Representation Learning for Out-of-distribution Generalization in Reinforcement Learning

Frederik Träuble [*1] Andrea Dittadi [*2] Manuel Wüthrich [1] Felix Widmaier [1] Peter Gehler [3] Ole Winther [2]
Francesco Locatello [3] Olivier Bachem [4] Bernhard Schölkopf [1] Stefan Bauer [1 5]

Abstract

Learning data representations that are useful for various downstream tasks is a cornerstone of artificial intelligence. While existing methods are typically evaluated on downstream tasks such as classification or generative image quality, we propose to assess representations through their usefulness in downstream control tasks, such as reaching or pushing objects. By training over 10,000 reinforcement learning policies, we extensively evaluate to what extent different representation properties affect out-of-distribution (OOD) generalization. Finally, we demonstrate zero-shot transfer of these policies from simulation to the real world, without any domain randomization or fine-tuning. This paper aims to establish the first systematic characterization of the usefulness of learned representations for real-world OOD downstream tasks.

1 Introduction

Robust out-of-distribution (OOD) generalization is one of the key open challenges in machine learning. This is particularly relevant for the deployment of ML models to the real world, where we need systems that generalize well beyond the i.i.d. (independent and identically distributed) data setting (Schölkopf et al., 2021; Djolonga et al., 2020; Koh et al., 2020; Barbu et al., 2019; Azulay and Weiss, 2019; Roy et al., 2018; Gulrajani and Lopez-Paz, 2020; Hendrycks and Dietterich, 2019; Michaelis et al., 2019). One instance of such models are agents that learn by interacting with a training environment but cannot generalize and transfer their learned skill to other environments with different statistics (Zhang et al., 2018; Cobbe et al., 2019; Ahmed et al., 2021).

[*]Equal contribution [1]MPI for Intelligent Systems [2]Technical University of Denmark [3]Amazon [4]Google Brain [5]CIFAR Azrieli Global Scholar. Correspondence to: Frederik Träuble <frederik.traeuble@tuebingen.mpg.de>, Andrea Dittadi <adit@dtu.dk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Consider the example of a robot with the task of moving a cube to a target position: Such an agent can easily fail as soon as some aspects of the environment differ with respect to the training setup, e.g. the shape, color, and other object properties, or when transferring from simulation to real world. In particular, some of the main issues in deep reinforcement learning are data inefficiency, brittleness with respect to changes in the input data distribution, and poor interpretability (Garnelo et al., 2016; Lake et al., 2017; Kaiser et al., 2019; Li, 2018; Zambaldi et al., 2019; Lyu et al., 2019; Zhang et al., 2019; Heuillet et al., 2021).

Humans seem to not suffer from these pitfalls when transferring learned skills beyond a narrow training domain. In fact, one of the fundamental cognitive capabilities in humans is to represent visual sensory data in a useful and concise manner (Marr, 1982; Gordon and Irwin, 1996; Lake et al., 2017; Anand et al., 2019; Spelke, 1990). Therefore, a particularly promising path is to base decisions and predictions on such structured and meaningful lower-dimensional representations of our world (Bengio et al., 2013). The learned representation should facilitate efficient downstream learning (Eslami et al., 2018; Anand et al., 2019) and exhibit better generalization (Zhang et al., 2020; Srinivas et al., 2020).

While previous work shows that good internal representations of raw observations are important for domain adaptation (Littman et al., 2001; Pan and Yang, 2009; Finn et al., 2016a; Barreto et al., 2017), so far representations are typically evaluated on downstream tasks such as classification or generative image quality which often serve as proxies for intended use cases. To move closer to realistic settings, we present a large-scale study (with 11,520 trained policies) investigating the relevance of learning representations for real-world reinforcement learning and OOD generalization. This study is based on the practically relevant setting of robotics and empirically analyzes key principles for representations and downstream policies in simulation and real world. See Fig. 1 for an overview of the setup.

We summarize our contributions as follows:

- We conduct a large-scale study training 11,520 policies and empirically investigate the role of pre-trained representations in reinforcement learning tasks from camera input.[1]

- We extensively evaluate the out-of-distribution generalization of these policies to unseen environments and systematically investigate its dependence on the pre-trained representations.
- We deploy policies to the corresponding real-world robotic platform, observe zero-shot sim2real transfer without fine-tuning or domain randomization, and investigate the role of representations in this setting.

2 Background

In this section, we provide relevant background on the methods for representation learning, reinforcement learning, and evaluation of out-of-distribution generalization.

Variational autoencoders. VAEs (Kingma and Welling, 2014; Rezende et al., 2014) are a framework for optimizing a latent variable model $p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ with parameters θ, typically with a fixed prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, using amortized stochastic variational inference. A variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ with parameters ϕ approximates the intractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$. The approximate posterior and generative model, typically called encoder and decoder and parameterized by neural networks, are jointly optimized by maximizing the ELBO (Evidence Lower BOund) which is a lower bound to the log likelihood:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}} (q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}))$$

In β-VAEs, the KL term is modulated by a factor β to enforce a more structured latent space (Higgins et al., 2017a; Burgess et al., 2018). While VAEs are typically trained without supervision, in this work we will also employ a form of weak supervision proposed by Locatello et al. (2020) to learn disentangled representations.

Deep reinforcement learning. The problem setting in Reinforcement Learning (RL) is modeled via a Partially Observable Markov Decision Process (POMDP) defined by the tuple $(S, A, T, R, \Omega, O, \gamma, \rho_0, H)$ with states $s \in S$, actions $a \in A$ and observations $o \in \Omega$ determined by the state and action of the environment $O(o|s, a)$. $T(s_{t+1}|s_t, a_t)$ is the transition probability distribution function, $R(s_t, a_t)$ is the reward function, γ is the discount factor, $\rho_0(s)$ is the initial state distribution at the beginning of each episode, and H is the time horizon per episode. The objective in RL is to learn a policy $\pi : S \times A \to [0, 1]$, typically parameterized by a neural network, that maximizes the total discounted expected reward $J(\pi) = \mathbb{E}\big[\sum_{t=0}^{H} \gamma^t R(s_t, a_t)\big]$. There is a broad range of proposed model-free learning algorithms to find π^* by policy gradient optimization or learning value functions while trading off exploration and exploitation (Haarnoja et al., 2018; Schulman et al., 2017; Sutton et al., 1999; Schulman et al., 2015a;b; Silver et al., 2014; Fujimoto et al., 2018). Here, we optimize the objective above with *Soft Actor Critic* (SAC), a widely used off-policy method in control tasks due to its sample efficiency (Haarnoja et al., 2018). SAC aims to improve sample-inefficiency and convergence in RL by simultaneously maximizing the expected reward and the entropy $H(\pi(\cdot|s_t))$.

Robotic setup and related dataset. Recently, Dittadi et al. (2021) introduced a dataset of over 1M simulated and real world images derived from the Trifinger robot platform introduced by Wüthrich et al. (2020) we will base our study on. The scene comprises a robot finger with three joints that can be controlled to manipulate a cube in a bowl-shaped stage. See Fig. 1 (step 1) for an example. The data was generated from 7 different factors of variation (FoV): angles of the upper, middle, and lower joints, and position (x and y), orientation, and color of the cube. This dataset corresponds to a robotic setup, so that learned representations can be used for control and reinforcement learning in simulation and in the real world. However, the focus of that work was to scale VAE-based learning approaches to this complex dataset and conduct a large-scale empirical study on generalization to various transfer scenarios, with a particular emphasis on disentanglement. For this reason, the role of representations in robotic control downstream tasks was not investigated.

Measuring out-of-distribution generalization. We closely follow the framework for measuring out-of-distribution (OOD) generalization proposed by Dittadi et al. (2021). First, representations are learned on a training set D. Then, we investigate OOD generalization by training downstream models on a subset $D_1 \subset D$ to predict ground truth factor values from the learned representations. These models are then tested on a set D_2 that differs distributionally from the training set D_1, e.g. containing images corresponding to held-out values of a chosen factor of variation (FoV). Dittadi et al. (2021) consider two flavors of OOD generalization depending on the choice of D_2: The case when $D_2 \subset D$, i.e. the OOD test set is a subset of the dataset for representation learning, is denoted by **OOD1**, while in **OOD2** D and D_2 are disjoint and distributionally different.

For example, consider the case in which distributional shifts are based on one FoV: the color of the cube in our robotic setup. Then, we could define these datasets such that images in D always contain a red or blue object, and those in $D_1 \subset D$ always contain a red object. In the OOD1 scenario, images in D_2 would always contain a blue object, whereas in the OOD2 case they would always contain an object that is neither red nor blue. Dittadi et al. (2021) consider as regression models Gradient Boosted Trees (GBT) and MLPs

[1]Training the VAEs required approximately 0.62 GPU years on NVIDIA Tesla V100. Training and evaluating the downstream policies required about 86.8 CPU years on Intel Platinum 8175M.

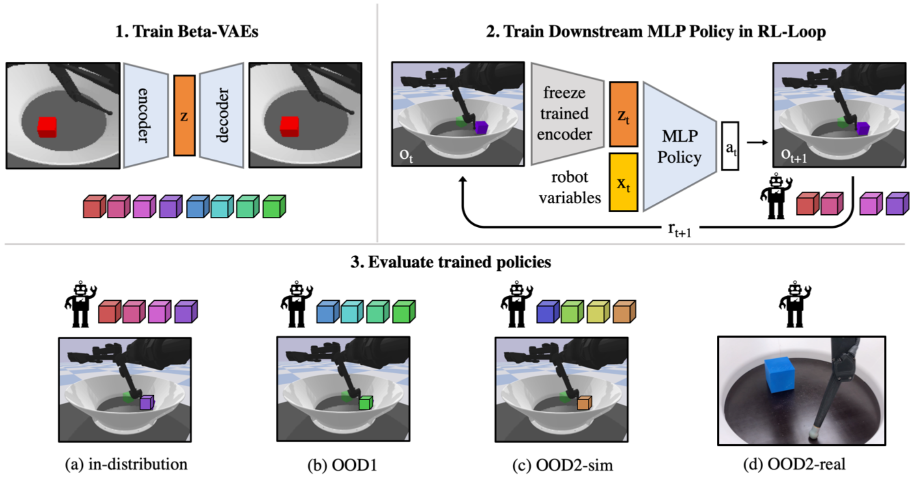

Figure 1. **Overview of our experimental setup for investigating out-of-distribution generalization in downstream tasks.** (1) We train 240 β-VAEs on the robotic dataset from Dittadi et al. (2021) . (2) We then train downstream policies to solve `ReachCube` or `Pushing`, using multiple random RL seeds per VAE. The input to a policy consists of the output of a pre-trained encoder and additional task-related observable variables. Crucially, the policy is only trained on a subset of the cube colors from the pre-training dataset. (3) Finally, we evaluate these policies on their respective tasks in four different scenarios: (a) in-distribution, i.e. with cube colors used in policy training; (b) OOD1, i.e. with cube colours previously seen by the encoder but OOD for the policy; (c) OOD2-sim, having cube colours also OOD to the encoder; (d) sim2real zero-shot on the real world setup.

with 2 hidden layers. Generalization in D_2 is measured by the (normalized) mean absolute prediction error across all FoVs except for the one that is OOD. In this work, we will use the *negative* error for interpretability, and decompose this metric on a per-factor level as well and refer to this generalization score as GS-OOD1, GS-OOD2-sim and GS-OOD2-real accordingly.

In contrast to Dittadi et al. (2021), who focus on introducing the dataset and evaluating generalization of simple factor prediction tasks, we leverage the broader potential of this robotic setup, which is to evaluate the usefulness of representations for more practically relevant robotic downstream tasks. Additionally, we cover more OOD2 scenarios both in simulation and in the real world, and investigate the relationship between a wide array of metrics and OOD generalization in RL tasks.

3 Related work

Lower-dimensional representations that can be flexibly used for a multitude of downstream tasks are considered an important component of any robust and generalizing machine learning system (Bengio et al., 2013; Schölkopf et al., 2021). In machine learning, these representations are typically learned from labels or rewards, which is often sample-inefficient. More sample-efficient alternatives that leverage the large amount of unstructured information in raw data include unsupervised (Kingma and Welling, 2014; Rezende et al., 2014; Dinh et al., 2016; Dumoulin et al., 2016) and self-supervised learning (Pathak et al., 2016; Doersch and Zisserman, 2017; Kolesnikov et al., 2019; Chen et al., 2020). In particular, disentangled representation learning aims at inferring the (causal) factors of the generative model of the data by enforcing sufficient structure on the latent space (Higgins et al., 2017a; Kim and Mnih, 2018; Burgess et al., 2018; Kumar et al., 2018; Chen et al., 2018; Locatello et al., 2019a; 2020).

Evaluating representations. In generative modeling, representations are typically evaluated by log likelihood, ELBO, or perceptual metrics such as FID (Heusel et al., 2017), IS (Salimans et al., 2016), or precision/recall (Sajjadi et al., 2018; Kynkäänniemi et al., 2019). Compression capability can also be evaluated e.g. in the context of bits-back coding, where it is formally related to the ELBO (Honkela and Valpola, 2004; Townsend et al., 2019; Kingma et al., 2019; Ruan et al., 2021). In general, representation quality has also been measured in terms of disentanglement (Higgins et al., 2017a; Kim and Mnih, 2018; Chen et al., 2018; Ridgeway and Mozer, 2018; Kumar et al., 2018; Eastwood and Williams, 2018), robustness (Suter et al., 2019), or the complexity of learning downstream predictors (Whitney et al., 2020). The evaluation framework in this paper is related to recent work that focuses on evaluating generalization in various practically relevant out-of-distributions settings (Gondal et al., 2019; Träuble et al., 2020; Dittadi et al., 2021). To the best of our knowledge, there is no rigorous and systematic study on the role of representations on downstream performance in robotic downstream tasks.

Learning representations for control. Learning low-dimensional representations that are capturing an environments' variations for RL agents in control scenarios is also often being described as state representation learning (Lesort et al., 2018): Methods therein are typically based on autoencoders (Watter et al., 2015; Ha and Schmidhuber, 2018; Higgins et al., 2017b; Van Hoof et al., 2016), video prediction (Oh et al., 2015; Finn et al., 2016b) or contrastive learning (Aytar et al., 2018; Sermanet et al., 2018; Anand et al., 2019). Learning policies from high-dimensional pixel observations is typically not very sample-efficient, a well-known problem of deep RL (Lake et al., 2017; Kaiser et al., 2019). It is thus becoming increasingly popular to leverage pre-trained representations and effectively decouple representation learning and policy learning in pixel-based environments (Eslami et al., 2018; Cuccu et al., 2018). For instance, Stooke et al. (2020) propose that decoupling representation learning from RL is more efficient than learning reward structure from pixels on their contrastive method. Similarly, CURL (Srinivas et al., 2020) investigates contrastive representation learning simultaneously with an off-policy RL algorithm, and match the sample-efficiency of policy learning from state-based features. Previous works highlight that for domain adaptation it is important to have good internal representations of raw observations (Littman et al., 2001; Pan and Yang, 2009; Finn et al., 2016a; Barreto et al., 2017). It is argued that these representations should be learned from the source domain only, because it might be difficult or expensive to obtain training data from the target domain (Finn et al., 2017; Rusu et al., 2017). Importantly, by simply training deep RL from scratch, the policies will often overfit to the source distribution (Rusu et al., 2017; Lake et al., 2017).

Closing the (sim2real) generalization gap in real-world RL. A key unsolved challenge in RL is that agents are very brittle to distribution shifts in their environment, even if the underlying structure is largely unchanged (Cobbe et al., 2019; Ahmed et al., 2021). DARLA (Higgins et al., 2017b) focuses on domain adaptation and zero-shot transfer for RL in DeepMind Lab and MuJoCo environments, and claim disentangled representations improve robustness. To obtain better transfer capabilities, Asadi et al. (2020) argue for discretizing the state space in continuous control domains by clustering together states where the optimal policy is similar. Transfer becomes especially challenging from the simulation to the real-world, a phenomenon often referred to as the sim2real gap. This is particularly crucial in RL, as real-world training is expensive, requires sample-efficient methods and is sometimes unfeasible if the reward structure requires accurate ground truth labels (Dulac-Arnold et al., 2019; Kormushev et al., 2013). Typically this issue is tackled by using large-scale domain randomization in simulation (Akkaya et al., 2019; James et al., 2019). Yan et al. (2020) propose using segmentation as a domain-invariant state representation.

4 Study design

Fig. 1 provides an overview of our setup. We study the role of visual representation learning for reinforcement learning in two control tasks:

1. `ReachCube`: Reach a fixed cube at random positions with a time limit of 2 seconds.
2. `Pushing`: A more challenging task, where the goal is to push the cube to a random goal pose in the arena within a maximum time of 4 seconds.

We derive both tasks from the CausalWorld benchmark environments (Ahmed et al., 2021). The scene comprises a robot finger with three joints that can be controlled to manipulate a cube in a bowl-shaped stage. The robot is derived from the TriFinger design from Wüthrich et al. (2020). The input variables at time t are the camera observation o_t and a vector of observable variables x_t, which contains the joint angles and velocities in both tasks, as well as the target position for the cube in `Pushing`. We then feed the camera observation o_t into a given encoder e that was pre-trained on the dataset in Dittadi et al. (2021). The resulting $z_t = e(o_t)$ is concatenated with x_t, yielding a state vector $s_t = [x_t, z_t]$. For each task we then use SAC to train the policy with s_t as input, implemented with (Hill et al., 2018). The policy, value and Q networks are all implemented as MLPs with 2 hidden layers of size 256. Note that the representation function is fixed when training the policies, i.e. the encoder is not fine-tuned, as our goal is to investigate the link between representation properties and downstream RL performance. We perform a large-scale empirical study on the setup introduced above by training 11,520 policies across both tasks. The hyperparameter sweep is defined as follows:

- We train 240 β-VAEs, with a subset of the hyperparameter configurations and neural architecture from Dittadi et al. (2021). Specifically, we consider $\beta \in \{1, 2, 4\}$, β annealing in $\{0, 50000\}$ steps, unsupervised and weakly supervised training (Higgins et al., 2017a; Locatello et al., 2020), with and without input noise, and 10 random seeds per configuration. The latent space size is fixed to 10.
- For the `ReachCube` task, we train 20 downstream policies (with different random seeds) for each of the 240 VAEs. This results in 4,800 policies, which we train with SAC for 400k steps (approximately 2,400 episodes).
- Since the `Pushing` task takes substantially longer to train, we limit the number of policies to be trained on this task. First, we choose a subset of 96 VAEs corresponding to using only 4 random seeds for the

VAEs. Then, we only use 10 random seeds per representation. The resulting 960 policies are trained with SAC for 3M steps (approximately 9,000 episodes).

- Finally, for both tasks we also investigate the role of regularization on the policy. More specifically, we repeat the two training sweeps from above (5,760 policies), with the difference that now the policies are trained with L1 regularization on the first layer.

We evaluate the generalization of policies in three different environments: (1) in the training environment which is the default setting in RL, (2) the OOD1 setting where we use cube colors that are OOD for the MLP policy but still in-distribution for the encoder, and (3) the more challenging OOD2-sim setting where the cube colors are also OOD for the encoder. Finally, we evaluate zero-shot sim2real transfer on the real-world robotic platform. We refer to this generalization scenario as OOD2-real (this corresponds to OOD2-B in (Dittadi et al., 2021)). See Appendix A for further implementation details.

5 Results

We split the discussion of our results into three parts: We start in Section 5.1 by presenting the training results of our large-scale sweep, and how different components of the pre-trained representations and regularization affects in-distribution performance and training reward. Next, an extensive account of predictive factors for out-of-distribution generalization for RL from pre-trained representations is given in Section 5.2, focusing on the simulated environment. Finally, we extensively evaluate zero-shot sim2real transfer to the real robot without any fine-tuning in Section 5.3 and also discuss predictors for OOD generalization when going to the real world.

5.1 Results in the training environment

Fig. 2 shows the training curves of all policies for `ReachCube` and `Pushing` in terms of the task-specific success metric. While here we use success metrics for interpretability, in general we will measure performance by the cumulative reward. In `ReachCube`, the success metric indicates progress from the initial end effector position to the optimal distance from the center of the cube. It is 0 if the final distance is greater than or equal to the initial distance, and 1 if the end effector is touching the center of a face of the cube. In `Pushing`, the success metric is defined as the volumetric overlap of the cube with the goal cube, and the task can be visually considered solved with a score around 80%.

From the training curves we can conclude that both tasks can be consistently solved from pixel data using pre-trained representations. In particular, all policies on `ReachCube` attain almost perfect scores. `Pushing` is a much more complex tasks, involving learning the non-linear rigid-body interactions. Unsurprisingly, this task requires significantly more training, and the variance of performance across policies is larger. Nonetheless, almost all policies learn to solve the task satisfactorily. To investigate the effect of representations on the training reward, we now compute its Spearman rank correlations with various supervised and unsupervised metrics of the representations (Fig. 2 bottom). On `ReachCube`, the final reward correlates with ELBO and reconstruction loss. A simple supervised metric to evaluate a representation is how well a small downstream model can predict the ground-truth FoV. Following Dittadi et al. (2021), we use the MLP10000 and GBT10000 metrics, where MLPs and GBTs are trained for predicting the FoVs from 10,000 samples (we will simply call these metrics MLP and GBT in the following). Training reward correlates with these metrics as well, especially with the MLP accuracy. This is not entirely surprising: if an MLP can predict the FoVs from the representations, our policies using the same MLP architecture could in principle internally compute the FoVs relevant for the task. Interestingly, the correlation with the overall MLP accuracy mostly stems from the prediction accuracy of the cube pose FoVs, which are in fact the ones that are not included in the ground-truth robot state x_t. These results suggest that these metrics can be used to select good representations for downstream RL. On the more challenging task of `Pushing`, the correlations are milder but most of them still statistically significant. In general, all correlations discussed in this paper are statistically significant (colored coefficients in figures whenever p<0.05).

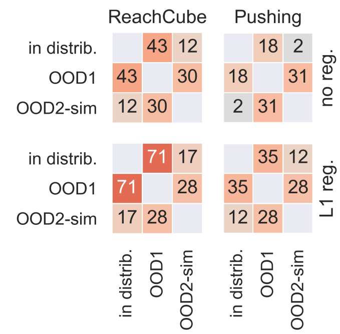

Figure 3. Rewards on the training environment (in distrib.) correlate with OOD rewards.

Summary. Both tasks can be consistently solved from pixels using pre-trained representations. Unsupervised (ELBO, reconstruction loss) and supervised (ground-truth factor prediction) metrics of the representations are correlated with reward in the training environment.

5.2 Out-of-distribution generalization in simulation

From train time performance to OOD generalization. Fig. 3 shows that in-distribution reward correlates with OOD1 performance on both tasks, especially with L1 regu-

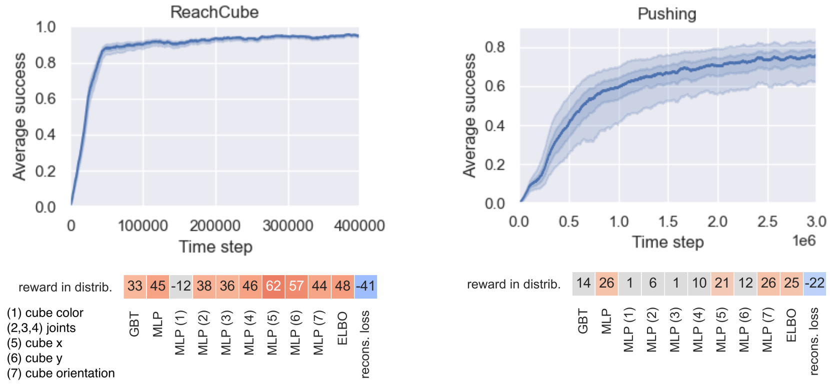

Figure 2. Top: Average training success, aggregated over *all* policies from the sweep (median, quartiles, 5th/95th percentiles). Bottom: Rank correlations between representation metrics and training reward, in the case without policy regularization.

larization. Moreover, rewards in OOD1 and OOD2 environments are positively correlated to a similar extent in both tasks, with or without regularization. As a reminder, OOD1 means that the downstream policy is OOD but the encoder is still in-distribution, while OOD2 means that the encoder is also OOD.

Unsupervised metrics and informativeness. In Fig. 4 (left) we assess the relation of OOD reward with unsupervised metrics (ELBO and reconstruction loss) and downstream performance on factor prediction (using MLP and GBT). Both ELBO and reconstruction loss exhibit a correlation with OOD1 reward, but not with OOD2 reward. These unsupervised metrics can thus be useful for selecting representations that will lead to more robust downstream RL tasks, as long as the representation function is in-distribution. While the GBT score is not correlated with reward under distribution shift, we observe a significant correlation between OOD1 reward and the MLP score, which measures downstream factor prediction accuracy of an MLP with the same architecture as the one parameterizing the policies. As in Section 5.1, we further investigate the source of this correlation, and find it in the pose parameters of the cube. Correlations in the OOD2 setting are much weaker, thus we conclude that these metrics do not appear helpful for model selection in this case. Our results on `Pushing` confirm these conclusions although correlations are generally weaker, presumably due to the more complicated nature of this task. (see Appendix B.2).

Representation robustness. We continue by analysing the link between generalization in RL and the generalization scores (GS) discussed in Section 2. We stress that, while the MLP metrics considered in the previous paragraph measure downstream FoV prediction in the VAE's training distribution, the GS scores assess the generalization of FoV predictors (using an MLP) under distribution shifts for a given representation. For both OOD scenarios, the distribution shifts underlying these GS scores are the same as the ones in the RL tasks in simulation. We summarize our findings in Fig. 4 (right) on the `ReachCube` task without regularization. Reward in the OOD1 setting is significantly correlated with the GS-OOD1 metric of the policies' underlying representation. We observe an even stronger correlation between the reward in the simulated OOD2 setting and the corresponding GS-OOD2-sim and GS-OOD2-real scores. On a per-factor level, we see that the source of the observed correlations primarily stems from the generalization scores w.r.t. the pose parameters of the cube. The OOD generalization metrics can therefore be used as proxies for the corresponding form of generalization in downstream RL tasks. This has practical implications for the training of RL downstream policies which are generally known to be brittle to distribution shifts, as we can assess a representations' generalization score from a few labeled images. This allows for selecting representations that yield more generalizing downstream policies.

Disentangled representations. Almost perfect disentanglement has been shown to be helpful for downstream performance and OOD1 generalization even with MLP downstream tasks (Dittadi et al., 2021). However, in `ReachCube` without regularization, we only observe a weak correlation with some disentanglement metrics (Fig. 5). In agreement with (Dittadi et al., 2021), disentanglement does not seem to correlate with OOD2 generalization. Dittadi et al. (2021) observed that disentanglement correlates with the informativeness of a representation. To understand if these weak correlations originate from this common confounder, we investigate whether disentanglement is still correlated with a higher OOD1 reward if we compare representations with similar MLP FoV prediction accuracy. Given two representations with the same MLP accuracy, does the more disentangled one exhibit better OOD1 generalization? To measure this we predict success from the MLP accuracy using kNN (k=5) (Locatello et al., 2019b)

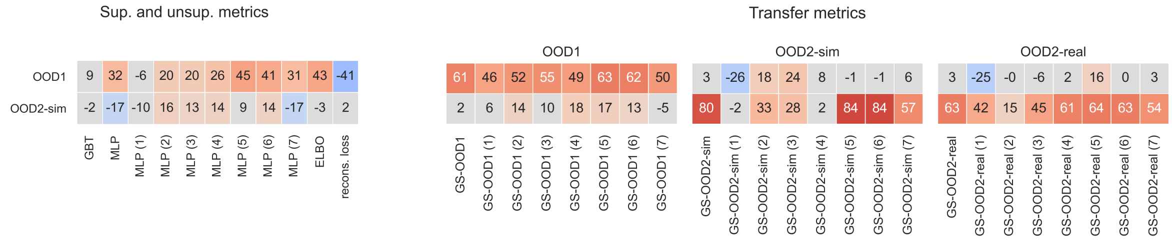

Figure 4. Rank correlations of representation properties (left: unsupervised metrics and informativeness, right: transfer metrics) with OOD1 and OOD2 reward on `ReachCube` without regularization. Numbering when splitting metrics by FoV: (1) cube color; (2–4) joint angles; (5–7) cube position and rotation.

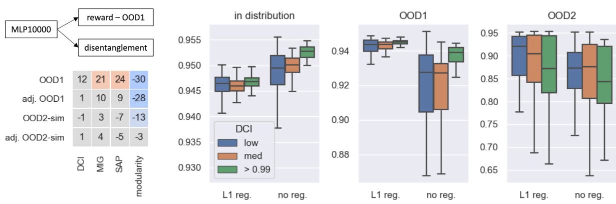

Figure 5. Box plots: fractional success on `ReachCube` split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. Although from the correlation matrix (left) we observe a mild correlation between some disentanglement metrics and OOD1 (but not OOD2) generalization, this does not hold when adjusting for representation informativeness. L1 regularization in the first layer of the MLP policy has a positive effect on OOD1 and OOD2 generalization with minimal sacrifice in terms of training reward (see scale).

and compute the residual reward by subtracting the amount of reward explained by the MLP accuracy. In Fig. 5 we see that this resolves the remaining correlations with disentanglement. Thus, for the downstream tasks considered here, disentanglement does not seem to be useful for downstream OOD generalization. We present similar results on `Pushing` in Appendix B.2.

Policy regularization and observation noise. It might seem unsurprising that disentanglement is not useful for generalization in RL, as MLP policies do not have any explicit inductive bias to exploit it. Thus, we attempt to introduce such inductive bias by repeating all experiments with L1 regularization on the first layer of the policy. As discussed in Appendix B.2, although regularization has a positive effect on OOD1 and OOD2 generalization in general (Fig. 5, right), we see no link with disentanglement. In agreement with (Dittadi et al., 2021), we also find that observation noise when training representations is beneficial for OOD2 generalization (see Appendix B.2).

Strong OOD shifts: evaluating on a novel shape. On the `ReachCube` task, we also tested generalization w.r.t. a novel object shape by replacing the cube with an unmovable sphere. This corresponds to a strong OOD2-type shift, since shape was never varied when training the representations. We then evaluated the trained policies as before, with the same color splits. Surprisingly, the policies appear to handle the novel shape. In fact, when the sphere has the same colors that the cube had during policy training, *all* policies get closer than 5cm to the sphere on average, with a mean success metric of about 95%. On sphere colors from the OOD1 split, more than 98.5% move the finger closer than this threshold, and on the strongest distribution shift (OOD2-sim colors and cube replaced by sphere) almost 70% surpass that threshold with an average success metric above 80%.

Summary. Reward from the training environment is significantly correlated with OOD generalization reward, as long as the encoder remains in its training distribution (OOD1 generalization). The OOD1 reward is significantly correlated with ELBO, reconstruction loss, and the MLP accuracy. This however does not hold for the OOD2-sim reward, hence these metrics cannot be used to predict OOD2 generalization in our experiments. The generalization metrics from (Dittadi et al., 2021), which measure robustness to distribution shifts, are significantly correlated with RL performance under similar distribution shifts. These metrics are thus useful for selecting representations that will yield robust downstream policies. Disentanglement does not seem to be beneficial for generalization in this setting, while input noise when training representations is beneficial for OOD2 generalization.

5.3 Deploying policies to the real world

We now evaluate a large subset of the trained models sim2real on the equivalent real robot (Wüthrich et al., 2020) without any additional fine-tuning. We are interested in quantifying if our models are able to generalize zero-shot on the real robot and attempt to uncover relevant metrics for predicting real world performance.

Reaching. We chose 960 policies trained in simulation, based on 96 representations and 10 random seeds, and evaluate them on two (randomly chosen, but significantly far apart) goal positions using a red cube. Note that although a

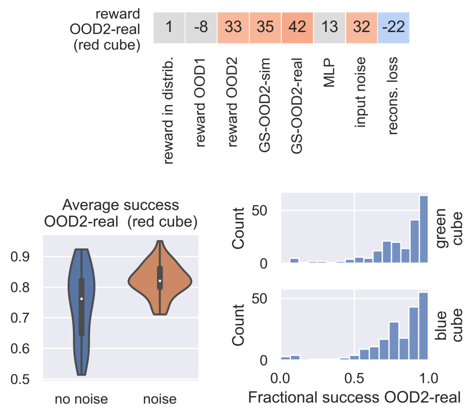

Figure 6. **Zero-shot sim2real** on `ReachBlock`. **Top:** Statistically significant rank-correlations on the real platform with a red cube. **Bottom left**: Training encoders with additive noise improves sim2real generalization. **Bottom right**: Histogram of fractional success in the more challenging OOD2-real-{green,blue} scenario from 50 policies across 4 goal positions.

red cube was in the training distribution in simulation, we consider this to be OOD2 because real world images already represent a strong distribution shift for the encoder (Dittadi et al., 2021; Djolonga et al., 2020). Although sim2real in robotics is considered to be very challenging without domain randomization or fine-tuning (Tobin et al., 2017; Finn et al., 2017; Rusu et al., 2017), many of our trained policies obtain a high fractional success score without resorting to these methods. In addition, in Fig. 6 (top) we observe consistent correlations between zero-shot real-world performance and some relevant quantities discussed previously. First, there is a positive correlation with the OOD2-sim reward: Policies that generalize well on unseen cube colors in simulation seem to generalize well to the real world, too. Second, representations with high GS-OOD2-sim and (especially) GS-OOD2-real scores are promising candidates for good sim2real transfer. Third, if no FoV labels are available, the weak statistically significant correlation with the reconstruction loss on the simulated images can be exploited for representation selection. Finally, as observed in (Dittadi et al., 2021) for significantly easier downstream tasks, input noise while learning representations is beneficial for sim2real generalization (see also Fig. 6, bottom left).

Based on these findings, we select 50 policies with high GS-OOD2-real, and evaluate them on the real world with a green and a blue cube, which is an even stronger OOD2 distribution shift than the one considered before. In Fig. 6 (bottom right), where performance metrics are averaged over 4 cube positions per policy, we observe that most policies can still reliably solve the task: approximately 80% of them position the finger less than 5 cm from the cube. Lastly, we mirrored the evaluations in simulation on an unseen green sphere object, and saw a surprisingly consistent finger movement to even such a new unseen object. We refer to Appendix B.3 in the supplementary material.

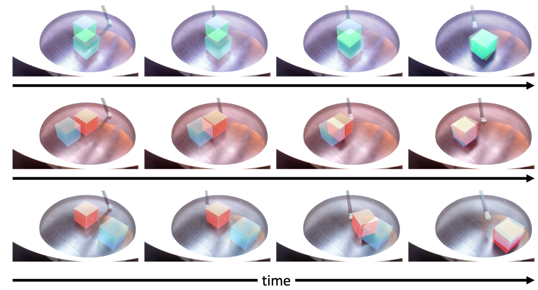

Figure 7. We select pushing control policies according to OOD2-real generalization with encoders trained on noisy input. We find that on the real robot setup (Wüthrich et al., 2020), these policies can zero-shot solve the task and push a cube to a specified goal position (transparent blue cube).

Pushing. We now test whether our real-world findings on `ReachCube` also hold for `Pushing`. We again selected a few policies with encoders being trained with added noise on the input and a high GS-OOD2-real score. We recorded episodes on diverse goal positions and cube colors to support our finding that it was also possible to obtain generalizing pushing policies on the real robot (Wüthrich et al., 2020) purely trained in simulation. In Fig. 7 we depict three representative episodes with successful task completions.

Summary. Policies trained solely within simulation can zero-shot solve the task on the real robot equivalent without any domain randomization or fine-tuning. We observe that OOD2 robustness of the underlying image encoder is a good predictor for real world performance as is the reconstruction loss of the VAE on simulated images and RL reward measured in a simulated OOD2 setting. For real-world application, we recommend using GS-OOD2-sim and GS-OOD2-real for model selection, and training the image encoder with additive noise.

6 Conclusion

Robust out-of-distribution (OOD) generalization is still one of the key open challenges in machine learning. With this work we presented a principled investigation of OOD generalization in the context of two practical downstream control tasks using RL from vision in simulation *and* the real world and how this is being driven by pre-trained representations. We worked out key predictors for various OOD generalization scenarios, whose statistical significance is supported by the over 10,000 control policies trained in this study. Ideally, our extensive investigation of representation learning for out-of-distribution generalization in reinforcement learning should encourage further work in this direction.

References

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. On robustness and transferability of convolutional neural networks. *arXiv preprint arXiv:2007.08558*, 2020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*, pages 9448–9458, 2019.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20(184): 1–25, 2019.

Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint 1807.10108*, 2018.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint 1907.07484*, 2019.

Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*, 2018.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289. PMLR, 2019.

Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf, and Stefan Bauer. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. *International Conference for Learning Representations*, 2021.

Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Yuxi Li. Deep reinforcement learning. *arXiv preprint arXiv:1810.06339*, 2018.

Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, and Peter Battaglia. Deep reinforcement learning with relational inductive biases. In *International Conference on Learning Representations*, 2019.

Daoming Lyu, Fangkai Yang, Bo Liu, and Steven Gustafson. Sdrl: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1): 2970–2977, 2019.

Haodi Zhang, Zihang Gao, Yi Zhou, Hao Zhang, Kaishun Wu, and Fangzhen Lin. Faster and safer training by embedding high-level knowledge into deep reinforcement learning. *arXiv preprint arXiv:1910.09986*, 2019.

Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214:106685, 2021.

David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., 1982.

Robert D Gordon and David E Irwin. What's in an object file? evidence from priming studies. *Perception & Psychophysics*, 58(8):1260–1277, 1996.

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in atari. *arXiv preprint arXiv:1906.08226*, 2019.

Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive representations of state. *NIPS*, 14(1555):30, 2001.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016a.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*, pages 4055–4065, 2017.

Andrea Dittadi, Frederik Träuble, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. On the Transfer of Disentangled Representations in Realistic Settings. In *International Conference on Learning Representations*, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017a.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. *arXiv preprint arXiv:2002.02886*, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer, 1999.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015a.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015b.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.

Manuel Wüthrich, Felix Widmaier, Felix Grimminger, Joel Akpo, Shruti Joshi, Vaibhav Agrawal, Bilal Hammoud, Majid Khadiv, Miroslav Bogdanovic, Vincent Berenz, et al. Trifinger: An open-source robot for learning dexterity. *arXiv preprint arXiv:2008.03596*, 2020.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *International Conference on Learning Representations*, 2016.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, 2018.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019a.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *arXiv preprint arXiv:1904.06991*, 2019.

Antti Honkela and Harri Valpola. Variational learning and bits-back coding: an information-theoretic view to bayesian learning. *IEEE transactions on Neural Networks*, 15(4):800–810, 2004.

James Townsend, Tom Bird, and David Barber. Practical lossless compression with latent variables using bits back coding. *arXiv preprint arXiv:1901.04866*, 2019.

Friso Kingma, Pieter Abbeel, and Jonathan Ho. Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables. In *International Conference on Machine Learning*, pages 3408–3417. PMLR, 2019.

Yangjun Ruan, Karen Ullrich, Daniel Severo, James Townsend, Ashish Khisti, Arnaud Doucet, Alireza Makhzani, and Chris J Maddison. Improving lossless compression rates via monte carlo bits-back coding. *arXiv preprint arXiv:2102.11086*, 2021.

Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*, 2018.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

Raphael Suter, Djordje Miladinović, Stefan Bauer, and Bernhard Schölkopf. Interventional robustness of deep latent variable models. In *International Conference on Machine Learning*, 2019.

William F Whitney, Min Jae Song, David Brandfonbrener, Jaan Altosaar, and Kyunghyun Cho. Evaluating representations by the complexity of learning low-loss predictors. *arXiv preprint arXiv:2009.07368*, 2020.

Muhammad Waleed Gondal, Manuel Wüthrich, Djordje Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *Advances in Neural Information Processing Systems*, 2019.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. *arXiv preprint arXiv:2006.07886*, 2020.

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108: 379–392, 2018.

Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *arXiv preprint arXiv:1506.07365*, 2015.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *arXiv preprint arXiv:1809.01999*, 2018.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, 2017b.

Herke Van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3928–3934. IEEE, 2016.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *arXiv preprint arXiv:1507.08750*, 2015.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *arXiv preprint arXiv:1605.07157*, 2016b.

Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. *arXiv preprint arXiv:1805.11592*, 2018.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.

Giuseppe Cuccu, Julian Togelius, and Philippe Cudré-Mauroux. Playing atari with six neurons. *arXiv preprint arXiv:1806.01363*, 2018.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. *arXiv preprint arXiv:2009.08319*, 2020.

Chelsea Finn, Tianhe Yu, Justin Fu, Pieter Abbeel, and Sergey Levine. Generalizing skills with semi-supervised reinforcement learning. In *International Conference on Learning Representations*, 2017.

Andrei A Rusu, Matej Večerík, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *Conference on Robot Learning*, pages 262–270, 2017.

Kavosh Asadi, David Abel, and Michael L Littman. Learning state abstractions for transfer in continuous control. *arXiv preprint arXiv:2002.05518*, 2020.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Reinforcement learning in robotics: Applications and real-world challenges. *Robotics*, 2(3):122–148, 2013.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12627–12637, 2019.

Mengyuan Yan, Qingyun Sun, Iuri Frosio, Stephen Tyree, and Jan Kautz. How to close sim-real gap? transfer with segmentation! *arXiv preprint arXiv:2005.07695*, 2020.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John

Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. In *Advances in Neural Information Processing Systems*, pages 14611–14624, 2019b.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

A Implementation details

Task definitions and rewards. We derived both tasks, `ReachCube` and `Pushing`, from the CausalWorld environments introduced by Ahmed et al. (2021). We pre-train representations on the dataset introduced by Dittadi et al. (2021), and allow only one finger to move in our RL experiments. We propose the `ReachCube` environment as an intermediate simpler RL environment that involves a fixed cube that cannot be moved. We used reward structures similar to those in Ahmed et al. (2021):

- `ReachCube`: $r_t = -750\,[d(g_t, e_t) - d(g_{t-1}, e_{t-1})]$
- `Pushing`: $r_t = -750\,[d(o_t, e_t) - d(o_{t-1}, e_{t-1})] - 250\,[d(o_t, g_t) - d(o_{t-1}, g_{t-1})] + \rho_t$

where t denotes the time step, $\rho_t \in [0, 1]$ is the fractional overlap with the goal cube at time t, $e_t \in \mathbf{R}^3$ is the end-effector position, $o_t \in \mathbf{R}^3$ the cube position, $g_t \in \mathbf{R}^3$ the goal position, and $d(\cdot, \cdot)$ denotes the Euclidean distance. The cube in `ReachCube` is fixed, i.e. $o_t = g_t$ for all t.

Besides the accumulated reward along episodes, that is determined by the reward function, we also report two reward-independent normalized success definitions for better interpretability: In `ReachCube`, the success metric indicates progress from the initial end effector position to the optimal distance from the center of the cube. It is 0 if the final distance is greater than or equal to the initial distance, and 1 if the end effector is touching the center of a face of the cube. In `Pushing`, the success metric is defined as the volumetric overlap of the cube with the goal cube, and the task can be visually considered solved with a score around 80%. We observed that accumulated reward and success are highly correlated with each other, thus allowing to use one or the other for measuring performance.

During training, the goal position is randomly sampled at every episode. Similarly, the object color is being sampled from the 4 specified train colors from D_1 that are corresponding to the OOD1-B split from Dittadi et al. (2021).

For each policy evaluation (in-distribution and out-of-distribution variants), we average the accumulated reward and final success across 200 episodes with randomly sampled cube positions and the respective object color in both tasks.

SAC implementation. Our implementation of SAC builds upon the `stable-baselines` package (Hill et al., 2018). We used the same SAC hyperparameters used for pushing in Ahmed et al. (2021). When using L1 regularization, we add to the loss function the L1 norm of the first layers of all MLPs, scaled by a coefficient α. We gradually increase regularization by linearly annealing α from 0 to $5 \cdot 10^{-7}$ in 200,000 time steps in `ReachCube`, and from 0 to $6 \cdot 10^{-8}$ in 3,000,000 time steps in `Pushing`.

B Additional results

B.1 Training environment

Fig. 2 in the main text shows correlations of unsupervised and supervised metrics with in-distribution reward for `ReachCube` and `Pushing`, only in the case without regularization. In Fig. 8 we also show these results in the case with regularization, as well as when adjusting for MLP informativeness.

B.2 Out-of-distribution generalization in simulation

In Section 5.2 we discussed rank-correlations of OOD rewards with unsupervised, informativeness and generalization scores on `ReachCube` without regularization. In Fig. 9 we also show these results for the case with regularization and on `Pushing`, as well as when adjusting for MLP informativeness. Without regularization, we observe on `Pushing` very similar correlations along all metrics as we observed on `ReachCube`, confirming our conclusions on this more complex task. When using regularization, rank correlations are very similar across both tasks. Interestingly, the correlation between GS-OOD2 scores and OOD2 generalization of the policy is even stronger when using the here studied type of regularization. In contrast to our observations without regularization, we find that the correlation between GS-OOD1 and OOD1 generalization of the policy disappears when adjusting for MLP informativeness.

Disentangled representations. As discussed in Section 5.2 for `ReachCube` without regularization, we observe in Fig. 9 a weak correlation between some disentanglement metrics and OOD1 reward, which however vanishes when adjusting for MLP informativeness. In agreement with Dittadi et al. (2021), we observe no significant correlation between disentanglement and OOD2 generalization, for both tasks, with and without regularization. From Fig. 10 we see that in some cases, especially without regularization, a very high DCI score seems to lead to better performance on average. However, this behavior is not significant (within error bars), as opposed to the results shown in simpler downstream tasks by Dittadi et al. (2021). Furthermore, this trend is likely due to representation informativeness, since the correlations with disentanglement disappear when adjusting for the MLP score, as discussed above.

Regularization. As seen in Fig. 10, regularization generally has a positive effect on OOD1 and OOD2 generalization, especially prominent in the OOD1 setting. On the other hand, it leads to lower training rewards both in `ReachCube` and in `Pushing`. In the latter, the performance drop is par-

Setting	Row	GBT	MLP	MLP (1)	MLP (2)	MLP (3)	MLP (4)	MLP (5)	MLP (6)	MLP (7)	ELBO	recons. loss	GS-OOD1	GS-OOD1 (1)	GS-OOD1 (2)	GS-OOD1 (3)	GS-OOD1 (4)	GS-OOD1 (5)	GS-OOD1 (6)	GS-OOD1 (7)	GS-OOD2-sim	GS-OOD2-sim (1)	GS-OOD2-sim (2)	GS-OOD2-sim (3)	GS-OOD2-sim (4)	GS-OOD2-sim (5)	GS-OOD2-sim (6)	GS-OOD2-sim (7)	GS-OOD2-real	GS-OOD2-real (1)	GS-OOD2-real (2)	GS-OOD2-real (3)	GS-OOD2-real (4)	GS-OOD2-real (5)	GS-OOD2-real (6)	GS-OOD2-real (7)	DCI	MIG	SAP	modularity
ReachCube, no regularization	reward in distrib.	33	45	-12	38	36	46	62	57	44	48	-41	57	45	37	41	45	52	47	54	6	-25	28	34	22	-4	-4	12	12	-24	13	-15	7	22	4	9	34	40	38	-34
ReachCube, no regularization	(adj. MLP) reward in distrib.	17	17	-21	37	35	39	45	46	16	28	-24	38	38	35	34	39	42	39	33	7	-20	22	31	19	5	5	6	9	-13	15	-10	11	20	9	6	19	26	22	-40
Pushing, no regularization	reward in distrib.	14	26	1	6	1	10	21	12	26	25	-22	34	17	17	12	19	23	19	35	-13	-8	8	2	0	-14	-17	-7	-2	-17	-0	-9	-3	2	-7	-1	14	16	24	-10
Pushing, no regularization	(adj. MLP) reward in distrib.	-11	4	-1	-9	-9	-9	-3	-12	3	6	-10	9	19	5	1	0	7	6	6	-6	-14	-1	-4	4	-5	-7	-0	6	-10	-4	6	4	10	7	3	-11	-5	2	-11
ReachCube, L1 regularization	reward in distrib.	3	37	10	3	1	6	26	21	36	32	-36	30	16	10	8	6	31	22	32	6	-11	-5	-4	5	5	3	12	12	-11	0	4	3	21	11	9	3	4	8	-3
ReachCube, L1 regularization	(adj. MLP) reward in distrib.	-27	-15	3	-12	-15	-16	-19	-17	-17	-13	3	-15	-13	-7	-15	-14	1	-7	-17	8	6	-19	-19	-3	19	17	2	7	12	3	15	7	10	14	5	-27	-26	-21	2
Pushing, L1 regularization	reward in distrib.	44	23	15	30	28	31	31	32	23	28	-13	16	-7	1	12	22	14	12	21	-2	12	24	27	5	-9	-6	-0	-2	3	4	-12	-2	1	-6	2	43	39	31	0
Pushing, L1 regularization	(adj. MLP) reward in distrib.	22	8	13	18	19	18	9	10	6	16	-5	-1	-8	-10	1	8	4	6	2	4	9	20	22	12	-5	-1	8	1	6	-9	-5	1	3	4	6	23	23	16	3

Figure 8. Rank correlations between metrics and in-distribution reward, with and without adjusting for informativeness.

Setting	Row	GBT	MLP	MLP (1)	MLP (2)	MLP (3)	MLP (4)	MLP (5)	MLP (6)	MLP (7)	ELBO	recons. loss	GS-OOD1	GS-OOD1 (1)	GS-OOD1 (2)	GS-OOD1 (3)	GS-OOD1 (4)	GS-OOD1 (5)	GS-OOD1 (6)	GS-OOD1 (7)	GS-OOD2-sim	GS-OOD2-sim (1)	GS-OOD2-sim (2)	GS-OOD2-sim (3)	GS-OOD2-sim (4)	GS-OOD2-sim (5)	GS-OOD2-sim (6)	GS-OOD2-sim (7)	GS-OOD2-real	GS-OOD2-real (1)	GS-OOD2-real (2)	GS-OOD2-real (3)	GS-OOD2-real (4)	GS-OOD2-real (5)	GS-OOD2-real (6)	GS-OOD2-real (7)	DCI	MIG	SAP	modularity
ReachCube, no regularization	reward OOD1	9	32	-6	20	20	26	45	41	31	43	-41	61	46	52	55	49	63	62	50	3	-26	18	24	8	-1	-1	6	3	-25	-0	-6	2	16	0	3	12	21	24	-30
ReachCube, no regularization	(adj. MLP) reward OOD1	-0	9	-5	17	21	18	29	26	7	18	-21	36	28	37	38	34	44	44	25	-2	-17	12	19	11	1	1	-3	-7	-14	5	-3	-2	5	-7	-4	1	10	9	-28
ReachCube, no regularization	reward OOD2	-2	-17	-10	16	13	14	9	14	-17	-3	2	2	6	14	10	18	17	13	-5	80	-2	33	28	2	84	84	57	63	42	15	45	61	64	63	54	-1	3	-7	-13
ReachCube, no regularization	(adj. MLP) reward OOD2	0	-4	-4	10	10	10	9	9	-4	5	-8	6	3	7	5	9	12	6	3	59	-11	24	22	7	62	62	41	45	25	14	32	45	45	44	40	1	4	-5	-3
Pushing, no regularization	reward OOD1	2	32	-18	7	10	14	34	25	32	37	-34	69	69	60	57	48	63	69	56	-5	-40	6	13	0	-12	-14	6	7	-30	3	2	2	15	-2	11	4	14	33	-30
Pushing, no regularization	(adj. MLP) reward OOD1	-18	9	-22	-6	-3	1	13	8	10	18	-19	48	57	47	43	33	51	57	36	-18	-25	-8	-1	-2	-19	-20	-14	-7	-27	3	-3	-5	-1	-15	-6	-16	-6	16	-22
Pushing, no regularization	reward OOD2	4	2	-12	17	16	14	22	12	2	15	-12	17	25	24	24	30	25	24	12	78	-13	33	32	-2	76	77	64	75	36	13	49	68	72	70	63	5	9	9	-16
Pushing, no regularization	(adj. MLP) reward OOD2	-9	-7	-16	-1	3	-2	8	1	-4	2	-2	10	14	16	17	16	20	15	6	55	-12	16	18	1	57	58	41	52	29	12	33	50	50	50	40	-9	-7	-3	-15
ReachCube, L1 regularization	reward OOD1	11	34	9	10	10	16	38	32	32	37	-36	48	25	21	23	24	49	40	42	13	-19	8	6	0	12	11	15	12	-11	-1	4	6	27	12	12	13	17	15	-13
ReachCube, L1 regularization	(adj. MLP) reward OOD1	-10	-19	-7	-0	2	-7	-3	-5	-19	-13	10	-11	-7	-8	-10	-9	-2	-5	-15	-4	-1	-0	-1	3	5	2	-8	-15	2	0	2	-10	-2	-7	-14	-9	-9	-18	-0
ReachCube, L1 regularization	reward OOD2	-4	-11	-4	8	4	6	3	5	-12	-3	-1	-3	-0	1	-3	4	5	1	-5	89	-2	25	17	2	94	93	65	72	44	13	52	65	71	72	64	-3	-2	-11	-1
ReachCube, L1 regularization	(adj. MLP) reward OOD2	-7	-1	-3	-1	-3	-1	-3	-5	-1	4	-11	-1	-6	-7	-9	-6	-5	-8	-1	63	-12	8	5	1	68	66	45	52	23	11	36	47	47	51	46	-7	-8	-14	12
Pushing, L1 regularization	reward OOD1	20	42	3	12	12	17	39	29	42	51	-44	62	50	40	38	38	45	55	59	4	-27	19	22	11	-6	-6	9	13	-14	-4	-4	3	22	5	13	22	22	33	-9
Pushing, L1 regularization	(adj. MLP) reward OOD1	-15	-11	-1	-7	-11	-7	-6	-2	-9	3	-4	7	19	-3	-4	-2	7	17	9	-11	0	-2	-4	15	-9	-7	-16	-11	6	-4	-9	-9	-0	-9	-11	-15	-16	-5	2
Pushing, L1 regularization	reward OOD2	9	6	-2	16	11	11	18	10	7	14	-12	8	12	8	7	16	7	7	11	85	-10	34	30	2	83	85	69	79	43	15	52	69	77	75	68	9	9	9	-2
Pushing, L1 regularization	(adj. MLP) reward OOD2	-6	-0	-6	-0	-2	-4	4	-3	1	5	-8	-1	-1	-1	-1	-0	-3	-5	5	70	-8	17	16	7	71	73	55	63	40	14	41	56	61	63	54	-7	-10	-6	2

Figure 9. Rank correlations between metrics and OOD reward, with and without adjusting for informativeness.

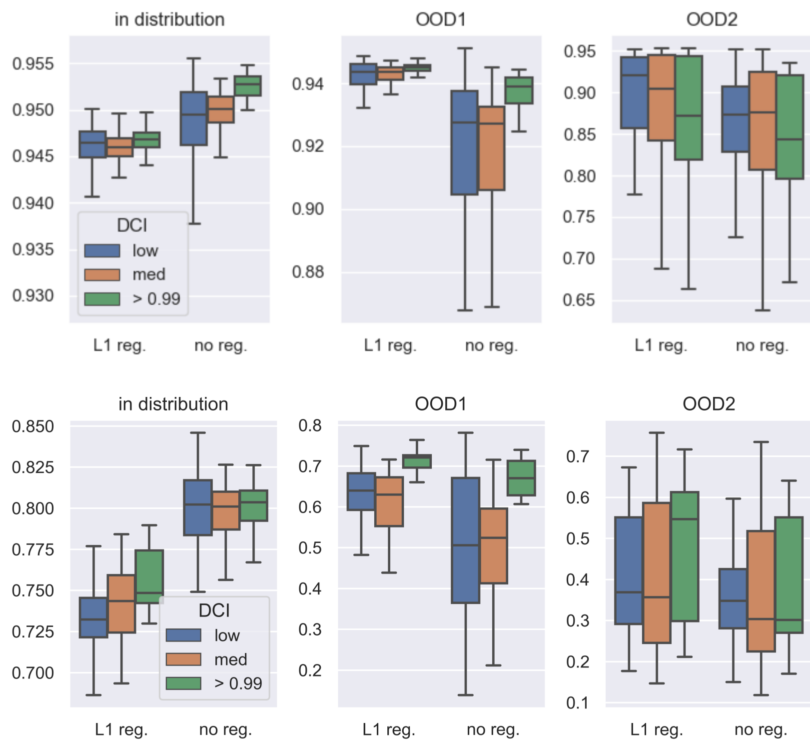

Figure 10. Fractional success on `ReachCube` (**top**) and `Pushing` (**bottom**), split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. Results for `ReachCube` are also reported in Fig. 5 in Section 5.2.

ticularly significant, while in `ReachCube` it is negligible.

Sample efficiency. In addition to the analysis reported in the main paper, we performed an analysis on representation properties affecting sample efficiency, which we summarize in Fig. 11 for `ReachCube` and Fig. 12 for `Pushing`. Specifically, we stored checkpoints of our policies at $t \in \{20k, 50k, 100k, 400k\}$ for `ReachCube` and $t \in \{200k, 500k, 1M, 3M\}$ for `Pushing`. We then evaluate policies at these time step on the same three different environments as before: (1) on the cube colors form training; (2) on the OOD1 cube colors; and (3) on the OOD2-sim cube colors.

On `ReachCube` (Fig. 11), we observe very similar trends with and without regularization: Unsupervised metrics (ELBO and reconstruction loss) display a correlation with the training reward, as do the supervised informativeness metrics (GBT and MLP). This is strongest on early timesteps, meaning these scores could be important for sample efficiency. Similarly, we observe a correlation with the disentanglement scores DCI, MIG and SAP. With the help of the additional evaluation of rewards adjusted for MLP informativeness, we can attribute this correlation again to this common confounder. Lastly, we see that the generalization scores are correlated with generalization of the corresponding policies under OOD1 and OOD2 shifts for all recorded time steps.

On `Pushing` (Fig. 12), many correlations at early checkpoints are significantly reduced, especially with regularization. This behavior might be due to the more complicated nature of the task, which involves learning to reach the cube first, and then push it to the goal. Correlations are primarily seen towards the end of training, with similar spurious correlations with disentanglement as elaborated above. Importantly, correlations between generalization scores and policy generalization under the same distribution shift remain strong and statistically significant.

Generalization to a novel shape. As mentioned in Section 5.2, on the `ReachCube` task, we also tested generalization w.r.t. a novel object shape by replacing the cube with an unmovable sphere. Remember, this corresponds to a strong OOD2-type shift, since shape was never varied when training the representations. We then evaluated a subset of 960 trained policies as before, with the same color splits. Surprisingly, the policies appear to handle the novel shape as we see from the histograms in Fig. 13 in terms of success and final distance. In fact, when the sphere has the same colors that the cube had during policy training, *all* policies get closer than 5cm to the sphere on average, with a mean

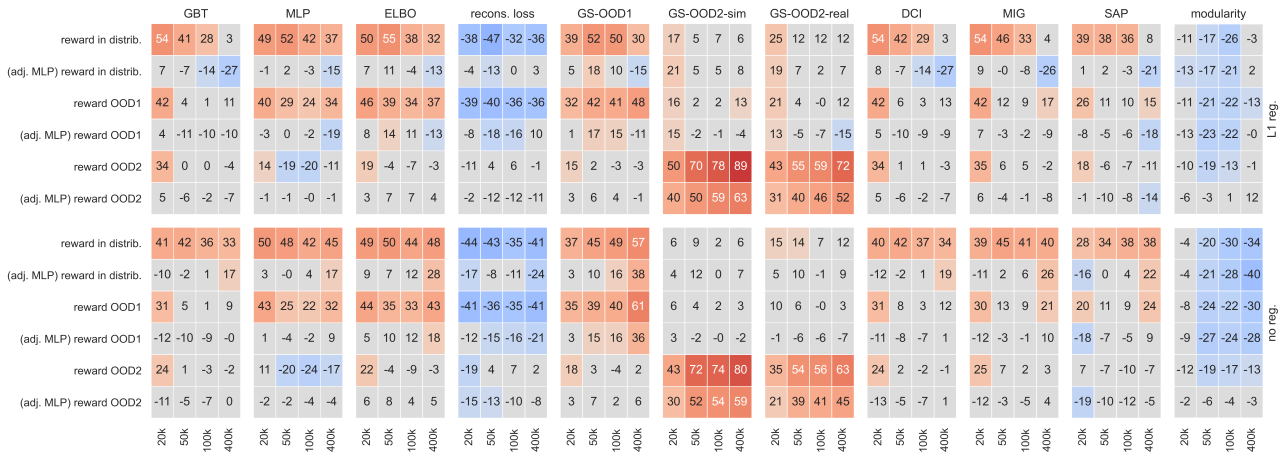

Figure 11. Sample efficiency analysis for `ReachCube`. Rank correlations of rewards with relevant metrics along multiple time steps

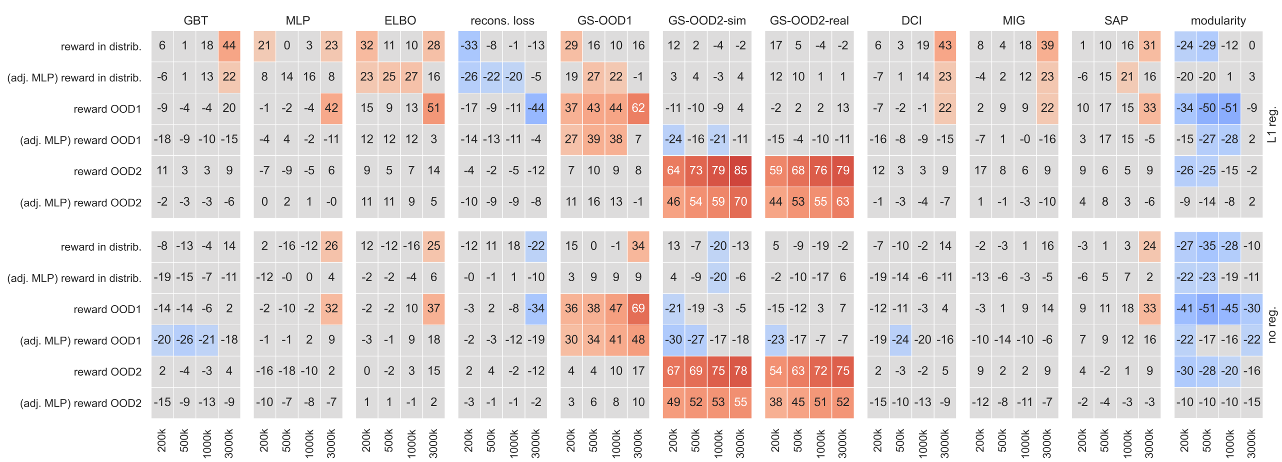

Figure 12. Sample efficiency analysis for `Pushing`. Rank correlations of rewards with relevant metrics along multiple time steps

success metric of about 95%. On sphere colors from the OOD1 split, more than 98.5% move the finger closer than this threshold, and on the strongest distribution shift (OOD2-sim colors and cube replaced by sphere) almost 70% surpass that threshold with an average success metric above 80%.

B.3 Deploying policies to the real world

In Fig. 14 we depict three representative episodes of testing a reach policy on the real robot for the strong OOD shift with a novel sphere object shape instead of the cube from training.

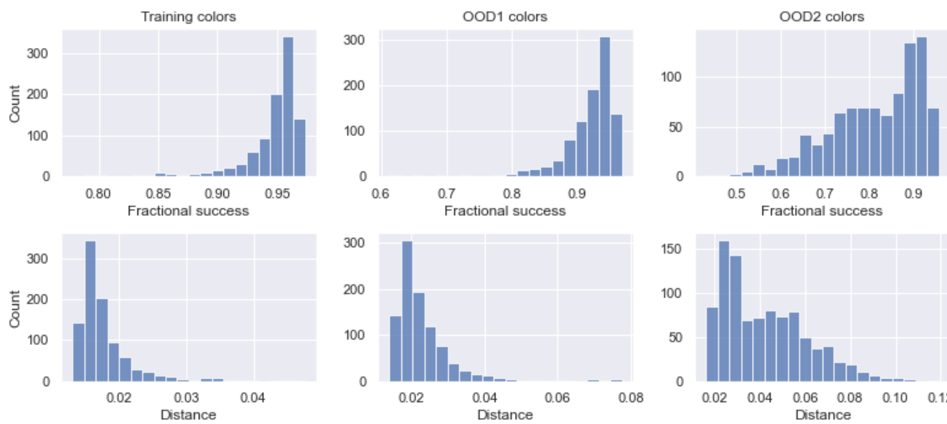

Figure 13. Testing `ReachCube` policies under the same IID, OOD1 and OOD2 evaluation protocols regarding object color in simulation but replacing the cube with a novel shape in the form of a sphere.

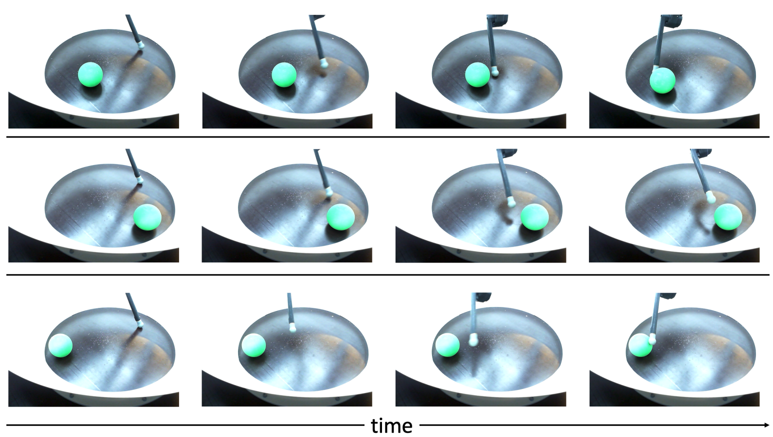

Figure 14. Transferring `ReachCube` models to the real robot setup without any fine-tuning on a green sphere (unseen shape *and* color).