Learning Representations for Pixel-based Control: What Matters and Why?

Manan Tomar[*12], **Utkarsh A. Mishra**[*12], **Amy Zhang**[34] **& Matthew E. Taylor**[12]
[1] **University of Alberta** [2] **Amii** [3] **FAIR, Menlo Park** [4] **UC Berkeley**

Reviewed on OpenReview: https://openreview.net/forum?id=wIXHG8LZ2w

Abstract

Learning representations for pixel-based control has garnered significant attention recently in reinforcement learning. A wide range of methods have been proposed to enable efficient learning, leading to sample complexities similar to those in the full state setting. However, moving beyond carefully curated pixel data sets (centered crop, appropriate lighting, clear background, etc.) remains challenging. In this paper, we adopt a more difficult setting, incorporating background distractors, as a first step towards addressing this challenge. We start by exploring a simple baseline approach that does not use metric-based learning, data augmentations, world-model learning, or contrastive learning. We then analyze when and why previously proposed methods are likely to fail or reduce to the same performance as the baseline in this harder setting and why we should think carefully about extending such methods beyond the well curated environments. Our results show that finer categorization of benchmarks on the basis of characteristics like density of reward, planning horizon of the problem, presence of task-irrelevant components, etc., is crucial in evaluating algorithms. Based on these observations, we propose different metrics to consider when evaluating an algorithm on benchmark tasks. We hope such a data-centric view can motivate researchers to rethink representation learning when investigating how to best apply RL to real-world tasks. Code available: https://github.com/UtkarshMishra04/pixel-representations-RL

1 Introduction

Learning useful representations for downstream tasks is a key component for success in rich observation environments (Du et al., 2019; Mnih et al., 2015; Silver et al., 2017; Wahlström et al., 2015; Watter et al., 2015). Consequently, a significant amount of work proposes various representation learning objectives that can be tied to the original reinforcement learning (RL) problem. Such auxiliary objectives include the likes of contrastive learning losses (Oord et al., 2018; Laskin et al., 2020b; Chen et al., 2020), state similarity metrics like bisimulation or policy similarity (Zhang et al., 2020b;a; Agarwal et al., 2021a), and pixel reconstruction losses (Jaderberg et al., 2016; Gelada et al., 2019; Hafner et al., 2020; 2019). On a separate axis, data augmentations have been shown to provide huge performance boosts when learning to control from pixels (Laskin et al., 2020a; Kostrikov et al., 2020). Each of these methods has been shown to work well for particular settings and hence displayed promise to be part of a general purpose representation learning toolkit. Unfortunately, these methods were proposed with different motivations and tested on different tasks, making the following question hard to answer:

What really matters when learning representations for downstream control tasks?

Learning directly from pixels offers much richer applicability than when learning from carefully constructed states. Consider the example of a self-driving car, where it is nearly impossible to provide a complete state description of the position and velocity of all objects of interest, such as road edges, highway markers, other vehicles, etc. In such real world applications, learning from pixels offers a much more feasible option. However, this requires algorithms that

[*]Equal Contribution. Corresponding author: manan.tomar@gmail.com

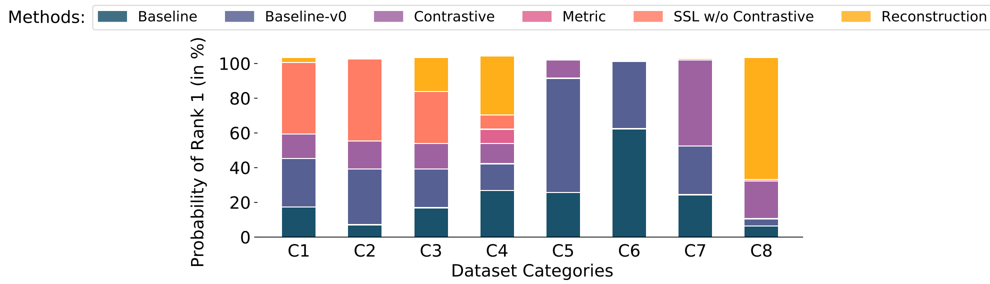

Figure 1: **Comparing pixel-based RL methods across finer categorizations** of evaluation benchmarks. Each category 'Cx' denotes different data-centric properties of the evaluation benchmark (e.g., C1 refers to discrete action, dense reward, without distractors, and with data randomly cropped (Kostrikov et al., 2020; Laskin et al., 2020a)). Exact descriptions of each category and the algorithms are provided in Table 4 and Table 5. Baseline-v0 refers to applying the standard deep RL agent (e.g., Rainbow DQN (van Hasselt et al., 2019) and SAC (Haarnoja et al., 2018)); Baseline refers to adding reward and transition prediction to baseline-v0, as described in Section 3; Contrastive includes algorithms such as PI-SAC Lee et al. (2020b) and CURL Laskin et al. (2020b); Metric denotes the state metric losses such as DBC (Zhang et al., 2020b); SSL w/o Contrastive includes algorithms such as SPR (Schwarzer et al., 2020); Reconstruction [1] includes DREAMER (Hafner et al., 2020) and TIA (Fu et al., 2021). For a given method, we always consider the best performing algorithm. Every method leads to varied performance across data categories, making a comparison which is an *average across all categories* highly uninformative.

can discern between task-relevant and task-irrelevant components in the pixel input, i.e., learn good representations. Focusing on task-irrelevant components can lead to brittle or non-robust behavior when put in slightly different environments. For instance, billboard signs over buildings in the background have no dependence on the task in hand while a self-driving car tries to change lanes. However, if such task-irrelevant components are not discarded, they can lead to sudden failure when the car drives through a different environment, say a forest where there are no buildings or billboards. Avoiding brittle behavior is therefore key to efficient deployment of artificial agents in the real world.

There has been a lot of work recently that tries to learn efficiently from pixels. A dominant idea throughout prior work has been that of attaching an auxiliary loss to the standard RL objective, with the exact mechanics of the loss varying for each method (Jaderberg et al., 2016; Zhang et al., 2020b; Laskin et al., 2020b). A related line of work learns representations by constructing world models directly from pixels (Schmidhuber, 2010; Oh et al., 2015; Ha & Schmidhuber, 2018; Hafner et al., 2020). We show that these work well when the world model is simple. However, as the world model gets even slightly more complicated, which is true of the real world and imitated in simulation with the use of video distractors (Zhang et al., 2018; Kay et al., 2017; Stone et al., 2021), such approaches can fail. For other methods, it is not entirely clear what component/s in auxiliary objectives can lead to failure when changing the environment, thus making robust behavior hard to achieve. Another distinct idea is of using data augmentations (Laskin et al., 2020a; Kostrikov et al., 2020) over the original observation samples, which seem to be quite robust across different environments. However, as we will show, a lot of the success of data augmentations is an artifact of how the benchmark environments save data, which is not replicable in the real world (Stone et al., 2021), thus resulting in failure[2]. It is important to note that some of these methods are not designed for robustness but instead for enhanced performance on particular benchmarks. For instance, the ALE (Bellemare et al., 2013) benchmark involves simple, easy to model objects, and it becomes hard to discern if methods that perform well are actually good candidates for answering 'what really matters for robust learning in the real world.'

Contributions. In this paper, we explore the major components responsible for the successful application of various representation learning algorithms. Based on recent work in RL theory for learning with rich observations (Farahmand et al., 2017; Ayoub et al., 2020; Castro, 2020), we hypothesize certain key components to be responsible for sample efficient learning. We test the role these play in previously proposed representation learning objectives and then consider an exceedingly simple *baseline* (see Figure 2) which takes away the extra "knobs" and instead combines two simple but key ideas, that of reward and transition prediction. We conduct experiments across multiple settings, including the MuJoCo domains from DMC Suite (Tassa et al., 2018) with natural distractors (Zhang et al., 2018; Kay et al., 2017; Stone et al., 2021), and Atari100K Kaiser et al. (2019) from ALE (Bellemare et al., 2013). Following this, we identify the failure modes of previously proposed objectives and highlight why they result in comparable or worse performance than the considered baseline. Our observations suggest that relying on a particular method across multiple evaluation settings does not work, as the efficacy varies with the exact details of the task, even within the same benchmark (see Figure 1). We note that a finer categorization of available benchmarks based on metrics like density of reward, presence

[1]For ALE we use the performance of DREAMER after 1M steps, whereas for DMC we consider the performance after 500k steps.
[2]It is also hard to pick exactly which data augmentation will work for a particular environment or task (Raileanu et al., 2020; Grigsby & Qi, 2020)

of task-irrelevant components, inherent horizon of tasks, etc., play a crucial role in determining the efficacy of a method. We list such categorizations as suggestions for more informative future evaluations. The findings of this paper advocate for a more data-centric view of evaluating RL algorithms (Co-Reyes et al., 2020), largely missing in current practice. We hope the findings and insights presented in this paper can lead to better representation learning objectives for real-world applications.

2 Related Work

Prior work on **auxiliary objectives** includes the Horde architecture (Sutton et al., 2011), UVFA (Schaul et al., 2015) and the UNREAL agent (Jaderberg et al., 2016). These involve making predictions about features or pseudo-rewards, however only the UNREAL agent used these predictions for learning representations. Even so, the benchmark environments considered there always included only task-relevant pixel information, thus not pertaining to the hard setting we consider in this work. Representations can also be fit so as to obey certain state similarities. If these **state metrics** preserve the optimal policies and are easy to learn/given a priori, such a technique can be very useful. Recent works have shown that we can learn efficient representations either by learning the metrics like that in bisimulation (Ferns et al., 2011; Zhang et al., 2020b;a; Biza et al., 2020), by recursively sampling states (Castro et al., 2021) or by exploiting sparsity in dynamics (Tomar et al., 2021). **Data augmentations** modify the input image into different distinct views, each corresponding to a certain type of modulation in the pixel data. These include cropping, color jitter, flip, rotate, random convolution, etc. Latest works (Laskin et al., 2020a; Yarats et al., 2021b) have shown that augmenting the state samples in the replay buffer with such techniques alone can lead to impressive gains when learning directly from pixels. Recently, Stone et al. (2021) illustrated the advantages and failure cases of augmentations. **Contrastive learning** involves optimizing for representations such that positive pairs (those coming from the same sample) are pulled closer while negative pairs (those coming from different samples) are pushed away (Oord et al., 2018; Chen et al., 2020). The most widely used method to generate positive/negative pairs is through various data augmentations (Laskin et al., 2020b; Schwarzer et al., 2020; Lee et al., 2020b; Stooke et al., 2021; Yarats et al., 2021a). However, temporal structure can induce positive/negative pairs as well. In such a case, the positive pair comes from the current state and the actual next state while the negative pair comes from the current state and any other next state in the current batch (Oord et al., 2018). Other ways of generating positive/negative pairs can be through learnt state metrics (Agarwal et al., 2021a) or encoding instances (Hafner et al., 2020). Another popular idea for learning representations is learning world models (Ha & Schmidhuber, 2018) in the pixel space. This involves learning prediction models of the world in the pixel space using a **pixel reconstruction** loss (Gelada et al., 2019; Hafner et al., 2019; 2020). Other methods that do not explicitly learn a world model involve learning representations using reconstruction based approaches like autoencoders (Yarats et al., 2019).

Quite a few papers in the past have analysed different sub-topics in RL through large scale studies. Engstrom et al. (2019) and Andrychowicz et al. (2020) have focused on analysing different policy optimization methods with varying hyperparameters. Our focus is specifically on representation learning methods that improve sample efficiency in pixel-based environments. Henderson et al. (2018) showed how RL methods in general can be susceptible to lucky seedings. Recently, Agarwal et al. (2021b) proposed statistical metrics for reliable evaluation. Despite having similar structure, our work is largely complimentary to these past investigations. Babaeizadeh et al. (2020) analysed reward and transition but only focused on the Atari 200M benchmark and pixel reconstruction methods. In comparison, our work is spread across multiple evaluation benchmarks, and our results show that reconstruction can be a fine technique only in a particular benchmark category.

3 Method

We model the RL problem using the framework of contextual decision processes (CDPs), a term introduced in Krishnamurthy et al. (2016) to broadly refer to any sequential decision making task where an agent must act on the basis of rich observations (context) x_t to optimize long-term reward. The true state of the environment s_t is not available and the agent must construct it on its own, which is required for acting optimally on the downstream task. Furthermore, the emission function which dictates what contexts are observed for a given state is assumed to only inject noise that is uncorrelated to the task in hand, i.e. it only changes parts of the context that are irrelevant to the task (Zhang et al., 2020b; Stone et al., 2021). Consider again the example of people walking on the sides of a road while a self-driving car changes lanes. Invariance to parts of the context that have no dependence on the task, e.g. people in the background, is

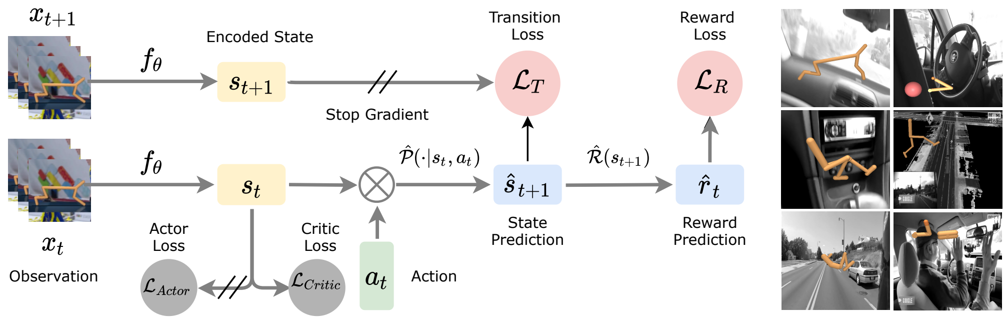

Figure 2: **(Left) Baseline for control over pixels**. We employ two losses besides the standard actor and critic losses, one being a reward prediction loss and the other a latent transition prediction loss. The encoded state s_t is the learnt representation. Gradients from both the transition/reward prediction and the critic are used to learn the representation, whereas the actor gradients are stopped. In the ALE setting, the actor and critic losses are replaced by a Rainbow DQN loss (van Hasselt et al., 2019). **(Right) Natural Distractor** in the background for standard DMC setting (left column) and custom off-center setting (right column). More details about the distractors can be found in Appendix 2.

an important property for any representation learning algorithm since we cannot expect all situations to remain exactly the same when learning in the real world. A more detailed description of the setup and all the prior methods used is provided in Appendix 1.

We start by exploring the utility of two fundamental components in RL, that of reward and transition prediction, in learning representations. A lot of prior work has incorporated these objectives either individually or in the presence of more nuanced architectures. Here, our aim is to start with the most basic components and establish their importance one by one. Particularly, we use a simple soft actor-critic setup with an embedding function $f_\theta : \mathcal{X} \rightarrow \mathcal{S}$ (similar to SAC-AE (Yarats et al., 2019)) as the base architecture, and attach the reward and transition prediction modules to it (See Figure 2). We define the transition prediction by $\hat{\mathcal{P}}(s_t, a_t)$ and the reward prediction by $\hat{\mathcal{R}}(s_{t+1})$ such that, $\hat{s}_{t+1} = \hat{\mathcal{P}}(s_t, a_t)$ and $\hat{r}_t = \hat{\mathcal{R}}(s_{t+1})$. Note that the transition network is over the encoded state $\hat{s}_t = f_\theta(x_t)$ and not over the observations x_t (Lee et al., 2020a). The overall auxiliary loss function is thus defined as follows:

$$\mathcal{L}_{Baseline} = \underbrace{\left(s_{t+1} - \hat{\mathcal{P}}(s_t, a_t)\right)^2}_{\text{Transition prediction loss}} + \underbrace{\left(\mathcal{R}(s_{t+1}) - \hat{\mathcal{R}}(\hat{\mathcal{P}}(s_t, a_t))\right)^2}_{\text{Reward prediction loss}} \tag{1}$$

Unless noted otherwise, we call this architecture as the *baseline* for all our experiments. Details about the implementation, network sizes and all hyperparameters is provided in Appendix 3 and Appendix 4 (Table 3) respectively.

Almost all methods considered in this paper can be boiled down to using two key ideas: that of reward and transition prediction. These can be instantiated in various ways, hence giving rise to differently performing methods. We therefore start with the simplest possible instantiation of these ideas, which corresponds to predicting the reward and the next encoded state.

Throughout the paper, we broadly categorize approaches into the following (based on how they were initially motivated):

- baseline (that simply predicts reward and next latent state),
- contrastive-based,
- metric-based,
- non-contrastive, self supervised based
- reconstruction-based

] Depending on the environment under consideration, each method category can be instantiated into a particular algorithm. For example, contrastive learning applied to MuJoCo environments is represented by DrQ. Note that even though we might have the same method category being applied to all possible environments, it is usually not possible to apply the same algorithm across all environments. This distinction is most apparent when switching from Atari to MuJoCo. We therefore discuss the reward and transition baselines only on the MuJoCo domains for simplicity and then describe the Atari results in a separate section to avoid confusion.

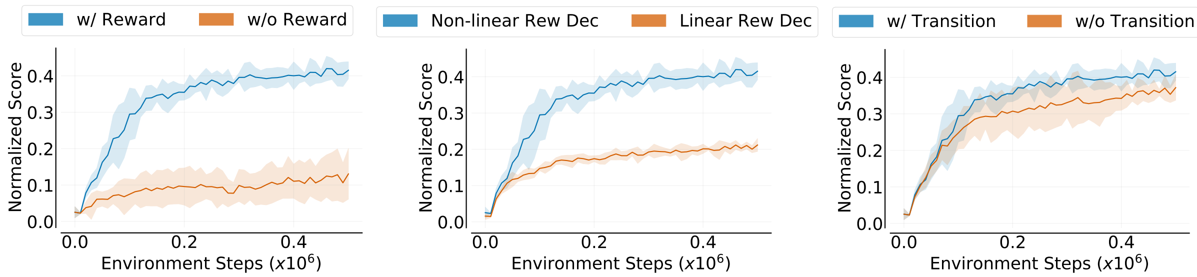

Figure 3: **Baseline Ablations**. Average normalized performance across six standard domains from DMC. Mean and std err. for 5 runs. **Left plot**: Baseline with *vs* without reward prediction **Middle plot**: Baseline with non-linear *vs* linear reward predictor/decoder. **Right plot**: Baseline with *vs* without transition prediction.

4 Empirical Study

In this section, we analyze the baseline architecture across six DMC tasks: Cartpole Swingup, Cheetah Run, Finger Spin, Hopper Hop, Reacher Easy, and Walker Walk. A common observation in our experiments is that the baseline is able to reduce the gap to more sophisticated methods significantly, sometimes even outperforming them in certain cases. This highlights that the baseline might serve as a stepping stone for other methods to build over. We test the importance of having both the reward and transition modules individually, by removing each of them one by one.

4.1 Reward Prediction

Figure 3 (left) shows a comparison of 'with *vs* without reward prediction'. All other settings are kept unchanged and the only difference is the reward prediction. When the reward model is removed, there remains no grounding objective for the transition model. This results in a representation collapse as the transition model loss is minimized by the trivial representation which maps all observations to the same encoded state leading to degraded performance. This hints at the fact that without a valid grounding objective (in this case from predicting rewards), learning good representations can be very hard. Note that it is not the case that there is no reward information available to the agent, since learning the critic does provide enough signal to learn efficiently when there are no distractions present. However, in the presence of distractions the signal from the critic can be extremely noisy since it is based on the current value functions, which are not well developed in the initial stages of training. One potential fix for such a collapse is to not use the standard maximum likelihood based approaches for the transition model loss and instead use a contrastive version of the loss, which has been shown to learn general representations in the self-supervised learning setting. We test this later in the paper and observe that although it does help prevent collapse, the performance is still heavily inferior to when we include the reward model. Complete performances for individual tasks are shown in Appendix 8.1.

Linear Reward Predictor. We also compare to the case when the reward decoder is a linear network instead of the standard 1 layer MLP. We see that performance decreases significantly in this case as shown in Figure 3 (middle), but still does not collapse like in the absence of reward prediction. We hypothesize that the reward model is potentially removing useful information for predicting the optimal actions. Therefore, when it is attached directly to the encoded state, i.e., in the linear reward predictor case, it might force the representation to only preserve information required to predict the reward well, which might not always be enough to predict the optimal actions well. For instance, consider a robot locomotion task. The reward in this case only depends on one variable, the center of mass, and thus the representation module would only need to preserve that in order to predict the reward well. However, to predict optimal actions, information about all the joint angular positions and velocities is required, which might be discarded if the reward model is directly attached to the encoded state. This idea is similar to why contrastive learning objectives in the self-supervised learning setting always enforce consistency between two positive/negative pairs *after* projecting the representation to another space. It has been shown that enforcing consistency in the representation space can remove excess information, which hampers final performance (Chen et al., 2020). We indeed see a similar trend in the RL case as well.

4.2 Transition Prediction

Similarly, Figure 3 (right) shows a comparison of 'with *vs* without transition prediction'. The transition model loss enforces temporal consistencies among the encoded states. When this module is removed, we observe a slight dip in

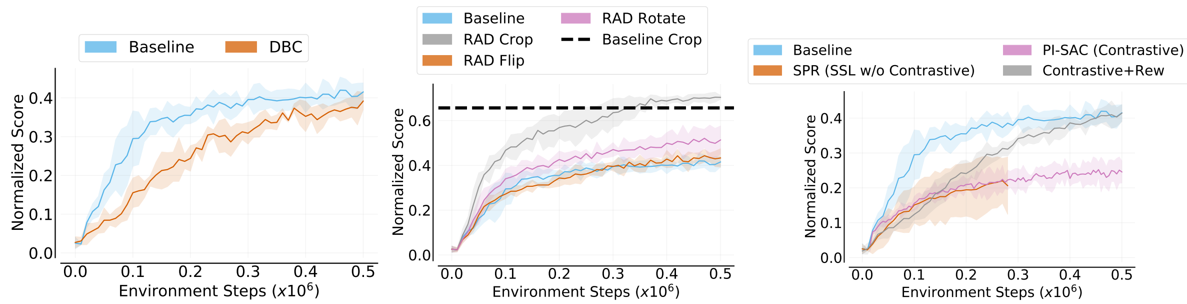

Figure 4: **Baseline Ablations**. Average normalized performance across six standard domains from DMC. Mean and std err. for 5 runs. **Left plot**: Baseline *vs* state metric losses (DBC (Zhang et al., 2020b)). The performance of baseline is compared with bisimulation metrics employed by DBC [3]. **Middle plot**: Data Augmentations. Cropping removes irrelevant segments while flip and rotate do not, performing similar to the baseline. Baseline with random crop performs equally as good as RAD. **Right plot**: Contrastive and SSL w/o Contrastive. We replace the transition loss of the baseline with a contrastive version (Contrastive + Rew). Further, we consider simple contrastive (PI-SAC (Lee et al., 2020b)) and SSL w/o contrastive (variant of SPR (Schwarzer et al., 2020) for DMC) losses as well.

performance across most tasks, with the most prominent drop in cartpole as shown in Appendix 8.1 (Figure 16). This suggests that enforcing such temporal consistencies in the representation space is indeed an important component for robust learning, but not a sufficient one. To examine if the marginal gain is an artifact of the exact architecture used, we explored other architectures in Appendix 8.2 but did not observe any difference in performance.

4.3 Connections to Value-Aware Learning

The baseline introduced above also resembles a prominent idea in theory, that of learning value aware models (Farahmand et al., 2017; Ayoub et al., 2020). Value-aware learning advocates for learning a model by fitting it to the value function of the task in hand, instead of fitting it to the true model of the world. The above baseline can be looked at as doing value aware learning in the following sense: the grounding to the representation is provided by the reward function, thus defining the components responsible for the task in hand and then the transition dynamics are learnt only for these components and not for all components in the observation space. There remains one crucial difference though. Value aware methods learn the dynamics based on the value function (multi-step) and not the reward function (1-step), since the value function captures the long term nature of the task in hand. To that end, we also test a more exact variant of the value-aware setup where we use the critic function as the target for optimizing the transition prediction, both with and without a reward prediction module (Table 1). Complete performances are provided in Appendix 8.8. We see that the value aware losses perform worse than the baseline. A potential reason for this could be that since the value estimates are noisy when using distractors, directly using these as targets inhibits learning a stable latent state representation. Indeed, more sophisticated value-aware methods such as in Temporal Predictive Coding (Nguyen et al., 2021) lead to similar scores as the baseline.

Table 1: **Truly value-aware objectives**. We report average final score after 500K steps across six standard domains from DMC.

	Baseline	Value-aware (w/ reward)	Value-aware (w/o reward)
Average Scores	0.42 ± 0.02	0.36 ± 0.03	0.23 ± 0.03

5 Comparison

So far, we have discussed why the two modules we identify as being vital for minimal and robust learning are actually necessary. Now we ask what other components could be added to this architecture which might improve performance, as has been done in prior methods. We then ask when do these added components actually improve performance, and when do they fail. More implementation details are provided in Appendix 3.

Metric Losses. Two recent works that are similar to the baseline above are DBC (Zhang et al., 2020b) and MiCO (Castro et al., 2021), both of which learn representations by obeying a distance metric. DBC learns the metric by estimating the reward and transition models while MiCO uses transition samples to directly compute the metric distance. We compare baseline's performance with DBC as shown in Figure 4 (left). Note that without the metric loss, DBC is similar to the baseline barring architectural differences such as the use of probabilistic transition models in DBC

[3]DBC (Zhang et al., 2020b) performance data is taken from their publication.

compared to deterministic models in the baseline. Surprisingly, we observe that the performance of the baseline exceeds that of DBC. To double check, we ran a version of DBC without the metric loss. Again, the "without metric" version lead to superior performance than the "with metric" one (DBC).

Data Augmentations. A separate line of work has shown strong results when using data augmentations over the observation samples. These include the RAD (Laskin et al., 2020a) and DrQ (Kostrikov et al., 2020) algorithms, both of which differ very minimally in their implementations. We run experiments for three different augmentations—'crop', 'flip', and 'rotate'. The 'crop' augmentation always crops the image by some shifted margin from the center. Interestingly, the image of the agent is also always centered, thus allowing 'crop' to always only remove background or task-irrelevant information and never remove the agent or task-relevant information. This essentially amounts to not having background distractors and thus we see that this technique performs quite well as shown in Figure 4 (middle). However, augmentations that do not explicitly remove the distractors, such as rotate and flip, lead to similar performance as the baseline. This suggests that augmentations might not be helpful when distractor information cannot be removed, or when we do not know where the objects of interest lie in the image, something true of the real world. We test this by shifting the agent to the side, thus making the task-relevant components off-center and by zooming out i.e. increasing the amount of irrelevant information even after cropping. We see that performance of 'crop' drops drastically in this case, showcasing that most of the performance gains from augmentations can be attributed to how the data is collected and not to the algorithm itself. Additional ablations are provided in Appendix 8.3.

Note that DrQ uses a different cropping scheme than RAD, one which preserves all irrelevant information. This highlights that invariance to irrelevant information might not be the only reason for the success of augmentations. Two other potential reasons are: 1) cropping leads to explicitly modeling translational invariance in the network, and 2) how the DrQ augmentation affects Q estimation (both in the current and target networks). These two can act as potential confounders in understanding the benefits of augmentation accurately.

Table 2: **RAD additional ablations**. We report average final score after 500K steps across Cheetah Run and Walker Walk domains from DMC. This illustrates that the performance of augmentations is susceptible to quality of data. Also, for the "Off-center + Zoomed Out" setting, it is worth noting that performance of 'Baseline' without augmentations is more robust to changes as compared to augmentations.

	Baseline	Baseline Crop	Baseline Flip	RAD Crop	RAD Flip
Standard	0.44 ± 0.04	0.89 ± 0.02	0.75 ± 0.06	0.93 ± 0.02	0.49 ± 0.06
Off-center + Zoomed Out	0.56 ± 0.01	0.72 ± 0.06	0.53 ± 0.06	0.70 ± 0.05	0.48 ± 0.04

Contrastive and SSL w/o Contrastive Losses. A lot of recent methods also deploy contrastive losses (for example, CPC Oord et al. (2018)) to learn representations, which essentially refers to computing positive/negative pairs and pushing together/pulling apart representations respectively. In practice, this can be done for any kind of loss function, such as the encoding function f_θ (Hafner et al., 2020), or using random augmentations (Laskin et al., 2020b; Lee et al., 2020b), so on and so forth. Therefore, we test a simple modification to the baseline, that of using the contrastive variant of the transition prediction loss than the maximum likelihood version. Specifically, the encoded current state s_t and the encoded next state s_{t+1} form a positive pair while any other random state in the batch forms a negative pair with s_t. Such a loss can be written down in terms of the standard InfoNCE Oord et al. (2018) objective, as in Eq. 2.

$$\mathcal{L}_{Contrastive} = \text{InfoNCE}\big(f(x_t),\ f(x_{t+1})\big), \tag{2}$$

where $f(x_t)$ includes a backbone network (typically a ResNet) and a projection network that maps the output of the backbone network to a much smaller space. We see, in Figure 4 (right), that the contrastive version leads to inferior results than the baseline, potentially suggesting that contrastive learning might not add a lot of performance improvement, particularly when there is grounding available from supervised losses. A similar trend has been witnessed in the self-supervised literature with methods like SimSiam (Chen & He, 2021), Barlow Twins (Zbontar et al., 2021), and BYOL (Grill et al., 2020) getting similar or better performance than contrastive methods like SimCLR (Chen et al., 2020). Complete performances are provided in Appendix 8.5.

SPR (Schwarzer et al., 2020) is known to be a prominent algorithm in the ALE domain, leading to the best results overall. SPR deploys a specific similarity loss for transition prediction motivated by BYOL (Grill et al., 2020). We follow the same setup and test a variant of the baseline which uses the cosine similarity loss from SPR and test its performance on DMC based tasks. A general description of such kind of loss functions is provided in Eq. 3 while a more specific version is provided in the Appendix 1. We again show in Figure 4 (right) that there is very little or no improvement in performance as compared to the baseline performance. This again suggests that the same algorithmic idea can have an entirely different performance just by changing the evaluation setting[4] (ALE to DMC).

$$\mathcal{L}_{w/o-Contrastive} = -\frac{p_t}{\|p_t\|^2} \cdot \frac{p_{t+1}}{\|p_{t+1}\|^2} \tag{3}$$

$p_t = h(f(x_t))$ is obtained by passing the encoder output through a prediction network, denoted by h.

Learning World Models. We test DREAMER (Hafner et al., 2020), a state of the art model-based method that learns world models though pixel reconstruction on two settings, with and without distractors. Although the performance in the "without distractors" case is strong, we see that with distractors, DREAMER fails on some tasks, while performing inferior to the baseline in most tasks (see Figure 5).

This suggests that learning world models through reconstruction might only be a good idea when the world models are fairly simple to learn. If world models are hard to learn, as is the case with distractors, reconstruction based learning can lead to severe divergence that results in no learning at all. We also compare against the more recently introduced method from Fu et al. (2021). Their method, called TIA(Fu et al., 2021) incorporates several other modules in addition to DREAMER and learns a decoupling between the distractor background and task relevant components. We illustrate the performance of each of the above algorithms in Figure 5 along with a version where we add full reconstruction loss to the baseline. Interestingly, TIA still fails to be superior to the baseline, particularly for simpler domains like Cartpole. Complete performances are provided in Appendix 8.6

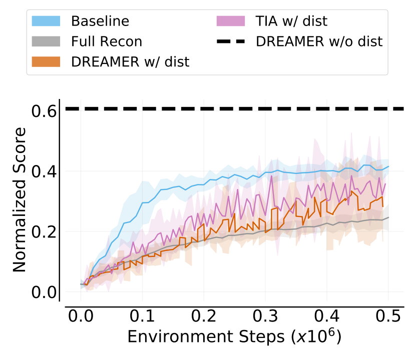

Figure 5: **Pixel reconstruction**. Average normalized performance across six DMC domains with distractors. Baseline achieves better performance than SOTA methods like DREAMER and TIA [5].

Relevant Reconstruction and Sparse Rewards. Since thus far we only considered dense reward based tasks, using the reward model for grounding is sufficient to learn good representations. More sophisticated auxiliary tasks considered in past works include prediction of ensemble of value networks, prediction of past value functions, prediction of value functions of random cumulants, and observation reconstruction. However, in the sparse reward case, grounding on only the reward model or on past value functions can lead to representation collapse if the agent continues to receive zero reward for a long period of time. Therefore, in such cases where good exploration is necessary, tasks such as observation reconstruction can help prevent collapse. Although this has been shown to be an effective technique in the past, we argue that full reconstruction can still harm the representations in the presence of distractors. Instead, we claim that reconstruction of *only* the task relevant components in the observation space results in learning good representations (Fu et al., 2021), especially when concerned with realistic settings like that of distractors. We conduct a simple experiment to show that in the sparse reward case, task-relevant reconstruction [6] is sufficient for robust performance. We show this in Figure 6 along with performance of baseline and augmentations. Of course, how one should come up with techniques that differentiate between task-relevant and task-irrelevant components in the observations, remains an open question [7]. Additional ablations are provided in Appendix 8.6.

Atari 100K. We study the effect of techniques discussed thus far for the Atari 100K benchmark, which involves 26 Atari games and compares performance relative to human-achieved scores at 100K steps or 400K frames. We consider the categorization proposed by Bellemare et al. (2016) based on the nature of reward (dense, human optimal, score exploit and sparse) and implement two versions of the baseline algorithm, one with both the transition and reward prediction modules and the other with only reward prediction. Our average results over all games show that the baseline performs comparably to CURL (Laskin et al., 2020b), SimPLe (Kaiser et al., 2019), DER (van Hasselt et al., 2019),

[4]The SPR version without augmentations actually uses two separate ideas for improvement in performance, a cosine similarity transition prediction loss and a separate convolution encoder for the transition network, making it hard to attribute gains over the base DER (van Hasselt et al., 2019) to just transition loss.

[4]TIA (Fu et al., 2021) performance data is taken from their publication.

[6]*Part Recons.* in Figure 6 amounts to reconstructing the DMC agent over a solid black background.

[7]As also evident by TIA's (Fu et al., 2021) performance for DMC ball-in-cup catch experiments.

and OTR (Kielak, 2020) while being quite inferior to DrQ[8] (Kostrikov et al., 2020; Agarwal et al., 2021b) and SPR (Schwarzer et al., 2020). Since our implementation of the baseline is over the DER code, similar performance to DER might suggest that the reward and transition prediction do not help much in this benchmark. Note that ALE does not involve the use of distractors and so learning directly from the RL head (DQN in this case) should be enough to encode information about the reward and the transition dynamics in the representation. This comes as a stark contrast to the without distractors case in DMC Suite, where transition and reward prediction still lead to better performance. Such differences can also be attributed to the continuous *vs* discrete nature of DMC and ALE benchmarks. More interestingly, we find that when plotting the average performance for only the dense reward environments, the gap in performance between DER and SPR/DrQ decreases drastically. Note that SPR builds over DER but DrQ builds over OTR.

We further delve into understanding the superior performance of SPR and DrQ. In particular, SPR combines a cosine similarity transition prediction loss with data augmentations. To understand the effect of each of these individually, we run SPR without data augmentations, referring to this version by SPR**[9]. We see that SPR** leads to performance similar to the baseline and the DER agent, suggesting that such a self-supervised loss may not lead to gains when run without data augmentations. Finally, we take the DER agent and add data augmentations to it (from DrQ). This is shown as DER + Aug in Figure 7. We see that this leads to collapse, with the worst performance across all algorithms. Note that DrQ builds over OTR and performs quite well whereas when the same augmentations are used with DER, which includes a distributional agent in it, we observe a collapse. This again indicates that augmentations can change data in a fragile manner, sometimes leading to enhanced performance with certain algorithms, while failing with other algorithms. Segregating evaluation of algorithms based on these differences is therefore of utmost importance. We show the individual performance on all 25 games in Appendix 8.5 (Table 7).

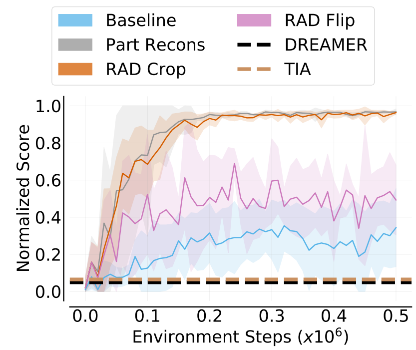

Figure 6: **Reconstruction and augmentations for sparse settings**. Normalized performance for ball-in-cup catch domain from DMC.

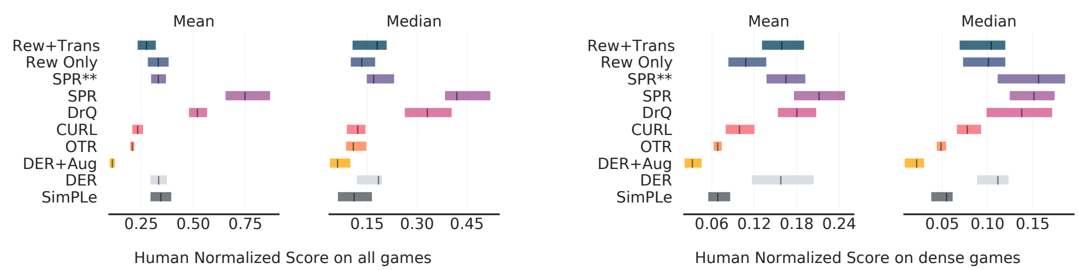

Figure 7: **Atari 100K**. Human normalized performance (mean/median) across 25 games from the Atari 100K benchmark. Mean and 95% confidence interval for 5 runs. **Left plot**: Comparison for all 25 games. **Right plot**: Comparison for only dense reward games (7 games from Table 7).

6 Discussion

The above description of results on DMC Suite and Atari 100K point to a very interesting observation, that evaluation of different algorithms is very much correlated with a finer categorization of the evaluation benchmark, and not the whole benchmark itself. Specifically, focusing on finer categorizations such as density of reward, inherent horizon of the problem, presence of irrelevant and relevant task components, discreteness *vs* continuity of actions etc. is vital in recognizing if certain algorithms are indeed *better* than others. Figure 1 stands as a clear example of such discrepancies. These observations pave the way for a better evaluation protocol for algorithms, one where we rank algorithms for different categories, each governed by a specific data-centric property of the evaluation benchmark. Instead of saying that algorithm X is better than algorithm Y in benchmark Z, our results advocate for an evaluation methodology which claims algorithm X to be better than algorithm Y in dense reward, short horizon problems (considered from benchmark

[8]We use the DrQ(ϵ) version from Agarwal et al. (2021b) for fair evaluation and denote it as DrQ.

[9]Note that this is different from the SPR without augmentations version reported in Schwarzer et al. (2020) since that version uses dropout as well which is not a fair comparison.

Z), i.e. enforcing less emphasis on the benchmark itself and more on certain properties of a subset of the benchmark. Having started with the question of what matters when learning representations over pixels, our experiments and discussion clearly show that largely it is the data-centric properties of the evaluation problems that matter the most.

7 Conclusion

In this paper we explore what components in representation learning methods matter the most for robust performance. As a starting point, we focused on the DMC Suite with distractors and the Atari 100k benchmark. Our results show that a simple baseline, one involving a reward and transition prediction modules can be attributed to a lot of performance benefits in DMC Suite with distractors. We then analysed why and when existing methods fail to perform as good or better than the baseline, also touching on similar observations on the ALE simulator. Some of our most interesting findings are as follows:

- Pixel reconstruction is a sound technique in the absence of clutter in the pixels, but suffers massively when distractors are added. In particular, DREAMER and adding a simple pixel reconstruction loss leads to worse performance than the baseline in DMC Suite (Figure 5).
- Contrastive losses in and of itself do not seem to provide gains when there is a supervised loss available in place of it. We observe that replacing the supervised state prediction loss of the baseline by the InfoNCE contrastive loss does not lead to performance improvements over the baseline in DMC Suite (Figure 4 right plot). On the other hand, using contrastive losses with data augmentations can lead to more robust improvements (Lee et al., 2020b; Fan & Li, 2021).
- Certain augmentations ('crop') do well when data is centered while dropping in performance when data is off-center or when cropping fails to remove considerable amounts of task-irrelevant information. Other augmentations ('flip' and 'rotate') show the opposite behavior (RAD ablations on DMC Suite in Table 2).
- SSL w/o contrastive losses does not provide much gains when used alone. With data augmentations, they lead to more significant gains. For Atari100k, Figure 7 shows that SPR, a state of the art non contrastive method leads to similar performance as the base DER agent when used without data augmentations (denoted by SPR**). Using the SPR inspired loss in DMC Suite also did not lead to gains over the baseline (in Figure 4 right plot).
- Augmentations are susceptible to collapse in the presence of distributional Q networks. Figure 7 shows that 'crop' and 'intensity' augmentations added to the DER agent lead to a complete failure in performance in Atari100k.

These results elicit the observation that claiming dominance over other methods for an entire benchmark may not be an informative evaluation methodology. Instead, focusing the discussion to a more data-centric view, one where specific properties of the environment are considered, forms the basis of a much more informative evaluation methodology. We argue that as datasets become larger and more diverse, the need for such an evaluation protocol would become more critical. We hope this work can provide valuable insights in developing better representation learning algorithms and spur further discussion in categorizing evaluation domains in more complex scenarios, such as with real world datasets and over a wider class of algorithmic approaches.

References

Agarwal, R., Machado, M. C., Castro, P. S., and Bellemare, M. G. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021a.

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., and Bellemare, M. G. Deep reinforcement learning at the edge of the statistical precipice. *arXiv preprint arXiv:2108.13264*, 2021b.

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pp. 463–474. PMLR, 2020.

Babaeizadeh, M., Saffar, M. T., Hafner, D., Erhan, D., Kannan, H., Finn, C., and Levine, S. On trade-offs of image prediction in visual model-based reinforcement learning. 2020.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29:1471–1479, 2016.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Biza, O., Platt, R., van de Meent, J.-W., and Wong, L. L. Learning discrete state abstractions with deep variational inference. *arXiv preprint arXiv:2003.04300*, 2020.

Castro, P. S. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10069–10076, 2020.

Castro, P. S., Kastner, T., Panangaden, P., and Rowland, M. Mico: Learning improved representations via sampling-based state similarity for markov decision processes. *CoRR*, abs/2106.08229, 2021. URL https://arxiv.org/abs/2106.08229.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Co-Reyes, J. D., Sanjeev, S., Berseth, G., Gupta, A., and Levine, S. Ecological reinforcement learning. *arXiv preprint arXiv:2006.12478*, 2020.

Du, S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudik, M., and Langford, J. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pp. 1665–1674. PMLR, 2019.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep rl: A case study on ppo and trpo. In *International conference on learning representations*, 2019.

Fan, J. and Li, W. Robust deep reinforcement learning via multi-view information bottleneck. *arXiv preprint arXiv:2102.13268*, 2021.

Farahmand, A.-m., Barreto, A., and Nikovski, D. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pp. 1486–1494. PMLR, 2017.

Ferns, N., Panangaden, P., and Precup, D. Bisimulation metrics for continuous markov decision processes. *SIAM J. Comput.*, 40(6):1662–1714, December 2011. ISSN 0097-5397. doi: 10.1137/10080484X. URL https://doi.org/10.1137/10080484X.

Fu, X., Yang, G., Agrawal, P., and Jaakkola, T. Learning task informed abstractions. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3480–3491. PMLR, 18–24 Jul 2021. URL http://proceedings.mlr.press/v139/fu21b.html.

Gelada, C., Kumar, S., Buckman, J., Nachum, O., and Bellemare, M. G. DeepMDP: Learning continuous latent space models for representation learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2170–2179. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/gelada19a.html.

Grigsby, J. and Qi, Y. Measuring visual generalization in continuous control from pixels. *arXiv preprint arXiv:2010.06740*, 2020.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565. PMLR, 2019.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Kielak, K. Importance of using appropriate baselines for evaluation of data-efficiency in deep reinforcement learning for atari. *arXiv preprint arXiv:2003.10181*, 2020.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *CoRR*, abs/2004.13649, 2020. URL https://arxiv.org/abs/2004.13649.

Krishnamurthy, A., Agarwal, A., and Langford, J. Pac reinforcement learning with rich observations. *arXiv preprint arXiv:1602.02722*, 2016.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19884–19895. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/file/e615c82aba461681ade82da2da38004a-Paper.pdf.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. *Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119*, 2020b. arXiv:2004.04136.

Lee, A. X., Nagabandi, A., Abbeel, P., and Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 741–752. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/file/08058bf500242562c0d031ff830ad094-Paper.pdf.

Lee, K.-H., Fischer, I., Liu, A., Guo, Y., Lee, H., Canny, J., and Guadarrama, S. Predictive information accelerates learning in rl. *arXiv preprint arXiv:2007.12401*, 2020b.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Nguyen, T., Shu, R., Pham, T., Bui, H., and Ermon, S. Temporal predictive coding for model-based planning in latent space. *arXiv preprint arXiv:2106.07156*, 2021.

Oh, J., Guo, X., Lee, H., Lewis, R., and Singh, S. Action-conditional video prediction using deep networks in atari games. *arXiv preprint arXiv:1507.08750*, 2015.

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

Raileanu, R., Goldstein, M., Yarats, D., Kostrikov, I., and Fergus, R. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*, 2020.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *International conference on machine learning*, pp. 1312–1320. PMLR, 2015.

Schmidhuber, J. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Stone, A., Ramirez, O., Konolige, K., and Jonschkowski, R. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pp. 9870–9879. PMLR, 2021.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 761–768, 2011.

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

Tomar, M., Zhang, A., Calandra, R., Taylor, M. E., and Pineau, J. Model-invariant state abstractions for model-based reinforcement learning. *arXiv preprint arXiv:2102.09850*, 2021.

van Hasselt, H. P., Hessel, M., and Aslanides, J. When to use parametric models in reinforcement learning? In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/1b742ae215adf18b75449c6e272fd92d-Paper.pdf.

Wahlström, N., Schön, T. B., and Deisenroth, M. P. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.

Watter, M., Springenberg, J. T., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. *arXiv preprint arXiv:1506.07365*, 2015.

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Reinforcement learning with prototypical representations. *arXiv preprint arXiv:2102.11271*, 2021a.

Yarats, D., Kostrikov, I., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Zhang, A., Wu, Y., and Pineau, J. Natural environment benchmarks for reinforcement learning. *arXiv preprint arXiv:1811.06032*, 2018.

Zhang, A., Lyle, C., Sodhani, S., Filos, A., Kwiatkowska, M., Pineau, J., Gal, Y., and Precup, D. Invariant causal prediction for block MDPs. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11214–11224. PMLR, 13–18 Jul 2020a. URL https://proceedings.mlr.press/v119/zhang20t.html.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning Invariant Representations for Reinforcement Learning without Reconstruction. *arXiv*, jun 2020b. ISSN 23318422. URL http://arxiv.org/abs/2006.10742.

Zhang, A., Sodhani, S., Khetarpal, K., and Pineau, J. Learning Robust State Abstractions for Hidden-Parameter Block MDPs. pp. 1–22, 2020c. URL http://arxiv.org/abs/2007.07206.