al. (2023); Shridhar et al. (2022; 2023) combine language models with the robot's perception for natural language-guided manipulation tasks using imitation learning. In contrast, *CoRAL* assumes an online RL setting and focuses on different modalities, namely images and proprioception.

3 Combining Contrastive Approaches and Reconstruction for State Space Represntations

Given trajectories of observations $\mathbf{o}_{1:T} = \{\mathbf{o}_t\}_{t=1:T}$ and actions $\mathbf{a}_{1:T} = \{\mathbf{a}_t\}_{t=1:T}$ we aim to learn a state representation that is well suited for RL. We assume the observations stem from K different sensors, $\mathbf{o}_t = \{\mathbf{o}_t^{(k)}\}_{k=1:K}$, where the individual $\mathbf{o}_t^{(k)}$ only contain partial information about the state. Further, even $\mathbf{o}_t$ may not contain all necessary information for optimal acting, i. e., the environment is partially observable, and the representation has to accumulate information over time. Our goal is to learn a concise, low dimensional representation $\phi(\mathbf{o}_{1:t}, \mathbf{a}_{1:t-1})$ that accumulates all relevant information until time step t. We provide this representation to a policy $\pi(\mathbf{a}_t|\phi(\mathbf{o}_{1:t}, \mathbf{a}_{1:t-1}))$ which aims to maximize the expected return in a given RL problem. In this setting, the policy's final return and the sample complexity of the entire system determine what constitutes a *good* representation.

State Space Models (SSMs) (Murphy, 2012) naturally lend themselves to sensor fusion and information accumulation problems. We assume a latent state variable, $\mathbf{z}_t$, which evolves according to a Markovian dynamics $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{a}_t)$ given an action $\mathbf{a}_t$. Furthermore, we assume the K observations at each time step are conditionally independent given the latent state, resulting in an observation model $p(\mathbf{o}_t|\mathbf{z}_t) = \prod_{k=1}^{K} p^{(k)}(\mathbf{o}_t^{(k)}|\mathbf{z}_t)$. The initial state is distributed according to $p(\mathbf{z}_0)$. Here, the belief over the latent state, taking into account all previous actions as well as previous and current observations $p(\mathbf{z}_t|\mathbf{a}_{1:t-1}, \mathbf{o}_{1:t})$ can be used as the representation. Yet, computing $p(\mathbf{z}_t|\mathbf{a}_{1:t-1}, \mathbf{o}_{1:t})$ analytically is intractable for models of relevant complexity and we use a variational approximation $\phi(\mathbf{o}_{1:t}, \mathbf{a}_{1:t-1}) \hat{=} q(\mathbf{z}_t|\mathbf{a}_{1:t-1}, \mathbf{o}_{1:t})$. This variational approximation also plays an integral part during training and is thus readily available as input for the policy.

We instantiate the generative SSM and the variational distribution using a *Recurrent State Space Model (RSSM)* (Hafner et al., 2019), which splits the latent state $\mathbf{z}_t$ into a stochastic and a deterministic part. Following Hafner et al. (2019; 2020), we assume the stochastic part of the *RSSM*'s latent state to be Gaussian. While the original *RSSM* only has a single observation model $p(\mathbf{o}_t|\mathbf{z}_t)$, we extend it to K models, one for each observation modality. The variational distribution takes the deterministic part of the state together with the K observations $\mathbf{o}_t = \{\mathbf{o}_t^{(k)}\}_{k=1:K}$ and factorizes as $q(\mathbf{z}_{1:t}|\mathbf{o}_{1:t}, \mathbf{a}_{1:t-1}) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t)$. To account for multiple observations instead of one, we first encode each observation individually using a set of K encoders, concatenate their outputs, and provide the result to the *RSSM*. Finally, we also learn a reward model $p(r_t|\mathbf{z}_t)$ to predict the reward from the representation. Following the findings of Srivastava et al. (2021) and Tomar et al. (2023) we also include reward prediction to learn the representations for model-free agents.

3.1 Learning the State Space Representation

We propose to combine reconstruction-based and contrastive approaches to train our representations. Training *RSSMs* can be based on either a variational viewpoint (Hafner et al., 2020; Ma et al., 2020) or a contrastive predictive coding (Oord et al., 2018) viewpoint (Nguyen et al., 2021; Srivastava et al., 2021). We investigate both approaches, as neither decisively outperforms the other.

Originally, Hafner et al. (2019) proposed leveraging a fully generative approach for *RSSMs*. Building on the stochastic variational autoencoding Bayes framework (Kingma & Welling, 2013; Sohn et al.,

2015), they derive a variational lower-bound objective

$$\mathbb{E}_{p(\mathbf{o}_{1:T},\mathbf{a}_{1:T})}\left[\log p(\mathbf{o}_{1:T}|\mathbf{a}_{1:T})\right] \geq \sum_{t=1}^{T} \mathbb{E}_{\hat{q}(\cdot)}\left[\log p(\mathbf{o}_t|\mathbf{z}_t) - \mathrm{KL}\left[q(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1},\mathbf{o}_t) \parallel p(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1})\right]\right],$$

where $\hat{q}(\cdot) = q(\mathbf{z}_{t-1:t}|\mathbf{o}_{1:t},\mathbf{a}_{1:t})p(\mathbf{o}_{1:t},\mathbf{a}_{1:t})$, i. e.,the variational distribution and sub-trajectories from a replay buffer. After inserting our assumption that each observation factorizes into K independent observations, i.e., $\log p(\mathbf{o}_t|\mathbf{z}_t) = \sum_{k=1}^{K} \log p^{(k)}(\mathbf{o}_t^{(k)}|\mathbf{z}_t)$,and adding a term for reward prediction, this results in

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{q}(\cdot)}\left[\sum_{k=1}^{K} \log p^{(k)}(\mathbf{o}_t^{(k)}|\mathbf{z}_t) + \log p(r_t|\mathbf{z}_t) - \mathrm{KL}\left[q(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1},\mathbf{o}_t) \parallel p(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1})\right]\right]. \tag{1}$$

Optimizing this bound using the reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014) and stochastic gradient descent simultaneously trains the variational distribution and all parts of the generative model. While this approach can be highly effective, reconstructing high-dimensional, noisy observations can also cause issues. First, it requires introducing large observation models. These observation models are unessential for the downstream task and are usually discarded after training. Second, the reconstruction forces the model to capture all details of the observations, which can lead to highly suboptimal representations if images contain task-irrelevant distractions.

Contrastive Variational Learning (CV) can remedy these problems. To introduce contrastive terms, we replace the individual reconstruction terms in Equation 1 with mutual information (MI) terms $I(\mathbf{o}_t^{(k)}, \mathbf{z}_t)$ by adding and subtracting $\log p^{(k)}(\mathbf{o}^{(k)})$ (Hafner et al., 2020; Ma et al., 2020)

$$\mathbb{E}_{\hat{q}(\cdot)}\left[\log p^{(k)}(\mathbf{o}_t^{(k)}|\mathbf{z}_t)\right] = \mathbb{E}_{\hat{q}(\cdot)}\left[\log \frac{p^{(k)}(\mathbf{o}_t^{(k)}|\mathbf{z}_t)}{p(\mathbf{o}_t^{(k)})} + \log p(\mathbf{o}_t^{(k)})\right] = \mathcal{I}(\mathbf{o}_t^{(k)}, \mathbf{z}_t) + c. \tag{2}$$

Intuitively, the MI measures how informative a given latent state is about the corresponding observations. Thus, maximizing it leads to similar latent states for similar sequences of observations and actions. While we cannot analytically compute the MI, we can estimate it using the InfoNCE bound (Oord et al., 2018; Poole et al., 2019). Doing so eliminates the need for generative reconstruction. It instead only requires a discriminative approach based on a score function $f_v^{(k)}(\mathbf{o}_t^{(k)}, \mathbf{z}_t) \mapsto \mathbb{R}_+$. This score function measures the compatibility of pairs of observations and latent states. It shares large parts of its parameters with the *RSSM*. We refer to Appendix B for details on the exact parameterization. This methodology allows the mixing of reconstruction and mutual information terms for the individual sensors, resulting in a generalization of Equation 1,

$$\sum_{t=1}^{T}\sum_{k=1}^{K} \mathcal{L}_v^{(k)}(\mathbf{o}_t^{(k)}, \mathbf{z}_t) + \mathbb{E}_{\hat{q}(\cdot)}\left[\log p(r_t|\mathbf{z}_t) - \mathrm{KL}\left[q(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1},\mathbf{o}_t) \parallel p(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1}])\right]\right]. \tag{3}$$

Here $\mathcal{L}_v^{(k)}$ is either $\mathbb{E}_{\hat{q}(\cdot)}\left[\log p(\mathbf{o}_t^{(k)}|\mathbf{z}_t)\right]$ or $\mathcal{I}(\mathbf{o}_t^{(k)}, \mathbf{z}_t)$. As we show in Section 4 choosing the terms corresponding to the properties of the corresponding modality can often improve performance.

Contrastive Predictive Coding (CPC) (Oord et al., 2018) provides an alternative to the variational approach. The idea is to maximize the MI between the previous latent variable $\mathbf{z}_{t-1}$ and the observation $\mathbf{o}^{(k)}$, i. e., $I(\mathbf{o}_t^{(k)}, \mathbf{z}_{t-1})$. While this approach seems similar to contrastive variational learning, we use the previous latent state $\mathbf{z}_{t-1}$ instead of the current $\mathbf{z}_t$ to estimate the MI. Thus, we explicitly predict one time step ahead to compute the loss. As we use the *RSSM's* dynamics model for the prediction, this formalism provides a training signal to the dynamics model. However, Levine et al. (2019); Shu et al. (2020); Nguyen et al. (2021) discuss how this signal alone is insufficient for model-based RL. Srivastava et al. (2021) show that similar ideas also benefit model-free RL and we follow their approach by regularizing the objective using KL-term from Equation 1 weighted with a small factor β. Additionally, we can turn individual contrastive MI terms into reconstruction terms

for suitable sensor modalities by reversing the principle of Equation 2. Including reward prediction, this results in the following maximization objective

$$\sum_{t=1}^{T}\sum_{k=1}^{K}\mathcal{L}_p^{(k)}(\mathbf{o}_{t+1}^{(k)},\mathbf{z}_t)+\mathbb{E}_{\hat{q}(\cdot)}\left[\log p(r_t|\mathbf{z}_t)-\beta\mathrm{KL}\left[q(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1},\mathbf{o}_t)\parallel p(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1})\right]\right], \tag{4}$$

where $\mathcal{L}_p^{(k)}$ is either the one-step ahead likelihood $\mathbb{E}_{\hat{q}(\cdot)}\left[\log p(\mathbf{o}_t^{(k)}|\mathbf{z}_{t-1})\right]$ or an InfoNCE estimate of $\mathcal{I}(\mathbf{o}_t^{(k)},\mathbf{z}_{t-1})$ using a score function $f_p^{(k)}(\mathbf{o}_t^{(k)},\mathbf{z}_{t-1})\mapsto\mathbb{R}_+$. From an implementation viewpoint, the resulting approach differs only slightly from the variational contrastive one. For CPC approaches, we use a sample from the *RSSM's* dynamics $p(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1})$ and for contrastive variational approaches we use a sample from the variational distribution $q(\mathbf{z}_t|\mathbf{z}_{t-1},\mathbf{a}_{t-1},\mathbf{o}_t)$ as input to the score function or decoder.

Estimating Mutual Information with InfoNCE. We estimate the mutual information (MI) using b mini-batches of sub-sequences of length l. After computing the latent estimates, we get $N=b\cdot l$ pairs $(\mathbf{o}_i,\,\mathbf{z}_i)$, i. e., we use both samples from the elements of the batch as well as all the other time steps within the sequence as negative samples. Using those, the symmetry of MI, the InfoNCE bound (Poole et al., 2019), and either $f=f_v^{(k)}$ or $f=f_p^{(k)}$, we can estimate the MI as

$$\mathcal{I}(\mathbf{o}_i,\mathbf{z}_i)\geq 0.5\left(\sum_{i=1}^{N}\log\frac{f(\mathbf{o}_i,\mathbf{z}_i)}{\sum_{j=1}^{N}f(\mathbf{o}_i,\mathbf{z}_j)}+\log\frac{f(\mathbf{o}_i,\mathbf{z}_i)}{\sum_{j=1}^{N}f(\mathbf{o}_j,\mathbf{z}_i)}\right).$$

3.2 Learning to Act Based on the Representation

Our representations are amenable to both model-free and model-based reinforcement learning. For the former, we use Soft Actor-Critic (SAC) (Haarnoja et al., 2018) on top of the representation by providing the deterministic part of the latent state and the mean of the stochastic part as input to both the actor and the critic. For the latter, we use *latent imagination* (Hafner et al., 2020), which propagates gradients through the learned dynamics model to optimize the actor. In both cases, we alternatingly update the *RSSM*, actor, and critic for several steps before collecting a new sequence in the environment. The *RSSM* uses only the representation learning loss and gets neither gradients from the actor nor the critic.

4 Experiments

Building on the previously introduced methodology, we build two versions of *Contrastive Reconstructive Aggregated representation Learning (CoRAL)* differing in the state space representation objective. *Variational CoRAL* (*V-CoRAL*), using the variational objective (Equation 3) and *Predictive CoRAL* (*P-CoRAL*), using the predictive coding objective (Equation 4). We evaluate the performance of *CoRAL* by using it for downstream online RL and assessing the average expected return or success rate.

To show the benefits of combining contrastive and reconstruction-based objectives, we compare with ablative variants that use the same loss for both modalities (*Same-Loss*), the naive approach of concatenating proprioception to image representations (*Concat*) and using only the image (*Img-Only*). We consider the contrastive variational (CV) and the contrastive predictive coding (CPC) paradigm for each of these approaches. For reference, we also include reconstruction-based (Recon.) approaches (Equation 1). Furthermore, we use SAC (Haarnoja et al., 2018) on only the proprioception (*ProprioSAC*), to show that proprioception alone is insufficient to solve the tasks. Finally, we consider the model-free *DrQ-v2* (Yarats et al., 2022) and model-based *RePo* (Zhu et al., 2023) as baselines to demonstrate the competitiveness of our approach. We extend both to also use proprioception and refer to the resulting approaches as *DrQ-v2(I+P)* and *RePo(I+P)* respectively.

Evaluation Protocol. We run 5 seeds for each task in each suite and build our analysis on the aggregated results across the entire suite. This process results in 35 runs for each method on *Video*

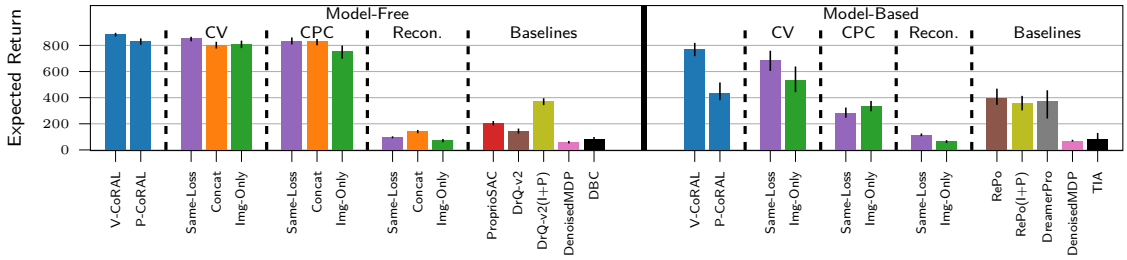

Figure 2: Aggregated performance after 10^6 environment steps on the 7 tasks from the *Video Background* suite (IQM and 95% CIs). For both model-free and model-based RL, *V-CoRAL* performs best among all considered methods, with the model-free performance being better than the model-based one. While some of the model-free ablations are competitive, they perform considerably worse in the model-based case. From the baselines, only *DrQ-v2* with additional proprioception, *RePo* (with and without proprioception), and *DreamerPro* get a final return of over 200. These results demonstrate how including readily available proprioception with appropriate losses for each modality helps to learn accurate dynamics required by model-based RL and provides a simple alternative to more tailored approaches.

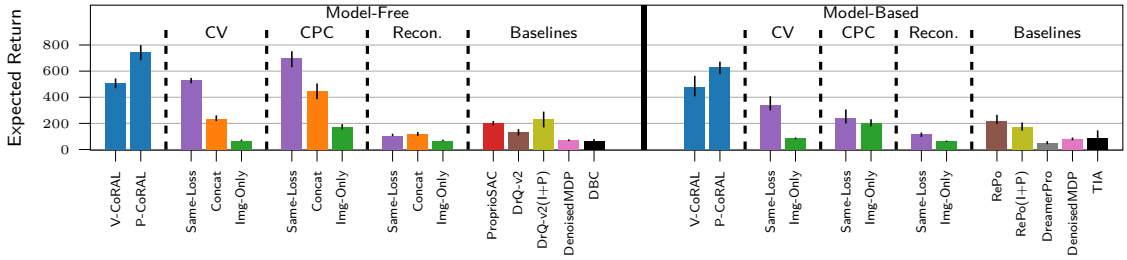

Figure 3: Aggregated performance after 10^6 environment steps on the 7 tasks from the *Occlusion* suite (IQM and 95% CIs). For both, model-free and model-based RL, *P-CoRAL* performs best among all considered methods, with the model-free version again outperforming its model-based counterpart. While all approaches handle *Occlusions* worse than *VideoBackgorund*, the performance drop is generally larger for the ablations and baselines. In particular, the *Concat* and model-based *Same-Loss* ablations suffer and no approach using only a single modality achieves an expected return of over 200. This indicates the importance of learning a multimodal representation using tailored losses over naively integrating proprioception.

Background and *Occlusions* and 30 runs for each method in the *Locomotion* and *Manipulation* suites. For aggregating the results over a suite, we follow Agarwal et al. (2021) and provide Interquartile Means (IQMs), which they found to be more meaningful and robust than alternatives such as mean or median in related scenarios. Similarly, we follow Agarwal et al. (2021) and provide 95% Stratified Bootstrapped Confidence Intervals (CIs) for the entire suite to quantify the statistical uncertainty in results. We indicate those with black bars in bar charts or shaded areas in reward curve plots.

Appendix A provides details for all tasks. Appendix B lists all hyperparameters of our approach and Appendix C provides further details on the baselines. Appendix D shows learning curves for all representation learning paradigms on all tasks, performance profiles, and per-environment results. Code for running *CoRAL* and ablations on all tasks is available[1].

4.1 Modified Deep Mind Control Suite Tasks

We use 7 tasks from the DeepMind Control Suite (DMC) (Tassa et al., 2018) that cover a wide range of challenges, namely `Ball-in-Cup Catch`, `Cartpole Swingup`, `Cheetah Run`, `Reacher Easy`, `Walker Walk`, `Walker Run`, and `Quadruped Walk`. We split their states into proprioceptive and non-proprioceptive entries, where the proprioception only contains partial information about the state.

[1] https://github.com/pbecker93/CoRAL/

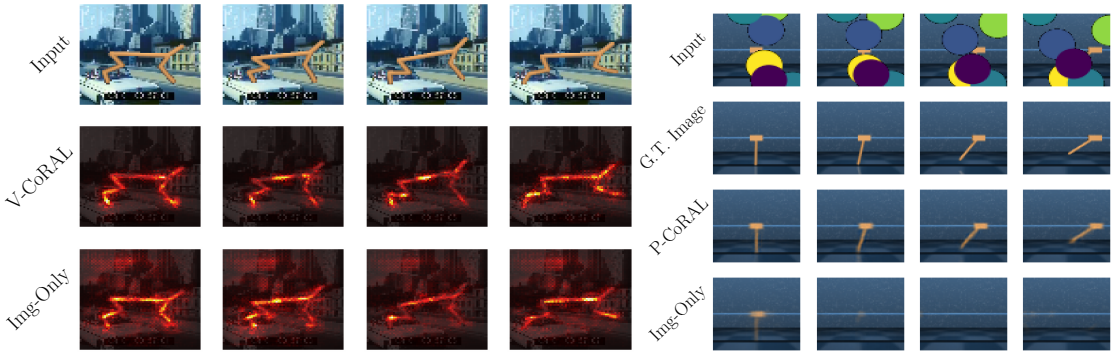

Figure 4: **Left:** Saliency Maps showing on which pixels the respective representation learning approaches focus in an example from *Video Prediction*. *V-CoRAL* focuses better on the task-relevant cheetah, while the corresponding contrastive variational *Img-Only* approach is more distracted by the video background. **Right:** For this *Occlusion* task, we train a separate decoder to reconstruct the occlusion-free ground truth from the (detached) latent representation. For `Cartpole Swingup` only the cart position is part of the proprioception. Still, *P-CoRAL* can capture both cart position and pole angle, while the contrastive predictive *Img-Only* approach fails to do so.

The remaining information has to be inferred from images. For example, in `Ball-in-Cup Catch` the cup's state is proprioceptive while the ball's state is not. Table 1 lists the splits for the remaining tasks. We create two suites by adding *Video Backgrounds* or *Occlusions* for all seven tasks. For *Video Backgrounds*, we follow (Nguyen et al., 2021; Deng et al., 2022) and render videos from the Kinetics400 dataset (Kay et al., 2017) behind the agent. For *Occlusions*, we add slowly moving disks in front of the agent. The upper row of Fig. 4 shows examples. For both suites, the challenge is to learn representations that filter out irrelevant visual details while focusing on relevant aspects. *Occlusions* also tests the approaches' capabilities to maintain a consistent representation across time under partial observability, considerably increasing the task's difficulty.

For these tasks, we consider the model-free and model-based versions of *V-CoRAL*, *P-CoRAL*, and all ablative variants. Note that the *Concat* ablations are inapplicable in the model-based setting, as the proprioception is not available during *latent imagination* (Hafner et al., 2020). Besides the *DrQ-v2* and *RePo* based baselines, we include several other visual RL approaches tailored for images with distractions to show the competitiveness of *CoRAL*. Those are the model-based *Task Informed Abstractions (TIA)* (Fu et al., 2021) and *DreamerPro* (Deng et al., 2022), the model-free *Deep Bisimulation for Control (DBC)* (Zhang et al., 2020) approach, and *DenoisedMDP* (Wang et al., 2022), which has both a model-free and model-based variant.

Fig. 2 and Fig. 3 show the results for the *Video Background* and *Occlusion* tasks respectively. We also include results for the *Standard Images* without any distractors or occlusions for reference and refer to Appendix D for those results. On *Natural Videos* *V-CoRAL* yields the best results among all approaches. However, the margin to some of the ablations is small with all of them closing in on the performance of the best approaches on images without background videos (Fig. 9). For model-based RL the results show clearer benefits of learning a multimodal representation by appropriately combining multiple losses. This difference is also much more pronounced for the more difficult *Occlusions* (Fig. 3) suite. Here, no image-only approach learns reasonable behavior or manages to outperform *ProprioSAC*, indicating a higher difficulty for representation learning. Our method *P-CoRAL* tends to perform best in this suite, achieving a return of around 750, and closing in on the best approaches on *Standard Images* which get around 900. Furthermore, using readily available proprioception for representation learning in a principled manner provides a simple alternative to the strong baselines listed above and also tends to outperform the naive *Concat* ablation that does not consider proprioception for representation learning but only for RL.

Variational vs. Predictive Approaches. Variational approaches tend to work better than predictive ones on *Video Backgrounds*, where the challenge is to focus on the relevant aspects while ignoring distractions. Yet, the predictive approaches work better on *Occlusions*, where information

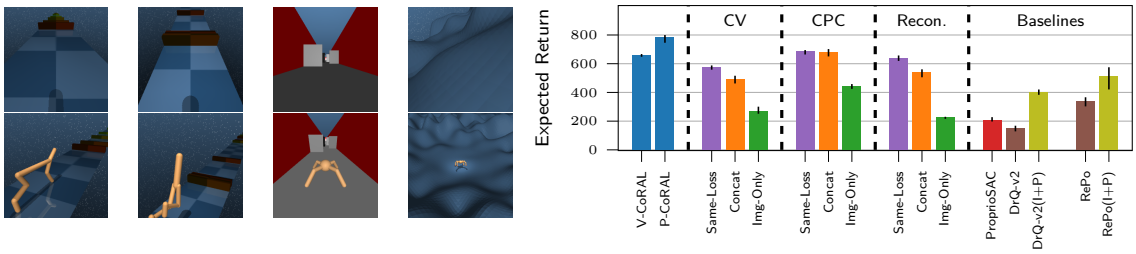

Figure 5: **Left**: Exemplary egocentric (upper row) and external example images (lower row) for the `Hurdle Cheetah Run`, `Hurdle Walker Run`, `Ants Walls`, and `Quadruped Escape` tasks of the *Locomotion* suite. Only the egocentric images are given to the agents, while the external images are solely for visualization of the tasks. **Right:** Aggregated performance on model-free agents and *RePo* after 10^6 environment steps on the 6 tasks of the *Locomotion* suite (IQM and 95% CIs). *P-CoRAL* significantly outperforms all ablative variants and baselines, highlighting how combining contrastive methods and reconstruction can form effective multimodal representations. It also outperforms purely reconstruction-based approaches, even with no distraction in the images.

has to be propagated over time. As the underlying tasks are identical, this highlights the benefits of considering both paradigms, depending on the perception challenges.

Visualization of Learned Representations. We qualitatively investigate some of the learned representations in Fig. 4, which illustrates how *CoRAL* helps the representation to focus on relevant aspects and extract all necessary information from an image.

Model Quality and Model-Based Approaches. While model-free and model-based agents perform similarly well for approaches that reconstruct images, model-based agents perform worse than their model-free counterparts for contrastive image losses (Fig. 2, Fig. 3, Fig. 9, Fig. 10). In line with previous findings (Hafner et al., 2020; Ma et al., 2020), this shows how contrastive approaches struggle to learn suitable long-term dynamics for model-based RL. However, this gap is larger for the *Same-Loss* and *Img-Only* ablations than for *CoRAL*, which almost closes the gap between model-free and model-based for *V-CoRAL* (Fig. 2, Fig. 3). This result demonstrates how *CoRAL* allows learning more precise long-term dynamics that enable more successful model-based RL.

4.2 *Locomotion* Suite

Building on the DeepMind Control Suite Tassa et al. (2020), we introduce a novel *Locomotion* suite consisting of six tasks: `Hurdle-Cheetah Run`, `Hurdle-Walker Walk`, `Hurdle-Walker Run`, `Ant-Empty`, `Ant-Walls` and `Quadruped Escape`. All tasks include obstacles that have to be localized through egocentric vision to be avoided. As the agents cannot observe themselves from the egocentric perspective, they additionally need proprioception. The left side of Fig. 5 provides some examples and Appendix A.2 provides further illustrations and specifications of all tasks. These tasks test the representations' ability to combine information from both sources to enable successful navigation and movement. For this more challenging suite, we focus on model-free RL for all representations due to the known performance gap for model-based RL with contrastive image losses (Hafner et al., 2020; Ma et al., 2020), (Fig. 2, Fig. 3). We include the model-based *RePo* for reference.

The results on the right side of Fig. 5 show that *P-CoRAL* excels in the *Locomotion* suite and has a significant edge over reconstruction or the pure CPC-based approach while *V-CoRAL* outperforms the related variational approaches. While highly relevant to the task, the obstacles appear at random and have random colors for some tasks, which makes reconstruction harder. The contrastive methods' advantage is pronounced in tasks with random colored obstacles (Fig. 21).

4.3 *Manipulation* Suite

For the *Manipulation* suite, we design 6 tasks based on ManiSkill2 (Gu et al., 2023), i.e., `LiftCube`, `PushCube`, `TurnFaucet`, `OpenCabinetDrawer(RGB)`, `OpenCanbinetDrawer(Depth)`, and

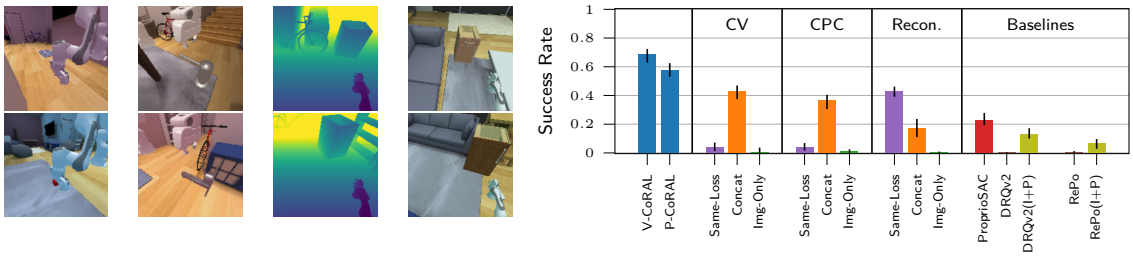

Figure 6: **Left**: Exemplary images of the `LiftCube`, `TurnFaucet`, `OpenCabinetDrawer(Depth)`, and `OpenCabinetDoor(RGBD)` tasks. For the last one, we only show the RGB part of the image and we provide two images per task to showcase the visual diversity and different geometries of the target objects. **Right:** Aggregated performance on model-free agents and *RePo* after 2×10^6 environment steps on the *Manipulation* suite (IQM and 95% CIs). Overall, *V-CoRAL* achieves the best average success rate by a significant margin, followed by *P-CoRAL*. While they achieve about 68% and 58% success respectively, no ablation gets over 42%. In particular, both fully contrastive *Same-Loss* ablations fail to succeed, which again highlights the importance of choosing an appropriate loss for each modality. While both *RePo* and *DrQ-v2* can utilize the additional proprioception, they are not competitive with *CoRAL* or even SAC trained solely on the proprioception.

`OpenCabinetDoor(RGBD)`. The first three are static manipulation tasks where the target object has to be localized (cube) or identified (faucet) for successful manipulation. The latter three are mobile manipulation tasks where the robot navigates to a cabinet and interacts with it using egocentric vision and proprioception. They also use different visual modalities, i.e., standard RGB images, depth only, or RGBD. For all tasks, we add visually realistic backgrounds using diverse scenes from the ReplicaCAD Dataset (Straub et al., 2019) and randomize the ambient lighting. The task's complexity stems from the visual realism of the background and the diverse geometry of the target objects, which require that the representations allow identification and precise localization. The left side of Fig. 6 provides example images showing the tasks' visual diversity and Appendix A.3 further examples and specifications for all tasks. We again focus on model-free RL and *RePo*.

The right sight of Fig. 6 shows the results. The *Manipulation* suite is the hardest set of tasks we consider and here the benefits of *CoRAL* are most obvious. Here, most of the considered baselines fail while only *V-CoRAL* and *P-CoRAL* achieve over 50% success rate, averaged over all tasks, with *V-CoRAL* giving the best result of 68%. In particular, the corresponding contrastive *same-loss* approaches fail almost completely, which puts additional emphasis on the importance of using appropriate losses for each modality. Using different image types for the 3 mobile manipulation tasks shows how *CoRAL* is beneficial across different visual modalities. Using depth images, `OpenCabinetDrawer(Depth)` effectively removes the lighting variations for this task which allows several approaches to achieve higher performance but has only minor effects on the ranking.

4.4 Discussion

Considering all task suites and the full results presented in Appendix D, we see the benefits of *CoRAL* compared to the ablations and a large selection of model-free and model-based baselines. Especially for the harder tasks, i.e., *Occlusions* (Fig. 3), *Locomotion* (Fig. 5), and *Manipulation* (Fig. 6), *CoRAL* can significantly outperform other methods working on the same observations, which shows that different modalities require distinct self-supervised loss functions while simply using the additional proprioception by concatenation or using the same self-supervised loss is often insufficient.

Variational vs. Predictive Approaches. While either *V-CoRAL* or *P-CoRAL* generally provides the best results on the considered tasks and both outperform the corresponding ablations, neither consistently outperforms the other across all task suites. While this prevents conclusive decisions about whether variational or predictive methods work generally better, we can observe a trend. The variational approaches appear more suited for tasks requiring the filtering out of visual distractions,

such as in *Video Background* and *Manipulation* scenarios, while predictive approaches perform better in tasks needing information to be carried over time, like *Occlusions* and *Locomotion*.

Baselines. Contrary to the results presented in the respective original works, *DBC* (Zhang et al., 2020), *TIA* (Fu et al., 2021), *DenoisedMDP* (Wang et al., 2022) and *RePo* (Zhu et al., 2023) underperform on the *Video Backgrounds* task. The discrepancy in performance is due to us using a more difficult experimental setup proposed by Deng et al. (2022), which features colored videos of greater diversity. We detail the differences and their effects in Appendix C. Furthermore, *RePo* fails in the *Manipulation* suite which seems to contradict results presented by Zhu et al. (2023) on three static manipulation tasks, similar to those in that suite. Again there are subtle differences in the task specification: While Zhu et al. (2023) only randomize the visual background we randomize both the visual background and the task's initial condition (cube position or faucet model) creating considerably more challenging scenarios.

Consistency Across Tasks. The additional result visualizations in Appendix D show that the aggregated performance underlying our analysis is mostly representative of the per-task performance, i.e., if an approach outperforms another when considering the aggregated performance, it generally also does so on a large majority of the individual tasks and runs. Furthermore, performance is consistent across the different observation types for the *DMC* tasks, i.e., *Occlusions* are more difficult than *Video Background*, which are more difficult than *Standard Images* (Fig. 9, Fig. 10).

5 Conclusion

We consider the problem of Reinforcement Learning (RL) from multiple sensors, in particular images and proprioception. We propose *Contrastive reconstructive Aggregated Representation Learning (CoRAL)*, an approach to learning multimodal state space representations for RL by combining contrastive and reconstruction losses. *CoRAL* builds on the insight that we can replace likelihood-based reconstruction terms with contrastive mutual information terms and vice-versa and is applicable for variational and predictive coding paradigms. We evaluate on modified versions of the DeepMind Control Suite and novel *Locomotion* and *Manipulation* suites. Our results show a consistent benefit of *CoRAL* due to the combination of contrastive approaches for images with reconstruction for low-dimensional, concise signals. These benefits are most pronounced for the hardest tasks we consider, i.e., the *Manipulation* suite, where *CoRAL*, allows us to solve complex tasks with realistic background scenes and varying target object geometries.

Limitations. Depending on the task, either *V-CoRAL* or *P-CoRAL* performs better. While our evaluation provides some insights about when to use either, further research into understanding their advantages and disadvantages and finding a unified approach that excels in all tasks is required. Additionally, even with *CoRAL*, model-free agents outperform their model-based counterparts when using contrastive image losses. We thus believe that contrastive learning of state space representations can be further improved, especially with regard to learning accurate system dynamics.

Acknowledgments

The authors acknowledge support by the state of Baden-Württemberg through bwHPC, as well as the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the German Federal Ministry of Education and Research.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2016.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759. PMLR, 2018.

Philipp Becker and Gerhard Neumann. On uncertainty in deep state space models for model-based reinforcement learning. *Transactions on Machine Learning Research*, 2022.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Fei Deng, Ingook Jang, and Sungjin Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pp. 4956–4975. PMLR, 2022.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*. PMLR, 2023.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pp. 3480–3491. PMLR, 2021.

Zipeng Fu, Ashish Kumar, Ananye Agarwal, Haozhi Qi, Jitendra Malik, and Deepak Pathak. Coupling vision and proprioception for navigation of legged robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17273–17283, June 2022.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. In *Advances in Neural Information Processing Systems*, 2022.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

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 *Conference on Robot Learning*, pp. 651–673. PMLR, 2018.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/laskin20a.html.

Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020.

Nir Levine, Yinlam Chow, Rui Shu, Ang Li, Mohammad Ghavamzadeh, and Hung Bui. Prediction, consistency, curvature: Representation learning for locally-linear control. In *International Conference on Learning Representations*, 2019.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Xiao Ma, Siwei Chen, David Hsu, and Wee Sun Lee. Contrastive variational reinforcement learning for complex observations. In *Conference on Robot Learning*, pp. 959–972. PMLR, 2020.

Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Tung D Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal predictive coding for model-based planning in latent space. In *International Conference on Machine Learning*, pp. 8130–8139. PMLR, 2021.

Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4209–4215. IEEE, 2021.

Masashi Okada and Tadahiro Taniguchi. Dreamingv2: Reinforcement learning with discrete world models without reconstruction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 985–991. IEEE, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *6th Annual Conference on Robot Learning*, 2022.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. *PMLR*, 2023.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023.

Rui Shu, Tung Nguyen, Yinlam Chow, Tuan Pham, Khoat Than, Mohammad Ghavamzadeh, Stefano Ermon, and Hung Bui. Predictive coding for locally-linear control. In *International Conference on Machine Learning*, pp. 8862–8871. PMLR, 2020.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

Nitish Srivastava, Walter Talbott, Martin Bertran Lopez, Shuangfei Zhai, and Joshua M. Susskind. Robust robotic control from pixels using contrastive recurrent state-space models. In *Deep RL Workshop NeurIPS 2021*, 2021.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pp. 9870–9879. PMLR, 2021.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

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 preprint arXiv:1801.00690*, 2018.

Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

Manan Tomar, Utkarsh Aashu Mishra, Amy Zhang, and Matthew E Taylor. Learning representations for pixel-based control: What matters and why? *Transactions on Machine Learning Research*, 2023.

Niklas Wahlström, Thomas B Schön, and Marc Peter Deisenroth. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.

Tongzhou Wang, Simon Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. In *International Conference on Machine Learning*, pp. 22591–22612. PMLR, 2022.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *6th Annual Conference on Robot Learning*, 2022.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pp. 11920–11931. PMLR, 2021a.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10674–10681, 2021b.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022.

Bang You, Oleg Arenz, Youping Chen, and Jan Peters. Integrating contrastive learning with dynamic models for reinforcement learning from images. *Neurocomputing*, 476:102–114, 2022.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2020.

Chuning Zhu, Max Simchowitz, Siri Gadipudi, and Abhishek Gupta. Repo: Resilient model-based reinforcement learning by regularizing posterior predictability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Table 1: Splits of the entire system state into proprioceptive and non-proprioceptive parts for the DeepMind Control Suite environments.

Environment	Proprioceptive	Non-Proprioceptive
Ball In Cup	cup position and velocity	ball position and velocity
Cartpole	cart position and velocity	pole angle and velocity
Cheetah	joint positions and velocities	global pose and velocity
Reacher	reacher position and velocity	distance to target
Quadruped	joint positions and velocities	global pose + velocity, forces
Walker	orientations and velocities of links	global pose and velocity, height above ground

Table 2: Splits of the entire system state into proprioceptive and non-proprioceptive parts for the Locomotion Suite. Some of the agents (Cheetah, Walker, Quadruped) require more proprioceptive information for the locomotion tasks with an egocentric vision than for the standard tasks with images from an external perspective.

Environment	Proprioceptive	Non-Proprioceptive
Ant	joint position and velocity global velocities	wall positions global position
Hurdle Cheetah	joint positions and velocities global velocity	hurdle positions and heights global position
Hurdle Walker	orientations and velocities of links	hurdle positions and height global position and velocity
Quadruped (Escape)	joint positions and velocities, torso orientation and velocity, imu, forces, and torques at joints	Information about terrain

A Environments

A.1 DeepMind Control Suite Tasks

Table 1 states how we split the states of the original DeepMind Control Suite (DMC) (Tassa et al., 2018) tasks into proprioceptive and non-proprioceptive parts. For the model-based agents, we followed common practice (Hafner et al., 2020; Fu et al., 2021; Wang et al., 2022; Deng et al., 2022) and use an action repeat of 2 for all environments. We do the same for the model-free agents except for: `Ball In Cup Catch` (4), `Cartpole Swingup` (8), `Cheetah Run` (4) and `Reacher Easy` (4). All environments in the locomotion suite also use an action repeat of 2, this includes `Hurdle Cheetah Run` which requires more fine-grained control than the normal version to avoid the hurdles.

Natural Background. Following (Zhang et al., 2020; Fu et al., 2021; Nguyen et al., 2021; Deng et al., 2022; Wang et al., 2022; Zhu et al., 2023) we render videos from the `driving car` class of the Kinetics400 dataset (Kay et al., 2017) behind the agents to add a natural video background. However, previous works implement this idea in two distinct ways. Nguyen et al. (2021) and Deng et al. (2022) use color images as background and pick a random sub-sequence of a random video for each environment rollout. They adhere to the train-validation split of the Kinetcs400 dataset, using training videos for representation and policy learning and validation videos during evaluation. Zhang et al. (2020); Fu et al. (2021); Wang et al. (2022); Zhu et al. (2023), according to the official implementations, instead work with gray-scale images and sample a single background video for the train set once during initialization of the environment. They do not sample a new video during the environment reset, thus all training sequences have the same background video.

We follow the first approach, as we believe it mimics a more realistic scenario of always changing and colored natural background.

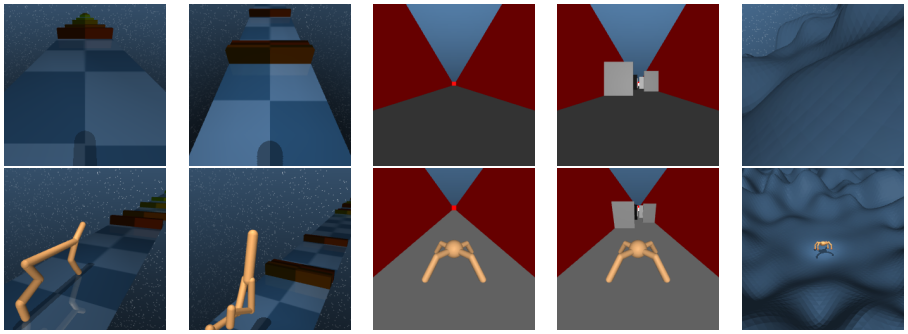

Figure 7: The environments in the Locomotion Suite are (from left to right) `Hurdle Cheetah Run`, `Hurdle Walker Walk / Run`, `Ant Empty`, `Ant Walls`, and `Quadruped Escape`. **Upper Row:** Egocentric vision provided to the agent. **Lower Row:** External image for visualization.

Occlusions. Following (Becker & Neumann, 2022), we render slow-moving disks over the original observations to occlude parts of the observation. The speed of the disks makes memory necessary, as they can occlude relevant aspects for multiple consecutive timesteps.

A.2 Locomotion Suite

The 6 tasks in the locomotion suite are `Ant Empty`, `Ant Walls`, `Hurdle Cheetah Run`, `Hurdle Walker Walk`, `Hurdle Walker Run`, and `Quadruped Escape`. Table 2 shows the splits into proprioceptive and non-proprioceptive parts. Fig. 7 displays all environments in the suite.

Both **Ant** tasks build on the locomotion functionality introduced into the DeepMind Control suite by (Tassa et al., 2020). For `Ant Empty`, we only use an empty corridor, which makes this the easiest task in our locomotion suite. For `Ant Walls`, we randomly generate walls inside the corridor, and the agent has to avoid those to achieve its goal, i.e., running through the corridor as fast as possible.

For the **Hurdle Cheetah** and **Hurdle Walker** tasks we modified the standard `Cheetah Run`, `Walker Walk`, and `Walker Run` tasks by introducing "hurdles" over which the agent has to step to move forward. The hurdles' positions, heights, and colors are reset randomly for each episode, and the agent has to perceive them using egocentric vision. For this vision, we added a camera in the head of the Cheetah and Walker. Note that the hurdle color is not relevant to avoid them and thus introduces irrelevant information that needs to be captured by reconstruction-based approaches.

The **Quadruped Escape** task is readily available in the DeepMind Control Suite. For the egocentric vision, we removed the range-finding sensors from the original observation and added an egocentric camera.

A.3 Manipulation Suite

The Manipulation Suite builds on Maniskill2 (Gu et al., 2023) and comprises 6 tasks, i.e., `LiftCube`, `PushCube`, `TurnFaucet`, `OpenCabinetDrawer(RGB)`, `OpenCabinetDrawer(Depth)` and `OpenCabinetDoor(RGBD)`. The first three involve table-top manipulation and are harder variations of some tasks considered by Zhu et al. (2023). The latter three are mobile manipulation tasks using different image modalities. For all tasks, we use scenes from the Replica Dataset Straub et al. (2019) (specifically: ReplicaCAD_baked_lighting[2]) to place the robot in a visually realistic scene. At the beginning of each episode, we randomly pick one of 80 curated scenes and randomly sample the ambient lighting to place the task in a varying and visually realistic scenery.

[2] https://huggingface.co/datasets/ai-habitat/ReplicaCAD_baked_lighting/

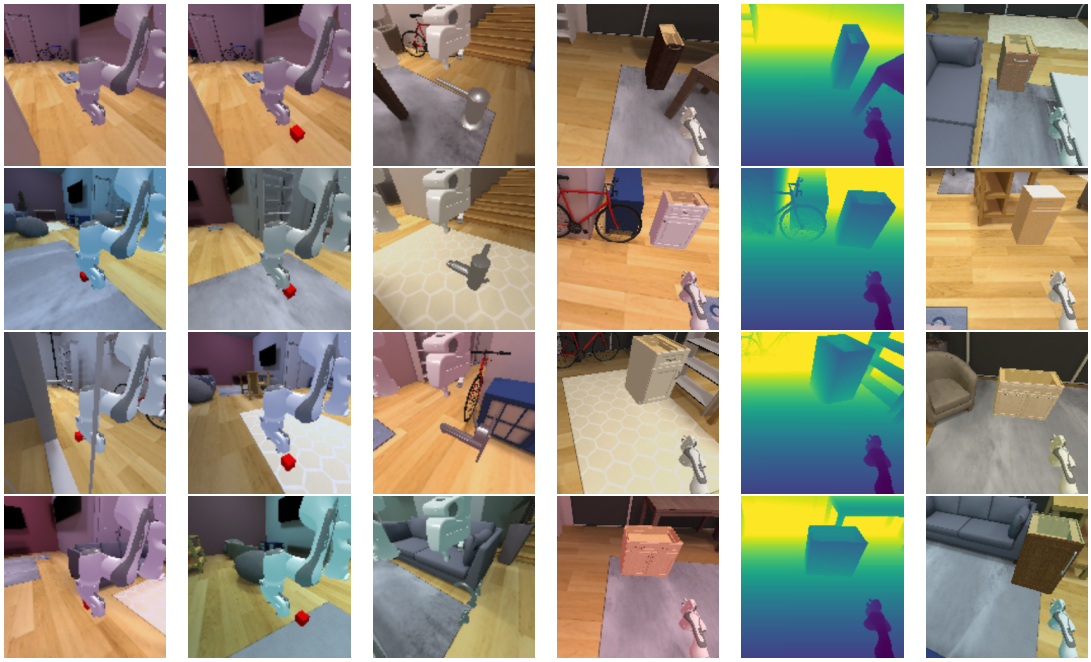

Figure 8: 4 Example images for each of the environments in the *Manipulation* Suite, showing the visual and geometric diversity within each task. The tasks are, from left to right, `LiftCube`, `PushCube`, `TurnFaucet`, `OpenCabinetDrawer(RGB)`, `OpenCabinetDrawer(Depth)`, `OpenCabinetDoor(RGBD)`. For the last, we visualize only the RGB part of the image.

We use delta joint position control, no action repeat, and dense normalized rewards for all tasks. For the depth images we use the depth camera functionality provided by ManiSkill2 and clip to values between 0 and 4 meters. Figure Fig. 8 shows example images for all environments.

LiftCube builds on Maniskill2's LiftCube task and involves picking up a cube and lifting it to a fixed target position. The proprioception includes the robot's joint positions, velocities, and end-effector pose, while the cube has to be localized and tracked via an image of an external camera. Opposed to Zhu et al. (2023) we randomize the initial cube position, requiring the agents to first localize the cube based on the representation, which makes the task considerably more difficult.

PushCube builds on the PushCube task introduced by Zhu et al. (2023), but we again randomize the initial cube position. Like in LiftCube, the proprioception includes the robot's joint positions, velocities, and end-effector pose, while the cube has to be localized and tracked via an image of an external camera.

TurnFaucet extends Maniskill2's TurnFaucet task and involves opening various faucets by turning the handle. The proprioception includes the robot's joint positions, velocities, and end-effector pose, while all information regarding the faucet has to be inferred from an image of an external camera. We sample one out of 15 different faucets at the beginning of each episode. As their geometry and opening mechanism vary any representation needs to capture detailed information about the faucet and allow the policy to identify it. This makes our task considerably more difficult than that proposed by Zhu et al. (2023), who use the same faucet model for all episodes.

OpenCabinetDrawer(RGB) is based on the mobile manipulation OpenCabinetDrawer task from ManiSkill2, where a mobile robot with a single arm has to navigate towards and then open a drawer one of 25 cabinets. We disable the rotation of the robot base, which prevents the robot from turning away from the cabinet during initial exploration and significantly speeds up learning for all considered approaches. This results in a 10 dimensional action space, consisting of the x and y velocities of the

base, desired changes for the 7 robot joints, and the gripper. Images are egocentric from the top of the robot base and the proprioception includes the entries from the ManiSkill2 "state dict".

OpenCabinetDrawer(Depth) is equivalent to OpenCabientDrawer(RGB) but the agent only receives an egocentric depth image instead of a color image. This effectively removes the variation in lighting from the environment.

For **OpenCabinetDoor(RGBD)** we build on the Maniskill2 task of the same name, use 25 different cabinet models, and the same action space as for OpenCabientDrawer(RGB). The sensory observations are also equivalent to the Drawer tasks, but we provide both color and depth information. While conceptually similar to the Drawer tasks opening the Door is considerably harder, as it requires coordination with the base not just to reach the handle, but also to pull back on it.

B Architecture Details and Training

We use the same hyperparameters for all experiments based on the DeepMind Control Suite (DMC), i.e., the standard tasks with the different observation types (*Video Background*, *Occlusions* and also *Standard Images*) as well as, the Locomotion Suite. For the ManiSkill2-based Manipulation Suite, we use a larger model and a more conservative update scheme for actors and critics. We use the ELU activation function unless otherwise mentioned.

B.1 *Recurrent State Space Model*

We denote the deterministic part of the *RSSM*'s state by $\mathbf{h}_t$ and the stochastic part by $\mathbf{s}_t$. The base-*RSSM* model without parts specific to the objective consists of:

- **Encoders:** $\psi_{\text{obs}}^{(k)}(\mathbf{o}_t)$, where ψ_{obs} is the convolutional architecture proposed by (Ha & Schmidhuber, 2018) and used by (Hafner et al., 2019; 2020) for image observations. For the low-dimensional proprioception, we used 3×400 fully connected layers for the DMC tasks and 4×512 fully connected layers Manipulation Suite.
- **Deterministic Path**: $\mathbf{h}_t = g(\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{h}_{t-1}) = \text{GRU}(\psi_{\text{det}}(\mathbf{z}_{t-1}, \mathbf{a}_{t-1}), \mathbf{h}_{t-1})$ (Cho et al., 2014), where ψ_{det} is a 2×400 fully connected NN and the GRU has a memory size of 200 for the DMC tasks. For the Manipulation Suite ψ_{det} has 2×512 units and the GRU a memory size of 400
- **Dynamics Model**: $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{a}_t) = \psi_{\text{dyn}}(\mathbf{h}_t)$, where ψ_{dyn} is a 2×400 units fully connected NN for the DMC tasks and a 2×512 units fully connected NN for the Manipulation Suite. The network learns the mean and standard deviation of the distribution.
- **Variational Distribution** $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t) = \psi_{\text{var}}\left(\mathbf{h}_t, \text{Concat}\left(\{\psi_{\text{obs}}^{(k)}(\mathbf{o}_t^{(k)})\}_{k=1:K}\right)\right)$, where ψ_{var} is a 2×400 units fully connected NN for the DMC tasks and a 2×512 units fully connected NN for the Manipulation Suite. Again, the network learns the mean and standard deviation of the distribution.
- **Reward Predictor** $p(r_t|\mathbf{z}_t)$: 2×128 units fully connected NN for model-free agents. 3×300 units fully connected NN with ELU activation for model-based agents. The network only learns the mean of the distribution. The standard deviation is fixed at 1. The model-based agents use a larger reward predictor as they rely on it for learning the policy and the value function. Model-free agents use the reward predictor only for representation learning and work with the ground truth rewards from the replay buffer to learn the critic.

B.2 Objectives

Image Inputs and Augmentation. Whenever we use a contrastive image loss, we randomly crop a 64×64 pixel image from the original image of size 76×76 pixels during training. Cropping is

temporally consistent, i.e., the same crop is used for all time steps in a sub-sequence. For evaluation, we corp at the center. For the ablations that reconstruct images, we downsize them directly to 64×64 pixels.

KL. For the KL terms in Equation 1 and Equation 3 we follow Hafner et al. (2023) and combine the KL-Balancing technique introduced in Hafner et al. (2021) with the *free-nats regularization* used in Hafner et al. (2019; 2020). Following Hafner et al. (2021) we use a balancing factor of 0.8. We give the algorithm 1 free nat for the DMC Tasks and 3 for the Manipulation Suite.

Contrastive Variational Objective. The score function for the contrastive variational objective is given as

$$f_v^{(k)}(\mathbf{o}_t^{(k)}, \mathbf{z}_t) = \exp\left(\frac{1}{\lambda}\rho_o\left(\psi_{\text{obs}}^{(k)}(\mathbf{o}_t^{(k)})\right)^T \rho_z(\mathbf{z}_t)\right),$$

where $\psi_{\text{obs}}^{(k)}$ is the *RSSM*'s encoder and λ is a learnable inverse temperature parameter. ρ_o and ρ_z are projections that project the embedded observation and latent state to the same dimension, i.e., 50. ρ_o is only a single linear layer while ρ_z is a 2×256 fully connected NN. Both use LayerNorm (Ba et al., 2016) at the output.

Contrastive Predictive Objective. The score function of the contrastive predictive objective looks similar to the one of the contrastive variational objective. The only difference is that the latent state is forwarded in time using the *RSSMs* transition model to account for the predictive nature of the objective,

$$f_p^{(k)}(\mathbf{o}_t^{(k)}, \mathbf{z}_{t-1}) = \exp\left(\frac{1}{\lambda}\rho_o\left(\psi_{\text{obs}}^{(k)}(\mathbf{o}_t^{(k)})\right)^T \rho_z(\phi_{\text{dyn}}(g(\mathbf{z}_{t-1}, \cdot))\right).$$

We use the same projections as in the contrastive variational case.

Following Srivastava et al. (2021) we scale the KL term using a factor of $\beta = 0.001$.

Reconstruction Objectives. Whenever we reconstruct images we use the up-convolutional architecture proposed by (Ha & Schmidhuber, 2018) and used by (Hafner et al., 2019; 2020). For low-dimensional observations, we use 3×400 units fully connected NN for the DMC tasks and a 4×512 Units fully connected NN for the Manipulation Suite. In all cases, only the mean is learned. We use a fixed variance of 1 for all image losses and the proprioception for the DMC tasks. For the Manipulation Suite, we set the variance for the proprioception to 0.04.

Optimizer. We used Adam Kingma & Ba (2015) with $\alpha = 3 \times 10^{-4}$, $\beta_1 = 0.99, \beta_2 = 0.9$ and $\varepsilon = 10^{-8}$ for all losses. We clip gradients if the norm exceeds 10.

B.3 Soft Actor Critic

Table 3 lists the hyperparameters used for model-free RL with SAC Haarnoja et al. (2018).

We collected $5,000$ initial steps at random. During training, we update the *RSSM*, critic, and actor in an alternating fashion for d steps before collecting a new sequence by directly sampling from the maximum entropy policy. Here, d is set to be half of the environment steps collected per sequence (after accounting for potential action repeats). Each step uses 32 subsequences of length 32, uniformly sampled from all prior experience.

B.4 Latent Imagination

Table 4 lists the hyperparameters used for model-based RL with latent imagination. They follow to a large extent those used in Hafner et al. (2020; 2021).

We again collect $5,000$ initial steps at random. During training, we update the *RSSM*, value function, and actor in an alternating fashion for 100 steps before collecting new sequences. Each step uses 50 subsequences of length 50, uniformly sampled from all prior experience. For collecting new data, we use constant Gaussian exploration noise with $\sigma = 0.3$.

Table 3: Hyperparameters used for policy learning with the Soft Actor-Critic.

Hyperparameter	DMC and Locomotion	Manipulation
Actor Hidden Layers	$3 \times 1,024$ Units	$3 \times 1,024$ Units
Actor Activation	ELU	ELU + LayerNorm
Critic Hidden Layers	$3 \times 1,024$ Units	$3 \times 1,024$ Units
Critic Activation	Tanh	ELU + LayerNorm
Discount	0.99	0.85
Actor Learning Rate	0.001	0.0003
Actor Gradient Clip Norm	10	10
Critic Learning Rate	0.001	0.0003
Critic Gradient Clip Norm	100	100
Target Critic Decay	0.995	0.995
Target Critic Update Interval	1	1
α learning rate	0.001	0.0003
initial α	0.1	1.0
target entropy	- action dim	- action dim

Table 4: Hyperparameters used for policy learning with *Latent Imagination.*

Hyperparameter	Value
Actor Hidden Layers	3×300 Units
Actor Activation	ELU
Critic Hidden Layers	3×300 Units
Critic Activation	ELU
Discount	0.99
Actor Learning Rate	8×10^{-5}
Actor Gradient Clip Norm	100
Value Function Learning Rate	8×10^{-5}
Value Gradient Clip Norm	100
Slow Value Decay	0.98
Slow Value Update Interval	1
Slow Value Regularizer	1
Imagination Horizon	15
Return lambda	0.95

C Details on Baselines and Ablations.

For *Dreamer-v3* (Hafner et al., 2023) we use the raw reward curve data provided with the official implementation[3]. For *DreamerPro* (Deng et al., 2022)[4], *Task Informed Abstractions* (Fu et al., 2021)[5], *Deep Bisumlation for Control* (Zhang et al., 2020)[6], *DenoisedMDP* (Wang et al., 2022)[7] and *DrQ-v2* (Yarats et al., 2022)[8] we use the official implementations provided by the respective authors.

[3] https://github.com/danijar/dreamerv3/blob/main/scores/data/dmcvision_dreamerv3.json.gz
[4] https://github.com/fdeng18/dreamer-pro
[5] https://github.com/kyonofx/tia/
[6] https://github.com/facebookresearch/deep_bisim4control/
[7] https://github.com/facebookresearch/denoised_mdp
[8] https://github.com/facebookresearch/drqv2

DrQ-(I+P) builds on the official implementation and uses a separate encoder for the proprioception whose output is concatenated to the image encoder's output and trained using the critics' gradients.

We implemented *RePo* and *RePo(I+P)* in our framework, reused the Hyperparameters form Zhu et al. (2023), and ensured the results of our implementation match the official implementation's[9] result on the DMC tasks with standard images. *RePo(I+P)* encodes the proprioception using a separate encoder and both the embedded image and proprioception are given to the *RSSM*.

Ablations that are Similar to related Approaches. Some of our ablations are very similar to related approaches. The model-based *Img-Only* ablation with reconstruction loss, is very similar to *Dreamer-v1* (Hafner et al., 2020). It differs from the *Dreamer-v1* (Hafner et al., 2020) in using the KL-balancing introduced in (Hafner et al., 2021) and in regularizing the value function towards its own exponential moving average, as introduced in (Hafner et al., 2023).

However, there are considerable differences between the contrastive version of Dreamer-v1(Hafner et al., 2020) and the contrastive variational *Img-Only* ablation. In particular, those regard the exact form of mutual information estimation and the use of image augmentations.

The model-free contrastive predictive *Img-Only* and *Same-Loss* ablations are similar to the approach of Srivastava et al. (2021). The main difference is that Srivastava et al. (2021) includes the critic's gradients when updating the representation while in our setting no gradients flow from the actor or the critic to the representation. Furthermore, we did not include the inverse dynamics objective used by Srivastava et al. (2021) as we did not find it to be helpful. Additionally, we adapted some hyperparameters to match those of our other approaches.

C.1 Hyperparameters of Abltions and Baselines.

Ablations. All *Same-Loss*, *Concat*, and *Img-Only* use the hyperparameters listed in Appendix B. They are merely missing certain parts of the model or use a different loss for one or both modalities. For the *Concat* baseline, we project the proprioception to the *RSSMs* latent state size (stochastic + deterministic) using a single linear layer before concatenation.

ProprioSAC uses the hyperparameters listed in Table 3, except for the learning rates. We reduced those to the SAC default values of 0.0003 for all environments, as we found the more aggressive updates used for *CoRAL* on *Video Background*, *Occlusions* and *Locomotion* can lead to instabilities when training directly on the proprioception.

Baselines. All our baselines were originally evaluated on standard DeepMind Control Suite tasks, modified DeepMind Control Suite tasks, or both. They were designed for problems very similar to *Occlusions* and, in particular, *Video Background* and we thus reuse the Hyperparameters originally proposed by the respective authors. For baselines using an *RSSM*, (*TIA*, *DreamerPro*, *Denoised-MDP*, and *RePo*) these are generally very similar and follow Hafner et al. (2020; 2021).

For the *Locomotion* suite all approaches, including *CoRAL* and the ablations, use the same Hyperparameters as they use for *Video Backgrounds* and *Occlusions*.

For the *Manipiluation* suite we increased the model sizes of *RePo* following those of *CoRAL*. For both the *DrQ-v2*-based and the *RePo*-based baselines we tried a discount factor of 0.85 and 0.99 to ensure the performance differences to *CoRAL* is not an artifact of the low discount of 0.85. However, the lower discount worked better for all methods.

C.2 On the Performance of Some Baselines in our Setting.

As described in Section A.1, there are distinct ways how to select and use the Kinetics400 videos in the existing literature. Nguyen et al. (2021), who first introduced the more challenging setting we use, already found *DBC* (Zhang et al., 2020) to struggle in this setting and our results align with those findings.

[9] https://github.com/zchuning/repo

TIA (Fu et al., 2021) and *DenoisedMDP* (Wang et al., 2022) factorize the latent variable into 2 distinct parts and formulate loss functions that force one part to focus on task-relevant aspects and the other on task-irrelevant aspects. However, the part responsible for the task-irrelevant aspects still has to model those explicitly. In the more complicated setting with randomly sampled, colored background videos, the *TIA* and *DenoisedMDP* world models underfit and thus fail to learn a good representation or policy. Contrastive approaches, such as our approach and *DreamerPro* (Deng et al., 2022), do not struggle with this issue, as they do not have to model task-irrelevant aspects but can learn to ignore them.

RePo (Zhu et al., 2023) was also evaluated on the simpler setting and Zhu et al. (2023) report an improved performance over *TIA* and *DenoisedMDP*. In the more challenging setting, this improvement persists and *RePo* performs similarly to *DreamerPro* (Fig. 2).

Furthermore, Zhu et al. (2023) presents results on ManiSkill2 environments similar to `LiftCube`, `PushCube`, and `TurnFaucet` of our *Manipulation* Suite. However, as detailed in Appendix A.3 our Manipulation Suite tasks randomize initial conditions (i.e., cube position or faucet model) which results in significantly more challenging tasks, in which *RePo* seems to struggle.

D Complete Results

The following pages list the aggregated results and performance profiles for all tasks, representation-learning approaches, and both model-free and model-based RL. We compute inter-quartile means and stratified bootstrapped confidence intervals, as well as the performance profiles according to the recommendations of Agarwal et al. (2021) using the provided library[10]. For each task in the suites, we ran 5 seeds per method, i.e., the results for *Standard Images*, *Video Backgrounds*, and *Occlusions* are aggregated over 35 runs, and those for *Locomotion* over 30 runs. For *OpenCabinetDrawer* we run 20 seeds per method. Fig. 9 lists the aggregated results for all model-free agents on the DeepMind Control (DMC) Suite tasks and Fig. 11 lists the corresponding performance profiles. Fig. 10 lists the aggregated results for all model-based agents on the DeepMind Control Suite tasks and Fig. 12 lists the corresponding performance profiles. Fig. 13 shows aggregated results and performance profiles for the *Locomotion* suite. Fig. 14 shows aggregated results and performance profiles for the *Manipulation* suite. We also list the per-task results for all task suits:

- Fig. 15: Model-free agents on DMC tasks with *Standard Images*
- Fig. 16: Model-free agents on DMC tasks with *Video Background*.
- Fig. 17: Model-free agents on DMC tasks with *Occlusions*.
- Fig. 18: Model-based agents on DMC tasks with *Standard Images*.
- Fig. 19: Model-based agents on DMC tasks with *Video Background*.
- Fig. 20: Model-based agents on DMC tasks with *Occlusions*.
- Fig. 21: Per Environment Results for the *Locomotion* suite.
- Fig. 22: Per Environment Results for the *Manipulation* suite.

[10] https://github.com/google-research/rliable

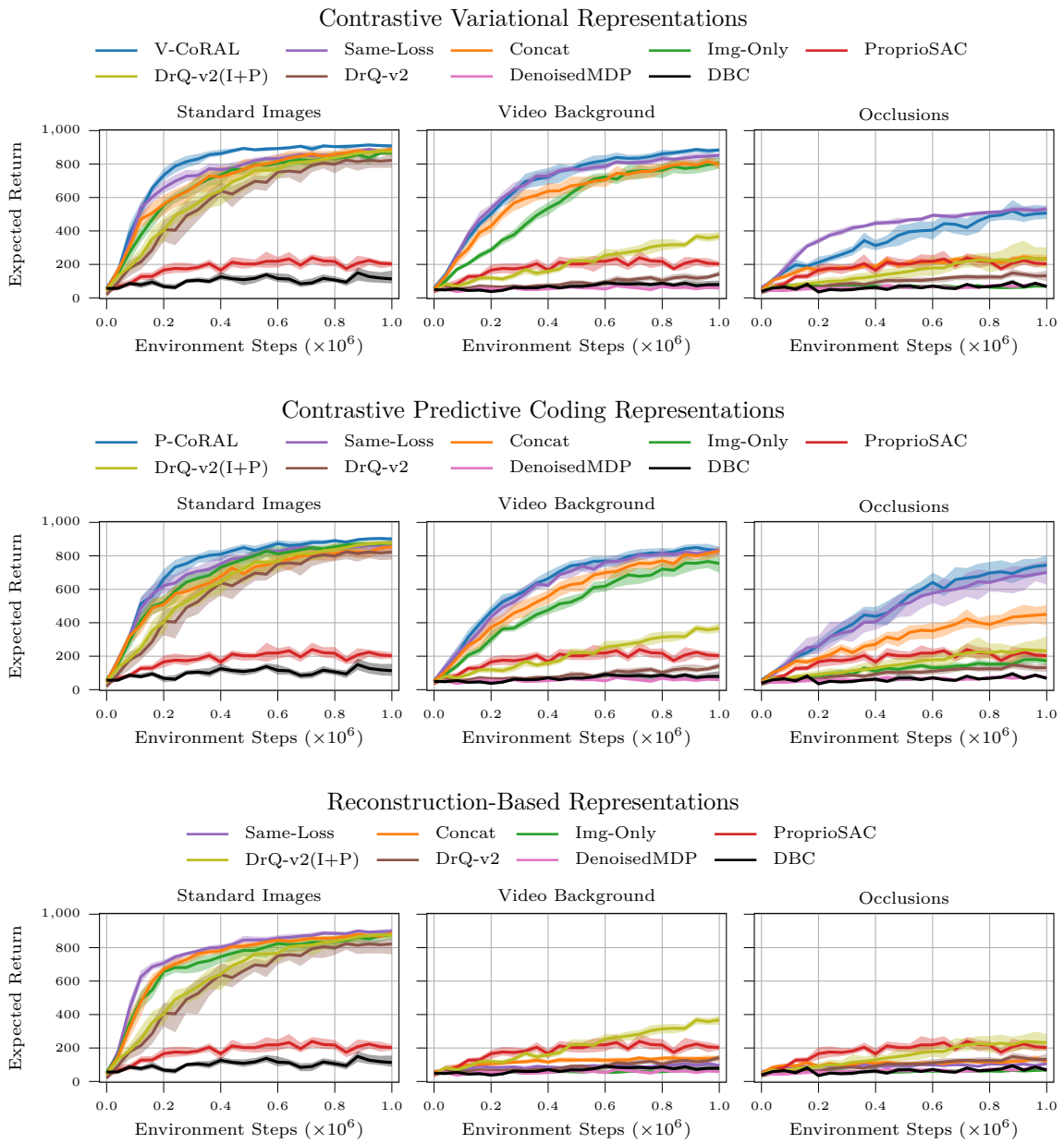

Figure 9: Aggregated results for all **model-free** agents on the DeepMind Control Suite environments with *Standard Images*, *Video Background*, and *Occlusions*. As expected, reconstruction-based approaches do not work on *Video Background* and *Occlusions*. Out of all considered approaches *V-CoRAL* achieves the highest performance on *Video Background* and *P-CoRAL* achieves the highest performance on *Occlusions*.

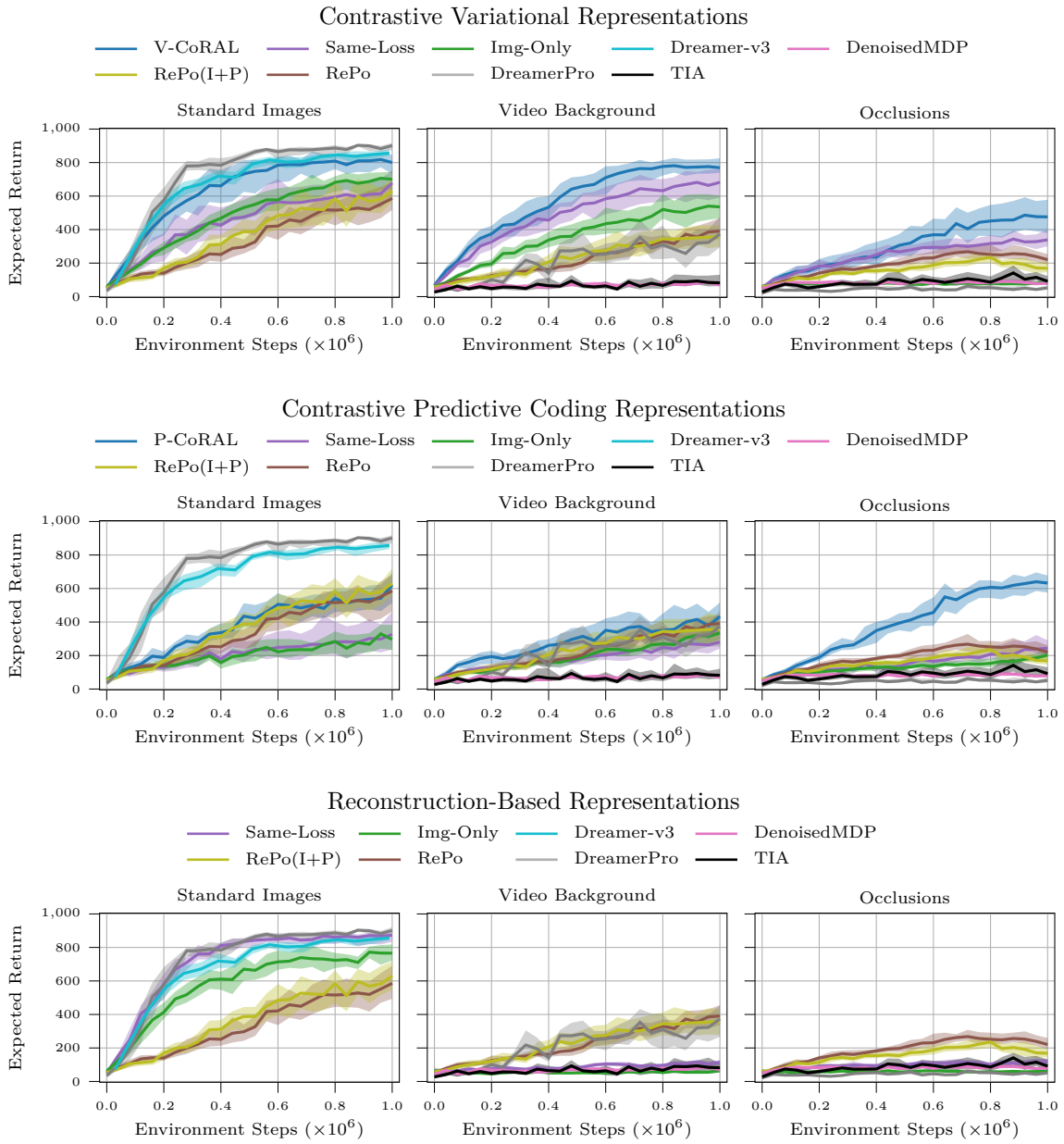

Figure 10: Aggregated results for all **model-based** agents on the DeepMind Control Suite environments with *Standard Images*, *Video Background*, and *Occlusions*. Compared to their model-free counterparts (Fig. 9), model-based agents perform worse, except if a reconstruction-based representation is used. Yet, the performance gap is larger for image-only and fully contrastive approaches. Especially *V-CoRAL* still achieves high performance on *Video Background*, almost matching the performance of *Dreamer-v3* on *Standard Images*. This further highlights the benefits of using *CoRAL*, which can significantly improve over tailored approaches such as *DreamerPro* or *RePo*.

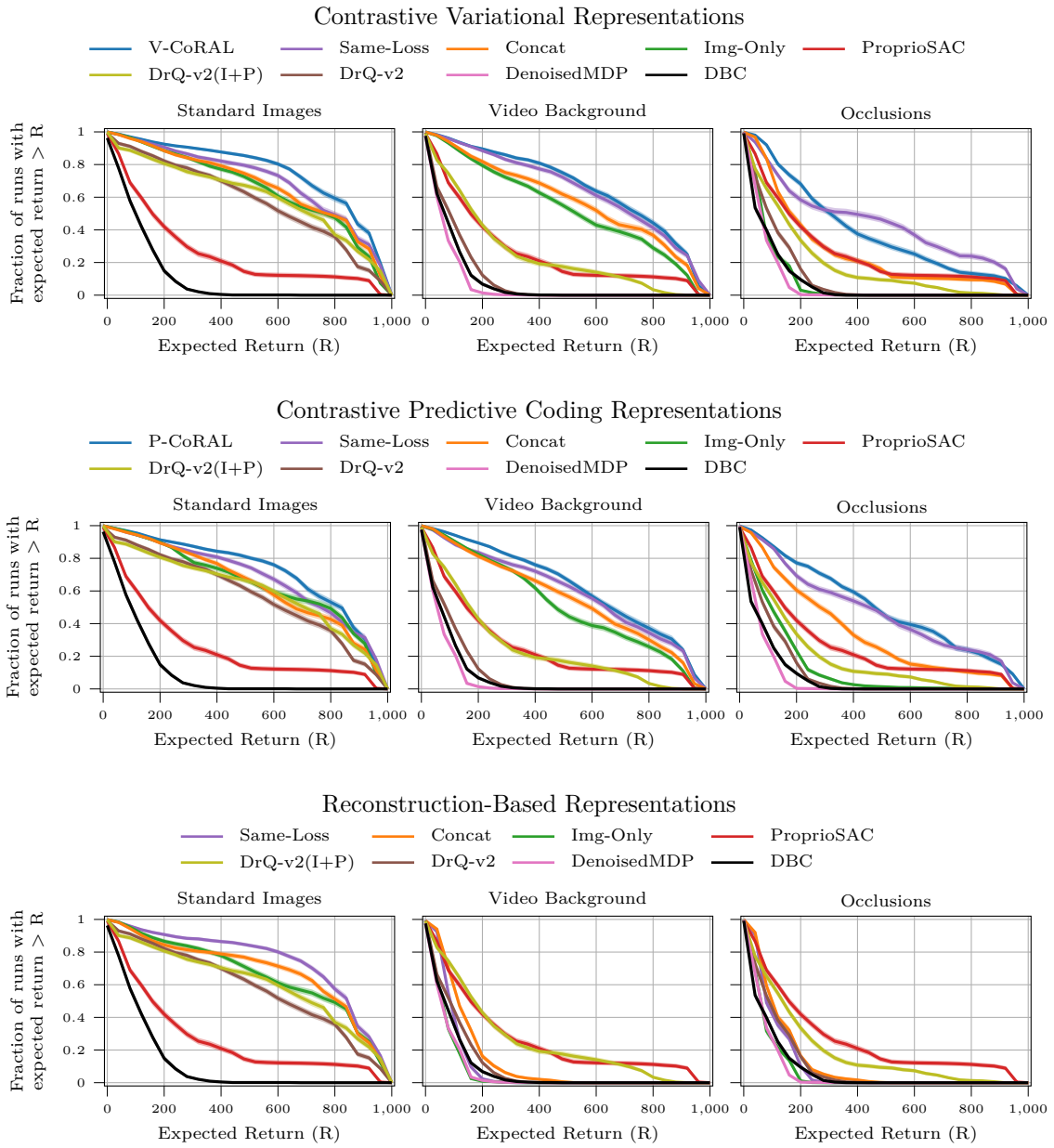

Figure 11: Performance profiles for all **model-free** agents on the DeepMind Control Suite tasks with *Standard Images*, *Video Background*, and *Occlusions*. They show that performance is largely consistent across the tasks. The sole exception is *V-CoRAL* and the contrastive variational approach with the same loss for both modalities on *Occlusions*. Here, the former fails for `Ball-in-Cup Catch` and `Cartpole Swingup`, while the latter underperforms for `Cheetah Run` (Fig. 17).

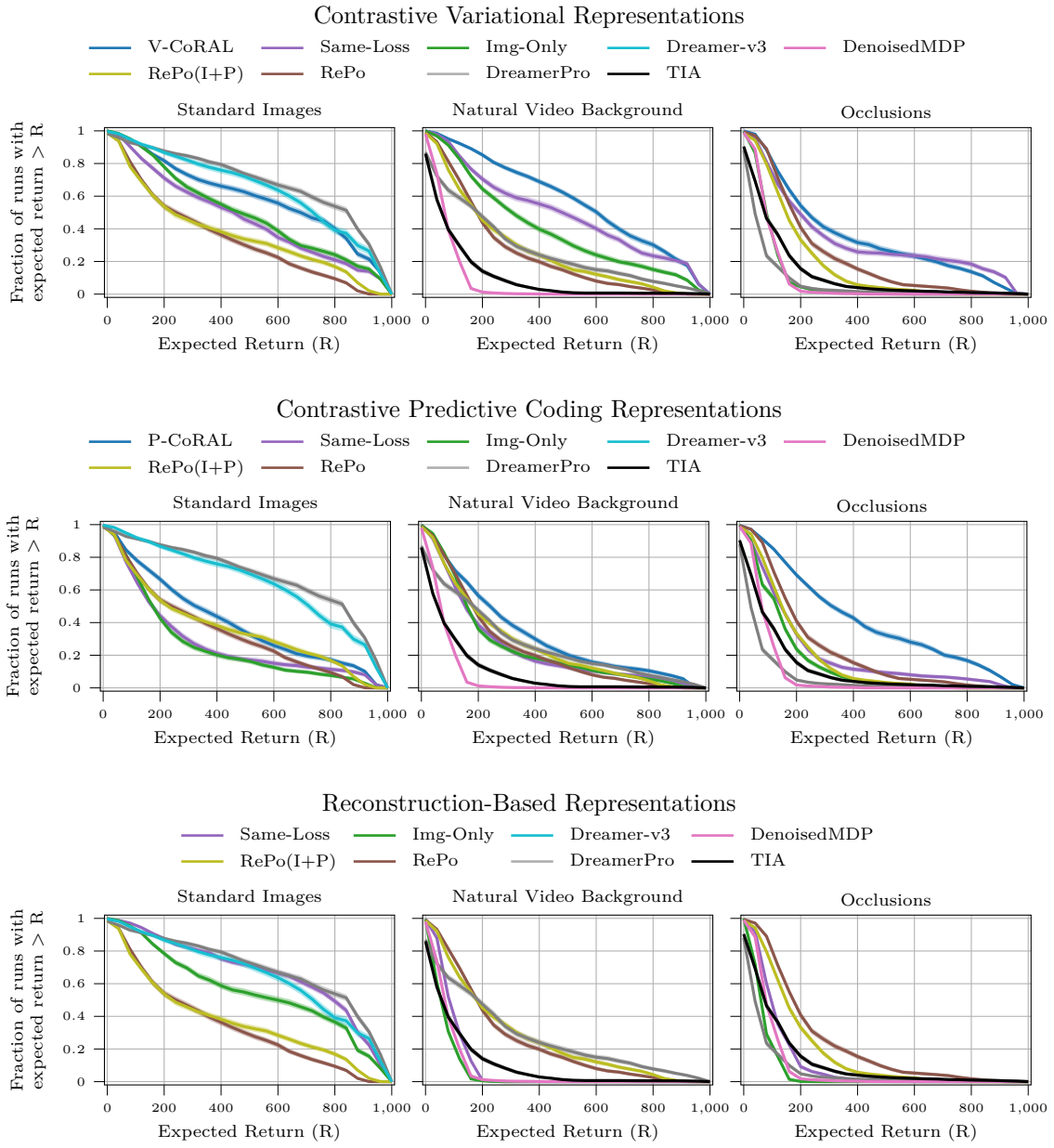

Figure 12: Performance profiles for all **model-based** agents on the DeepMind Control Suite environments with *Standard Images*, *Video Background*, and *Occlusions*. They indicate that performance is largely consistent across the environments.

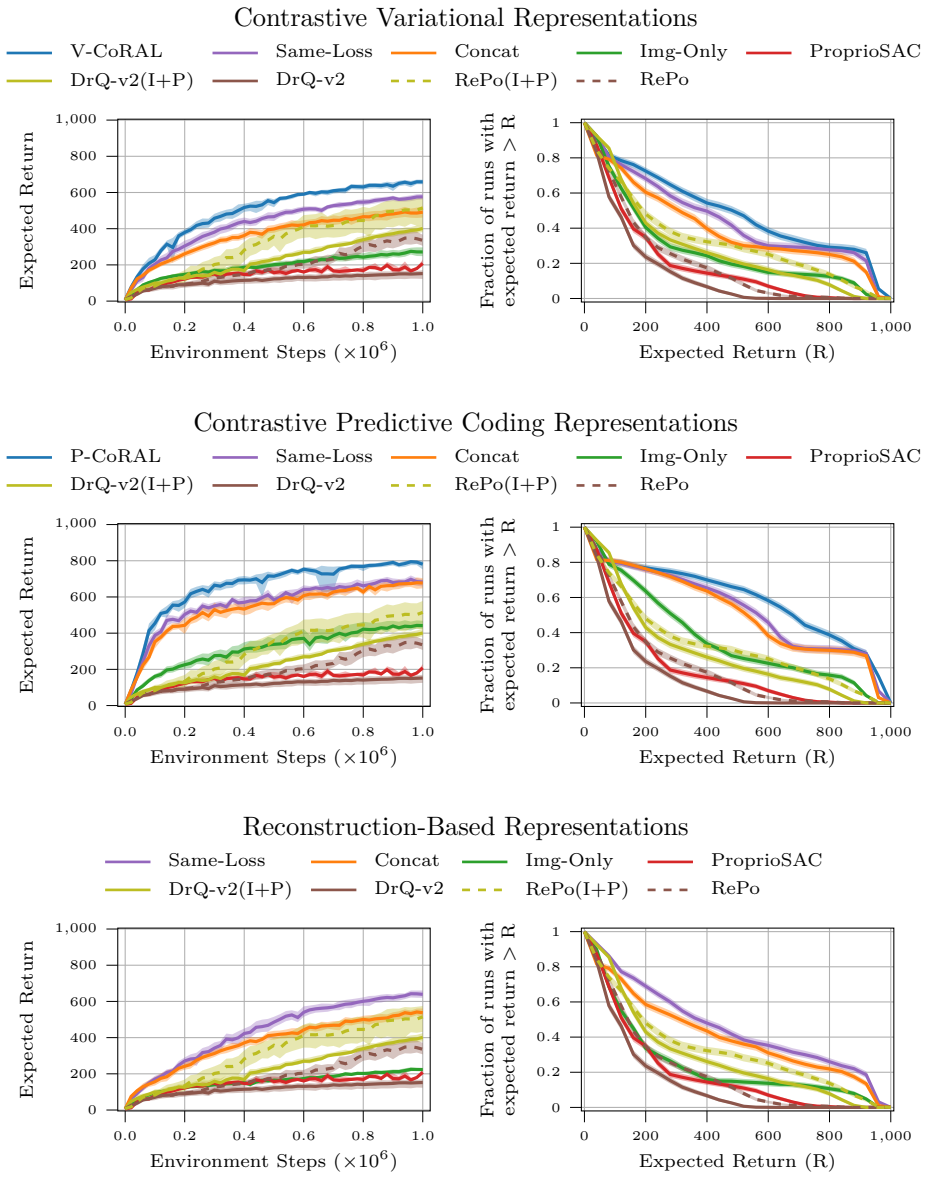

Figure 13: Aggregated results and performance profiles for the *Locomotion* suite. Both *V-CoRAL* and *P-CoRAL* outperform reconstruction and *P-CoRAL* gives the best results of all approaches by a significant margin Fig. 21 shows that the performance difference is larger in environments with randomly colored obstacles (`Hurdle Cheetah Run`, `Hurdle Walker Walk`, `Hurdle Walker Run`. The color is not relevant to avoid the obstacles but seems to hinder reconstruction.

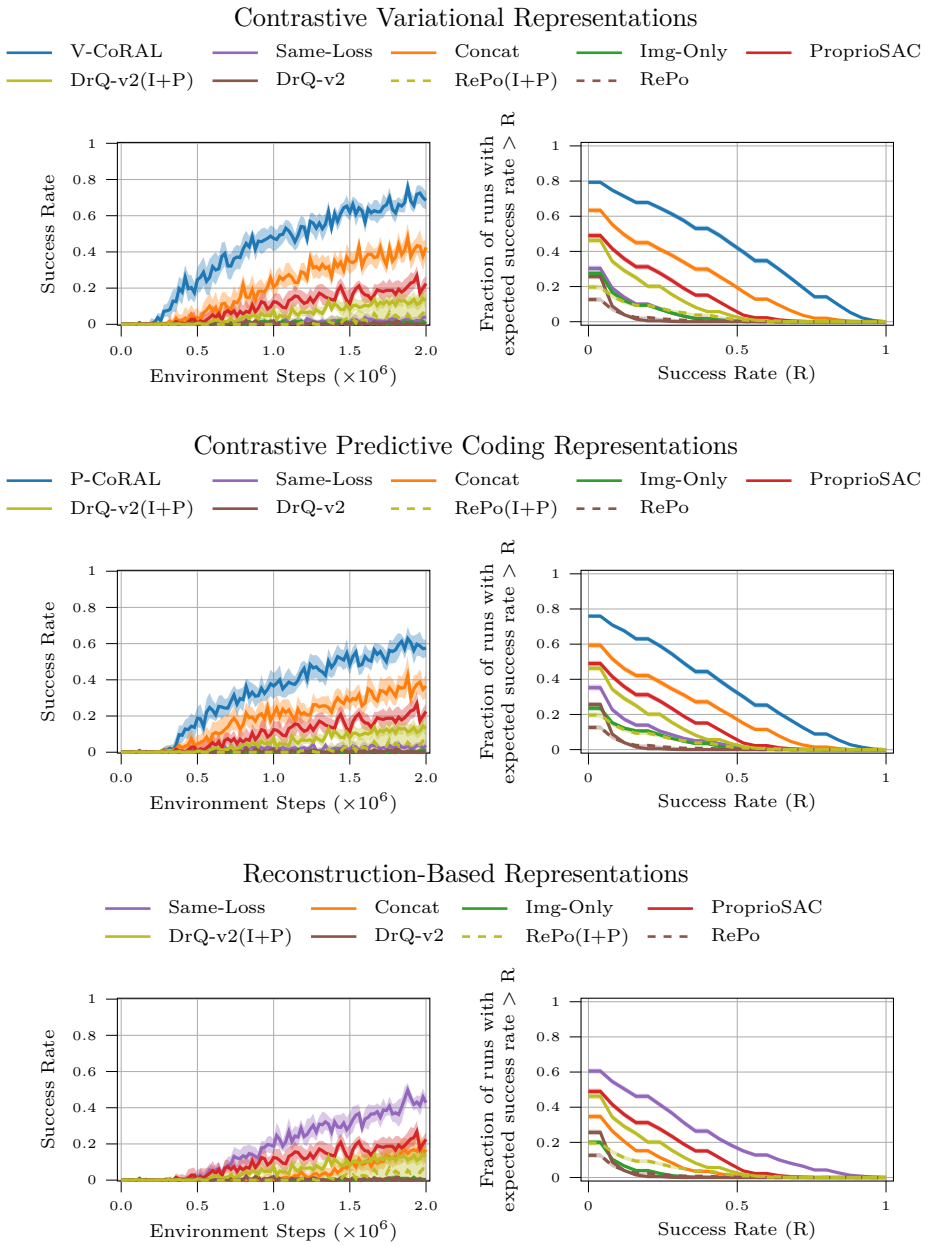

Figure 14: Aggregated results and performance profiles for the *Manipulation* Suite. *V-CoRAL* performs best by a significant margin, followed by *P-CoRAL* No approach that uses solely images, i.e., *Img Only*-ablations, *RePo* and *DrQ-v2*, or uses both modalities but has a fully constrastive objective achieves any notable success.

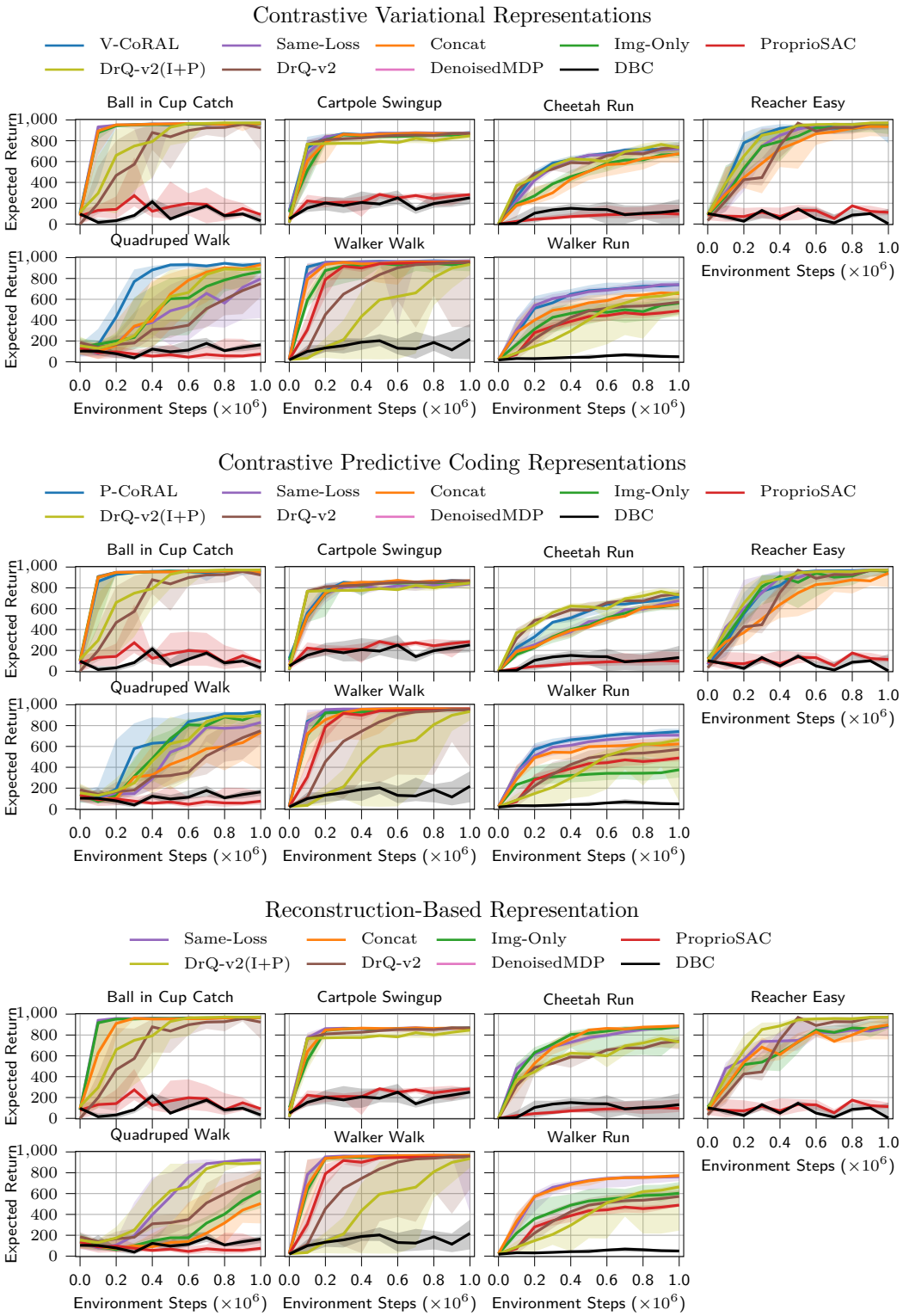

Figure 15: Per environment results for model-free agents on the DeepMind Control Suite with *Standard Images*.

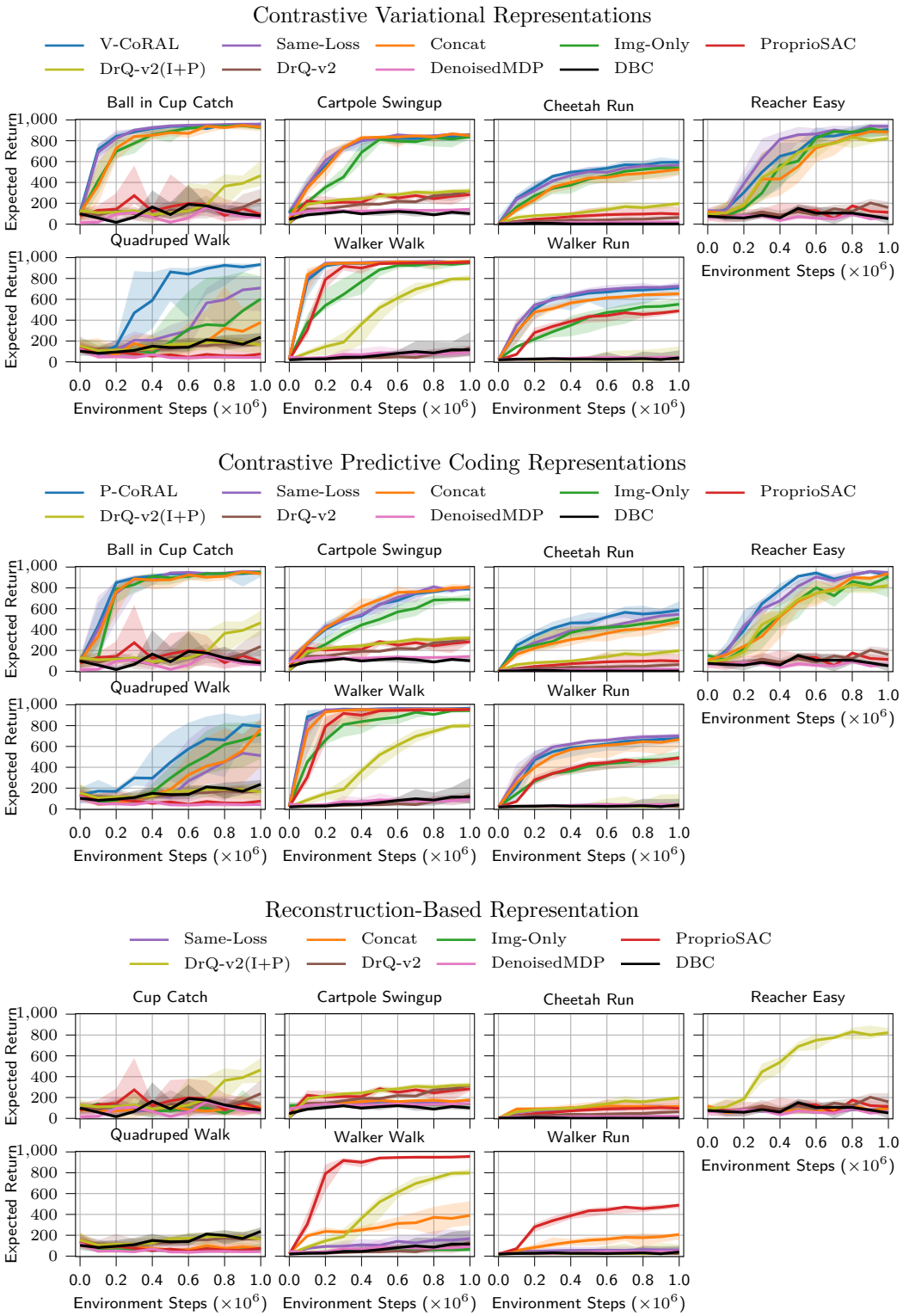

Figure 16: Per environment results for model-free agents on the DeepMind Control Suite with *Video Background*.

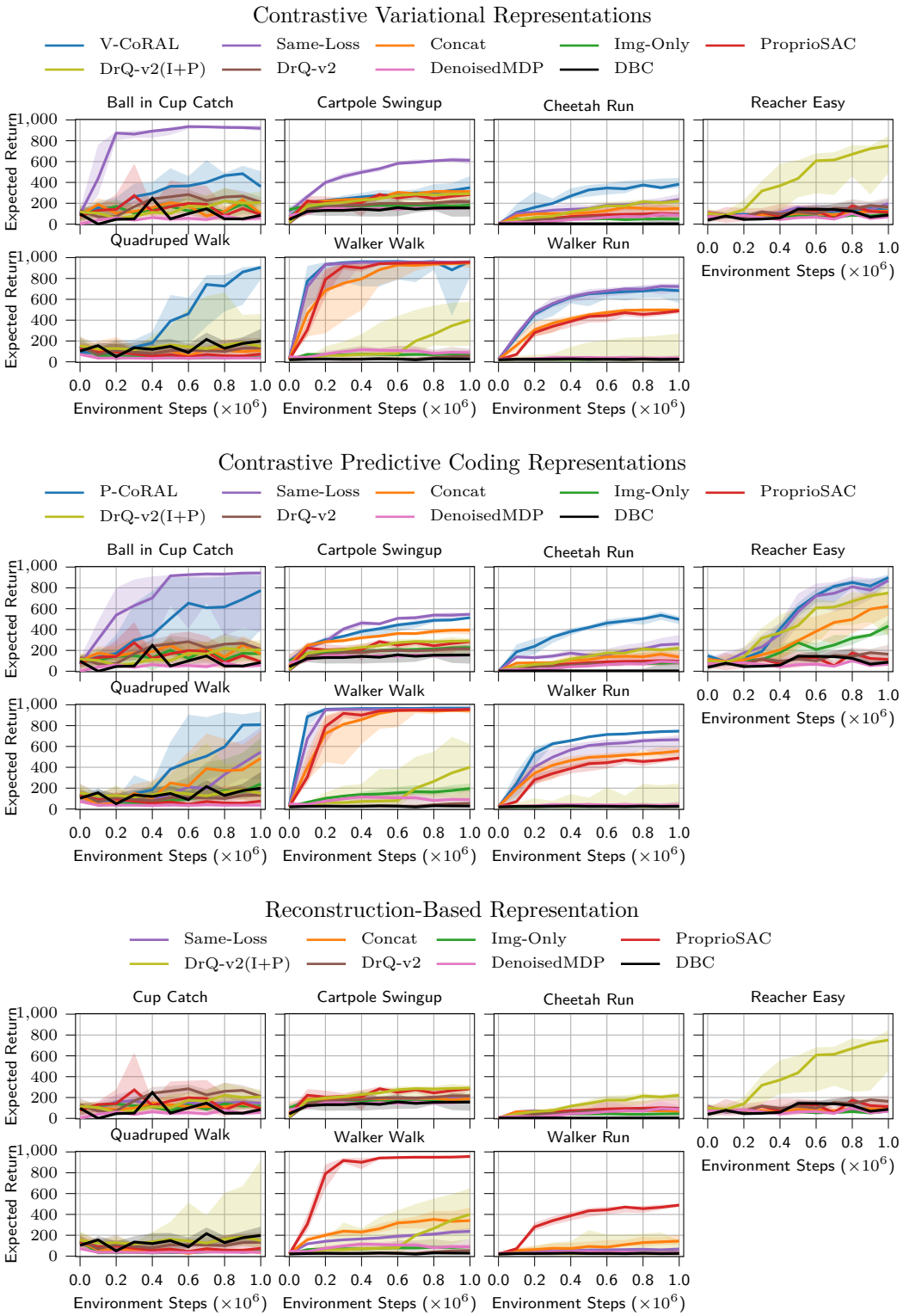

Figure 17: Per environment results for model-free agents on the DeepMind Control Suite with *Occlusions*.

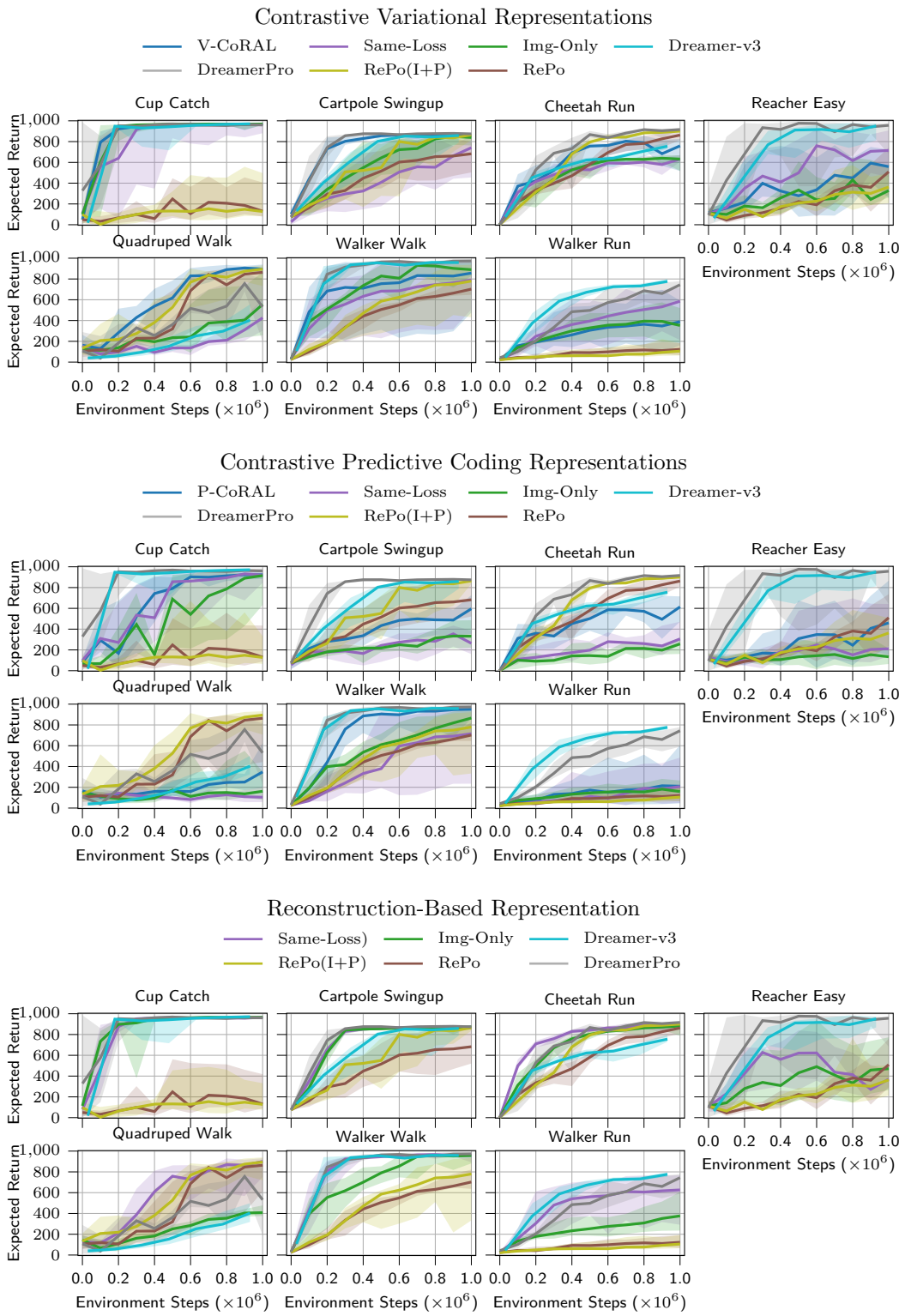

Figure 18: Per environment results for model-based agents on the DeepMind Control Suite with *Standard Images*.

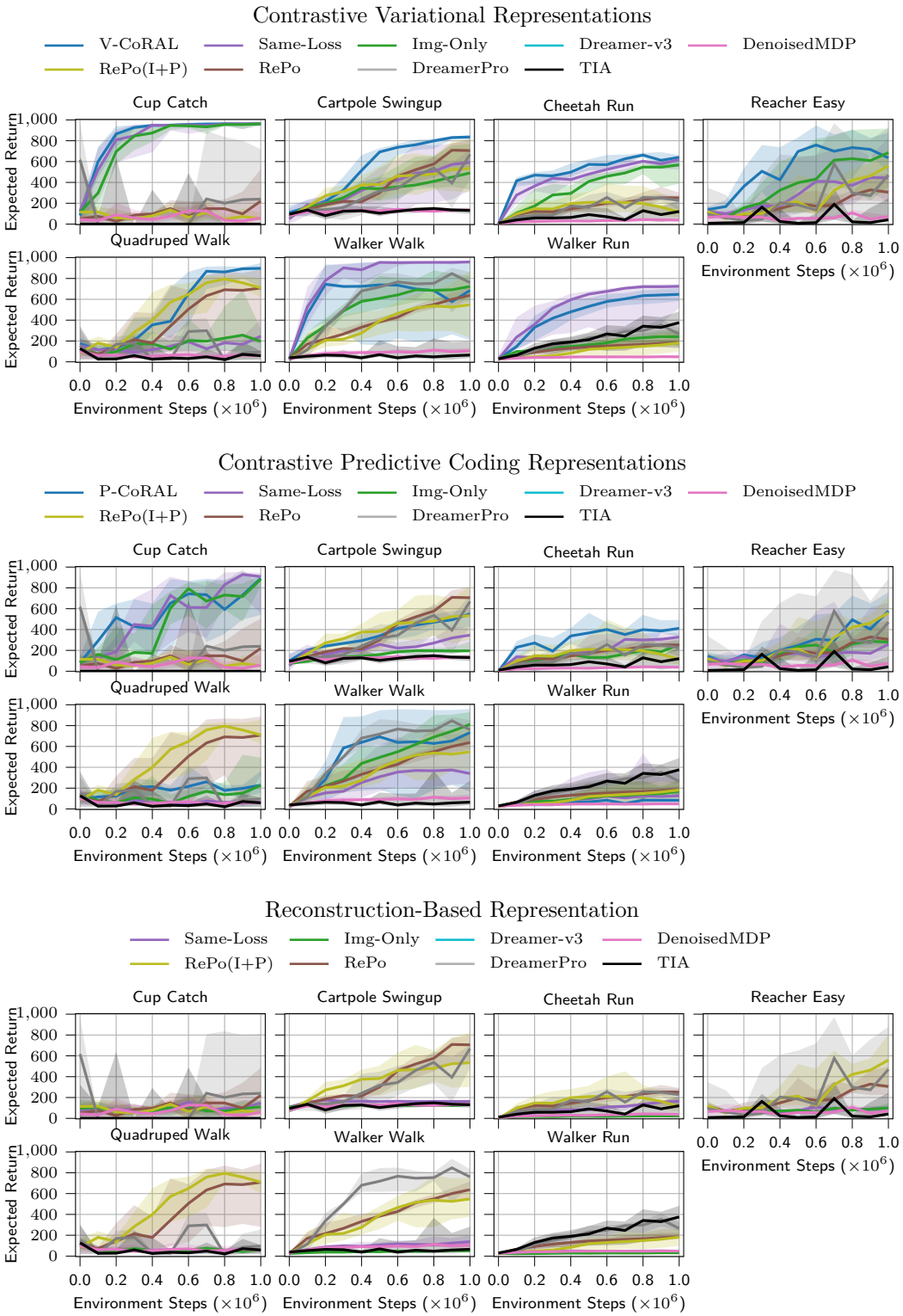

Figure 19: Per environment results for model-based agents on the DeepMind Control Suite with *Video Background*.

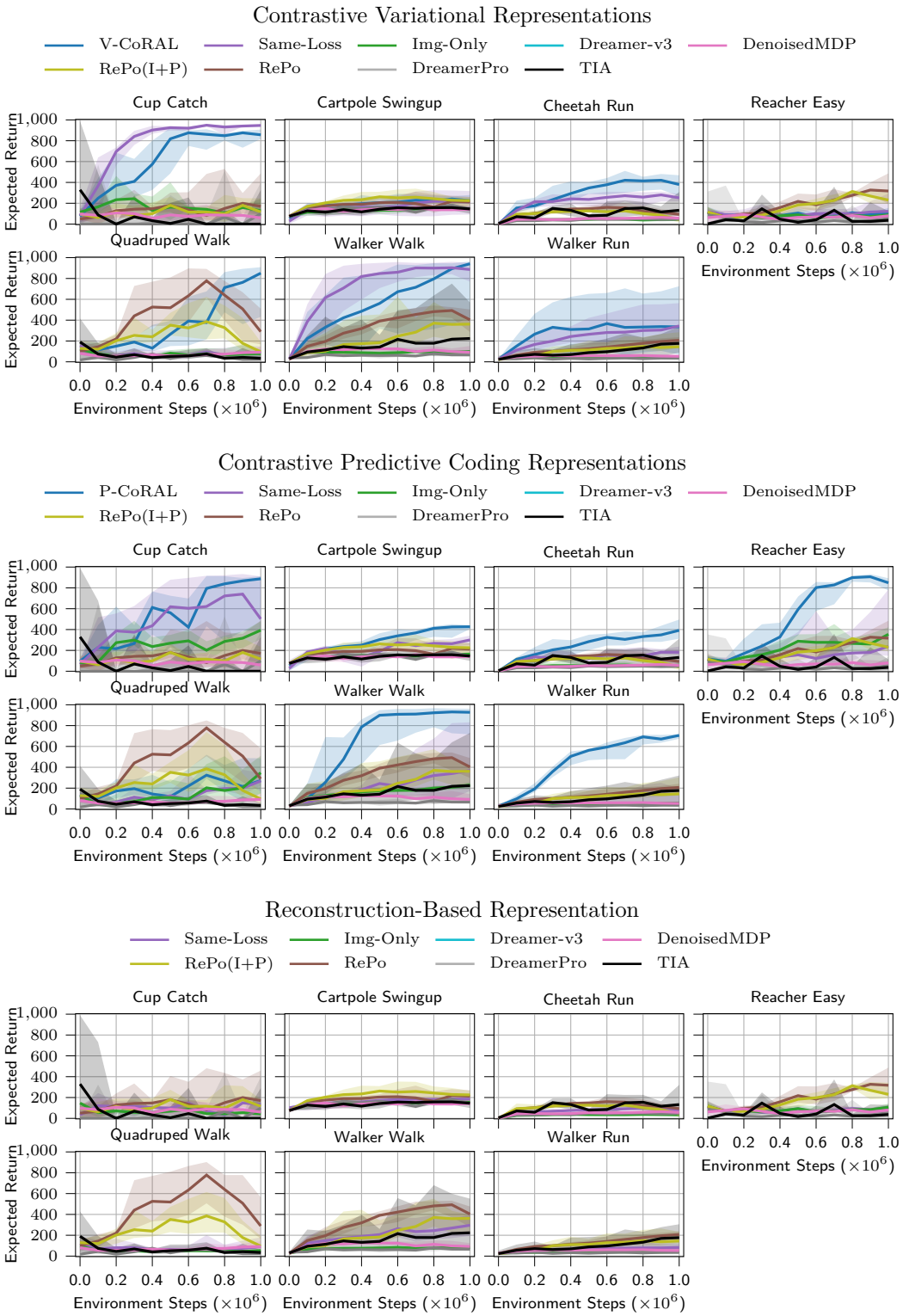

Figure 20: Per environment results for model-based agents on the DeepMind Control Suite with *Occlusions*.

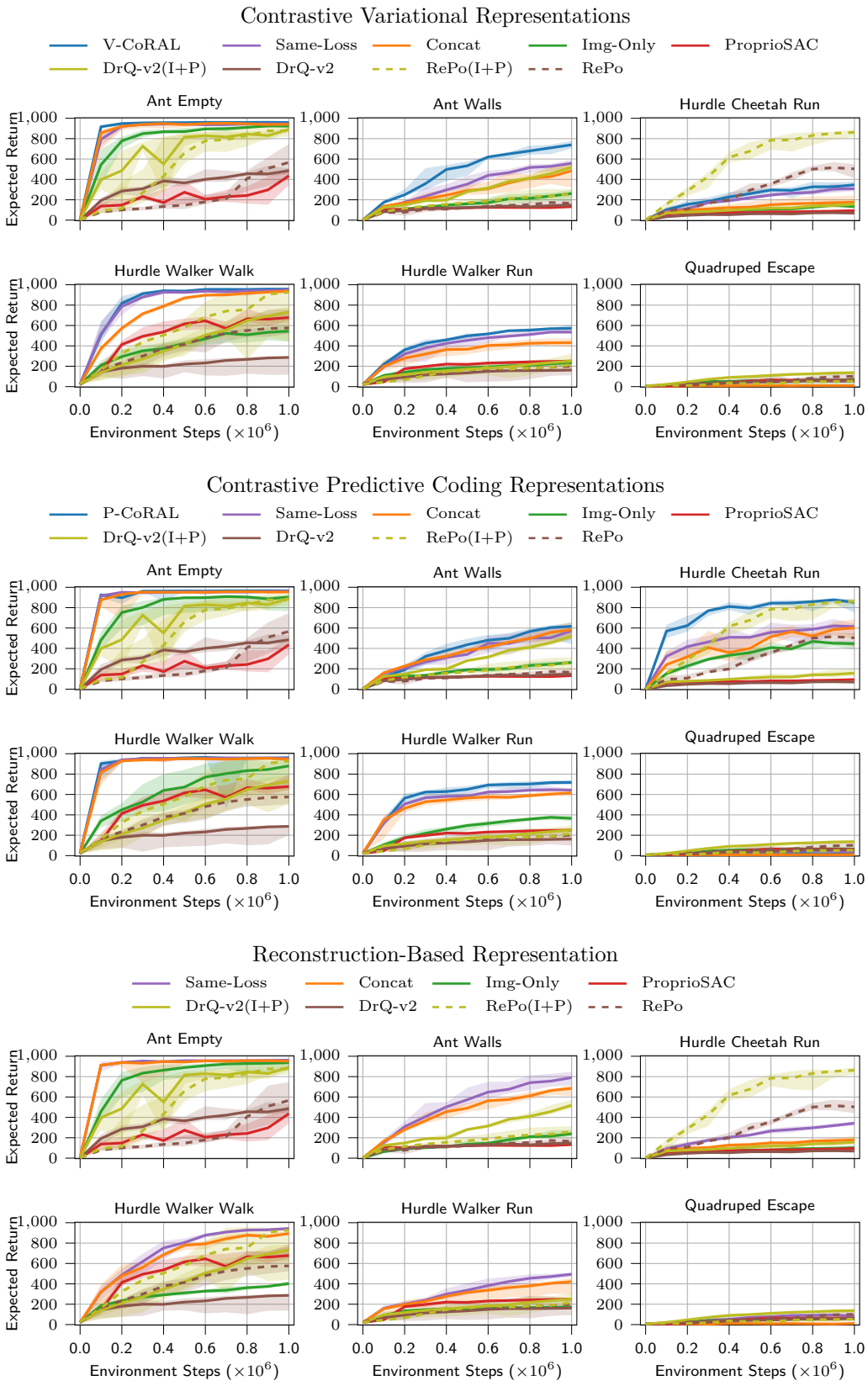

Figure 21: Per environment results for the *Locomotion* suite.

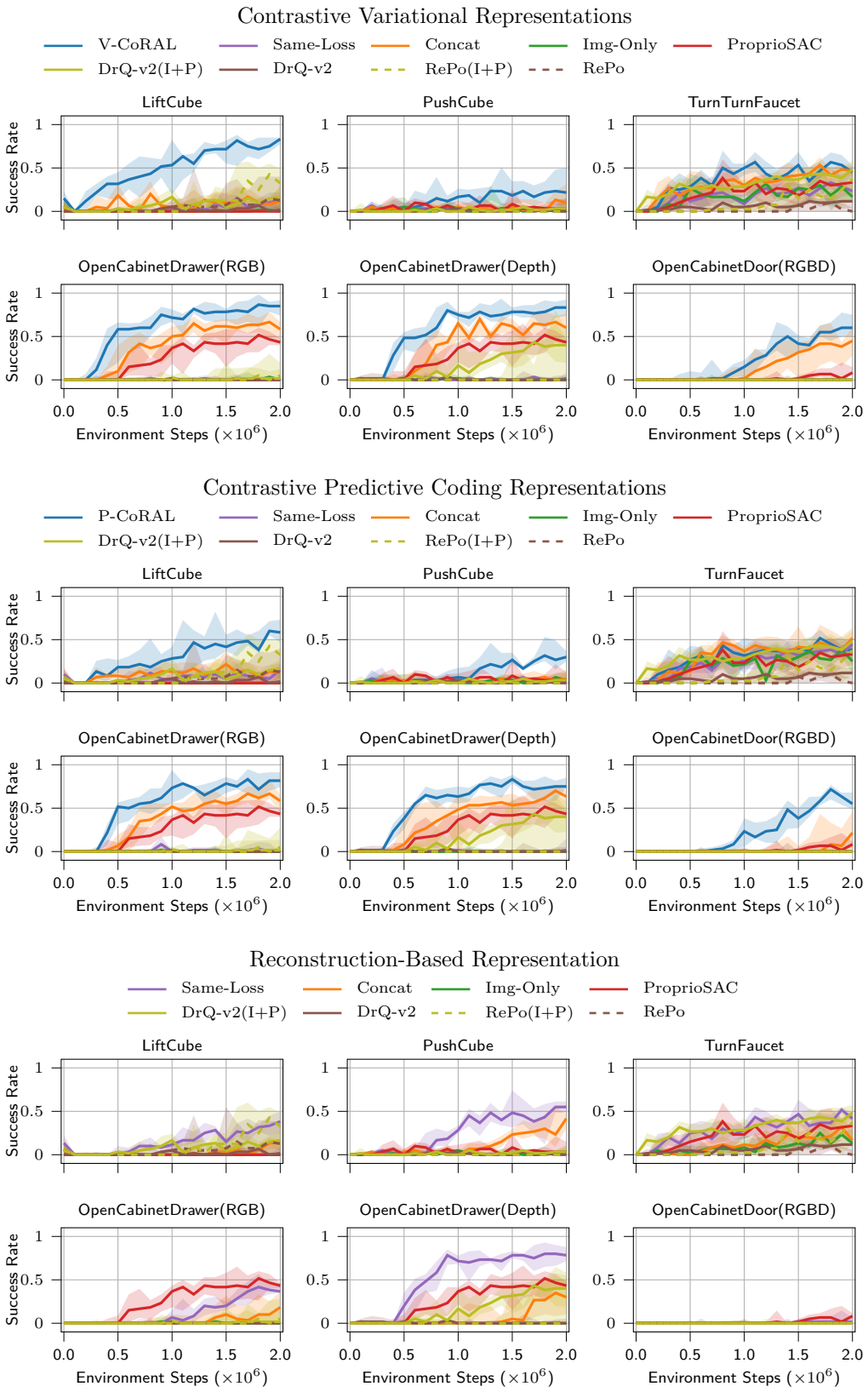

Figure 22: Per environment results for the *Manipulation* suite.
Reconstruction-Based Representation