THE ROLE OF PRETRAINED REPRESENTATIONS FOR THE OOD GENERALIZATION OF RL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Building sample-efficient agents that generalize out-of-distribution (OOD) in real-world settings remains a fundamental unsolved problem on the path towards achieving higher-level cognition. One particularly promising approach is to begin with low-dimensional, pretrained representations of our world, which should facilitate efficient downstream learning and generalization. By training 240 representations and over 10,000 reinforcement learning policies on a simulated robotic setup, we evaluate to what extent different properties of pretrained VAE-based representations affect the OOD generalization of downstream agents. We observe that many agents are surprisingly robust to realistic distribution shifts, including the challenging sim-to-real case. In addition, we find that the generalization performance of a simple downstream proxy task reliably predicts the generalization performance of our reinforcement learning control tasks under a wide range of practically relevant OOD settings. Such proxy tasks can thus be used to select pretrained representations that will lead to agents that generalize out-of-distribution.

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 beyond the i.i.d. (independent and identically distributed) data setting (Azulay & Weiss, 2019; Barbu et al., 2019; Djolonga et al., 2020; Funk et al., 2021; Gulrajani & Lopez-Paz, 2020; Hendrycks & Dietterich, 2019; Koh et al., 2021; Michaelis et al., 2019; Roy et al., 2018; Schölkopf et al., 2021). One instance of such models are agents that learn by interacting with a training environment and we would like them to generalize to other environments with different statistics (Ahmed et al., 2021; Cobbe et al., 2019; Ke et al., 2021; Pfister et al., 2019; Zhang et al., 2018). 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 from the training setup, e.g. the shape, color, and other object properties, or when transferring from simulation to real world.

Humans do not suffer from these pitfalls when transferring learned skills beyond a narrow training domain, presumably because they represent visual sensory data in a concise and useful manner (Anand et al., 2019; Gordon & Irwin, 1996; Lake et al., 2017; Marr, 1982; Spelke, 1990). Therefore, a particularly promising path is to base decisions and predictions on similar low-dimensional representations of our world (Barreto et al., 2017; Bengio et al., 2013; Finn et al., 2016; Kaiser et al., 2019). The learned representation should facilitate efficient downstream learning (Anand et al., 2019; Eslami et al., 2018; Stooke et al., 2020) and exhibit better generalization (Srinivas et al., 2020; Zhang et al., 2020). Learning such a representation from scratch for every downstream task and every new variation would be highly inefficient. If we learned to juggle three balls, we should be able to generalize to oranges or apples without learning again from scratch. We could even do it with cherimoyas, a type of fruit that we might have never seen before. We can effectively reuse our generic representation of the world.

We thus consider deep learning agents trained from pretrained representations and ask the following questions: To what extent do they generalize under distribution shifts similar to those mentioned above? Do they generalize in different ways or to different degrees depending on the type of distribution shift, including sim-to-real? Can we predict the OOD generalization of downstream agents from properties of the pretrained representations?

To answer the questions above, we need our experimental setting to be realistic, diverse, and challenging, but also controlled enough for the conclusions to be sound. We therefore base our study on the robot platform introduced by Wüthrich et al. (2020). The scene comprises a robot finger with three joints that can be controlled to manipulate a cube in a bowl-shaped stage. Dittadi et al. (2021) conveniently introduced a dataset of simulated and real-world images of this setup with ground-truth labels, which can be used to pretrain and evaluate representations. To train downstream agents, we adapted the simulated reinforcement learning benchmark CausalWorld from Ahmed et al. (2021) that was also developed for this robotic setup. Building upon these works, we can design our experimental study as follows: First, we pretrain representations from static simulated images of the setup and evaluate a collection of representation metrics (see Table 1). Following prior work (Eslami et al., 2018; Ghadirzadeh et al., 2017; Ha & Schmidhuber, 2018; Nair et al., 2018; Van Hoof et al., 2016; Watter et al., 2015), we focus on autoencoder-based representations. Then, we train downstream agents from this fixed representation on a set of environments. Finally, we investigate the zero-shot generalization of these agents to new environments that are out of the training distribution, including the real robot. Our setup is summarized in Fig. 1.

Metric class	Metrics
Unsupervised	ELBO, rec. loss
FoV prediction	MLP, GBT
Generalization	GS-OOD1, GS-OOD2-sim, GS-OOD2-real
Disentanglement	DCI, MIG, SAP, Modularity

Table 1: Overview of the 11 representation metrics discussed in this work.

The goal of this work is to provide the first systematic and extensive account of the OOD generalization of downstream RL agents in a robotic setup, and how this is affected by characteristics of the upstream pretrained representations. We summarize our contributions as follows:

- We train 240 representations and 11,520 downstream policies,[1] and systematically investigate the performance of these policies under a diverse range of distribution shifts.[2]
- We extensively analyze the relationship between the generalization of our RL agents and a substantial set of representation metrics (see Table 1).
- Notably, we find that a specific representation metric that measures the generalization of a simple downstream proxy task reliably predicts the generalization of downstream RL agents under the broad spectrum of OOD settings considered here. This metric can thus be used to select pretrained representations that will lead to more robust downstream policies.
- In the most challenging of our OOD scenarios, we deploy a subset of the trained policies to the corresponding real-world robotic platform, and observe surprising zero-shot sim-to-real generalization without any fine-tuning or domain randomization.

2 Background

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

Variational autoencoders. VAEs (Kingma & Welling, 2014; Rezende et al., 2014) are a framework for optimizing a latent variable model $p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) d\mathbf{z}$ 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 $q_\phi(\mathbf{z} \mid \mathbf{x})$ with parameters ϕ approximates the intractable posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$. The approximate posterior and generative model, typically called encoder and decoder and parameterized by neural networks, are jointly optimized by maximizing a lower bound to the log likelihood (the ELBO):

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - D_{\mathrm{KL}} \left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\right) = \mathcal{L}^{ELBO}_{\theta,\phi}(\mathbf{x}) \,. \tag{1}$$

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

Reinforcement learning. A Reinforcement Learning (RL) problem is typically modeled as a Partially Observable Markov Decision Process (POMDP) defined as a tuple $(S, A, T, R, \Omega, O, \gamma, \rho_0, H)$ with

[1] Training the representations 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 processors.

[2] Additional results and videos are provided at `https://sites.google.com/view/ood-rl`.

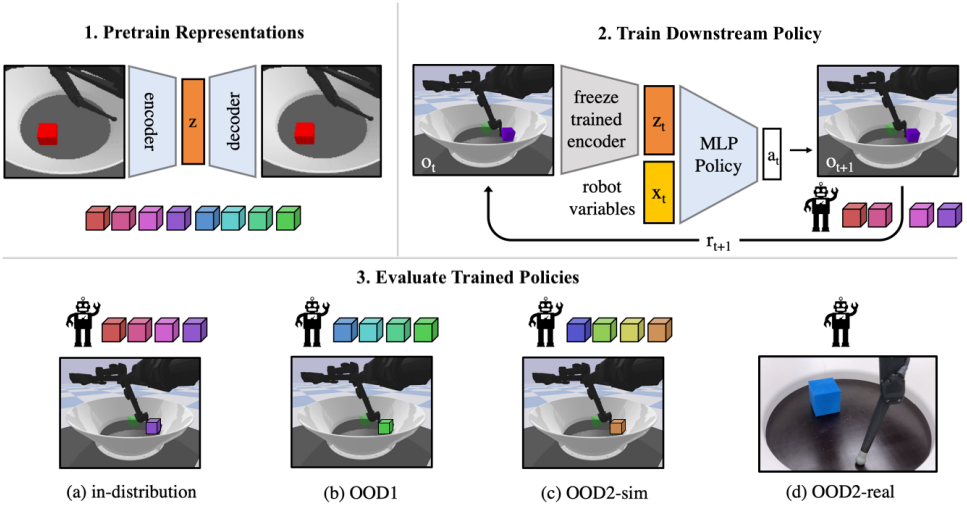

Figure 1: **Overview of our experimental setup for investigating out-of-distribution generalization in downstream tasks.** (1) We train 240 β-VAEs on the robotic dataset from Dittadi et al. (2021). (2) We then train downstream policies to solve *object reaching* or *pushing*, using multiple random RL seeds per VAE. The input to a policy consists of the output of a pretrained encoder and additional task-related observable variables. Crucially, the policy is only trained on a subset of the cube colors from the pretraining 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) sim-to-real zero-shot on the real-world setup.

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 (Fujimoto et al., 2018; Haarnoja et al., 2018b; Schulman et al., 2015a;b; 2017; Silver et al., 2014; Sutton et al., 1999). 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., 2018b). SAC aims to improve sample-inefficiency and convergence in RL by simultaneously maximizing the expected reward and the entropy $H(\pi(\cdot|s_t))$.

A robotic setup to evaluate out-of-distribution generalization. Our study is based on a real robot platform where a robotic finger with three joints manipulates a cube in a bowl-shaped stage (Wüthrich et al., 2020). We pretrain representations on a labeled dataset introduced by Dittadi et al. (2021) which consists of simulated and real-world images of this setup. This dataset has 7 underlying factors of variation (FoV): angles of the three joints, and position (x and y), orientation, and color of the cube. After training the representations, we train downstream agents and evaluate their generalization on an adapted version of the simulated CausalWorld benchmark (Ahmed et al., 2021) that was developed for the same setup. Finally, we can test sim-to-real generalization on the real robot.

Our experimental setup, illustrated in Fig. 1, allows us to systematically investigate a broad range of out-of-distribution scenarios in a controlled way. We pretrain our representations from this simulated dataset that covers 8 distinct cube colors. We then train an agent from this fixed representation on a subset of the cube colors, and evaluate it (1) on the same colors (this is the typical scenario in RL), (2) on the held-out cube colors known to the representation, or (3) OOD w.r.t. the representation training distribution, e.g. on novel colors and shapes or on the real world.

We closely follow the framework for measuring OOD generalization proposed by Dittadi et al. (2021). In this framework, a representation is initially learned on a training set $\mathcal{D}$, and generalization is then investigated by training a simple downstream model on a subset $\mathcal{D}_1 \subset \mathcal{D}$ to predict the

ground-truth factors from the learned representation. The model is then tested on a set $\mathcal{D}_2$ that differs distributionally from $\mathcal{D}_1$, 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 $\mathcal{D}_2$: First, the case when $\mathcal{D}_2 \subset \mathcal{D}$, i.e. the OOD test set is a subset of the dataset for representation learning. This is denoted by **OOD1** and corresponds to the scenario (2) from the previous paragraph. In the other scenario, referred to as **OOD2**, $\mathcal{D}$ and $\mathcal{D}_2$ are disjoint and distributionally different. This even stronger OOD shift corresponds to case (3) above. The generalization score for $\mathcal{D}_2$ is then measured by the (normalized) mean absolute prediction error across all FoVs except for the one that is OOD. As factor prediction model, we will likewise consider their simple 2-layer MLP. This simple and cheap generalization metric could serve as a convenient proxy for the generalization of more expensive downstream tasks. From now on, we will report the *negative* error for interpretability, and decompose this metric in addition on a per-factor level. We refer to these generalization scores as GS-OOD1, GS-OOD2-sim, and GS-OOD2-real accordingly.

The focus of Dittadi et al. (2021) was to scale VAE-based learning approaches to more realistic scenarios and study the generalization of these simple downstream tasks, with a particular emphasis on disentanglement. Building upon their contributions, we can leverage the broader potential of this robotic setup with many more OOD2 scenarios to study our research questions: To what extent can agents generalize under distribution shift? Do they generalize in different ways depending on the type of distribution shift (including sim-to-real)? Can we predict the OOD generalization of downstream agents from properties of the pretrained representations such as the GS metrics from Dittadi et al. (2021)?

3 STUDY DESIGN

Robotic setup. Our setup is based on TriFinger (Wüthrich et al., 2020) and consists of a robotic finger with three joints that can be controlled to manipulate an object (e.g. a cube) in a bowl-shaped stage. The agent receives a camera observation consistent with the images in Dittadi et al. (2021) and outputs a three-dimensional action. During training, which always happens in simulation, the agent only observes a cube of four possible colors, randomly sampled at every episode (see Fig. 1, step 2).

Distribution shifts. After training, we evaluate these agents in 7 environments: (1) the training environment, which is the typical setting in RL, (2) the OOD1 setting with cube colors that are OOD for the agent but still in-distribution for the encoder, (3) the more challenging OOD2-sim setting where the colors are also OOD for the encoder, (4-6) the OOD2 settings where the object colors are as in the 3 previous settings but the cube is replaced by a sphere, an unseen shape, (7) the OOD2-real setting, where we evaluate zero-shot sim-to-real transfer on the real robotic platform.

Tasks. We begin our study with the *object reaching* downstream control task, where the agent has to reach an object placed at an *arbitrary* random position in the arena. This is significantly more challenging than directly predicting the ground-truth factors, as the agent has to learn to reach the cube by acting on the joints, with a scalar reward as the only learning signal. Consequently, the compute required to learn this task is about 1,000 times greater than in the simple factor prediction case. We additionally include in our study a *pushing* task which consists of pushing an object to a goal position that is sampled at each episode. Learning this task takes one order of magnitude more compute than *object reaching*, likely due to the complex rigid-body dynamics and object interactions. To the best of our knowledge, this is the most challenging manipulation task that is currently feasible on our setup. Ahmed et al. (2021) report solving a similar pushing task, but require the full ground-truth state to be observable.

Training the RL agents. The inputs at time t are the camera observation o_t and a vector of observable variables x_t containing the joint angles and velocities, as well as the target object position in *pushing*. We then feed the camera observation o_t into an encoder e that was pretrained on the dataset in Dittadi et al. (2021). The result is concatenated with x_t, yielding a state vector $s_t = [x_t, e(o_t)]$. We then use SAC to train the policy with s_t as input. The policy, value, and Q networks are implemented as MLPs with 2 hidden layers of size 256. When training the policies, we keep the encoder frozen.

Model sweep. To shed light on the research questions outlined in the previous sections, we perform a large-scale study in which we train 240 representation models and 11,520 downstream policies, as described below. See Appendix A for further implementation details.

- We train 120 β-VAEs (Higgins et al., 2017a) and 120 Ada-GVAEs (Locatello et al., 2020) 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 over $\{0, 50000\}$ steps, with and without input noise, and 10 random seeds per configuration. The latent space size is fixed to 10 following prior work (Chen et al., 2018; Kim & Mnih, 2018; Locatello et al., 2020; Träuble et al., 2021).

- For *object reaching*, we train 20 downstream policies (varying random seed) for each of the 240 VAEs. The resulting 4,800 policies are trained for 400k steps (approximately 2,400 episodes).
- Since the *pushing* task takes substantially longer to train, we limit the number of policies 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 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.

Limitations of our study. Although we aim to provide a sound and extensive empirical study, such studies are inevitably computationally demanding. Thus, we found it necessary to make certain design choices. For each of these choices, we attempted to follow common practice, in order to maintain our study as relevant, general, and useful as possible. One such decision is that of focusing on autoencoder-based representations. To answer our questions on the effect of upstream representations on the generalization of downstream policies, we need a diverse range of representations. How these representations are obtained is not directly relevant to answer our research question. Following Dittadi et al. (2021), we chose to focus on β-VAE and Ada-GVAE models, as they were shown to provide a broad set of representations, including fully disentangled ones. Although we conjecture that other classes of representation learning algorithms should generally reveal similar trends as those found in our study, this is undoubtedly an interesting extension. As for the RL algorithm used in this work, SAC is known to be a particularly sample-efficient model-free RL method that is a popular choice in robotics (Haarnoja et al., 2018a; Kiran et al., 2021; Singh et al., 2019). Extensive results on pushing from ground-truth features on the same setup in Ahmed et al. (2021) indicate that methods like TD3 (Fujimoto et al., 2018) or PPO (Schulman et al., 2017) perform very similarly to SAC under the same reward structure and observation space. Thus, we expect the results of our study to hold beyond SAC. Another interesting direction is the study of additional regularization schemes on the policy network, an aspect that is often overlooked in RL. We expect the potential insights from extending the study along these axes to not justify the additional compute costs and corresponding carbon footprint. However, with improving efficiency and decreasing costs, we believe these could become worthwhile validation experiments in the future.

4 Results

We discuss our results in three parts: In Section 4.1, we present the training results of our large-scale sweep, and how policy regularization and different properties of the pretrained representations affect in-distribution reward. Section 4.2 gives an extensive account of which metrics of the pretrained representations predict OOD generalization of the agents in simulated environments. Finally, in Section 4.3 we perform a similar evaluation on the real robot, in a zero-shot sim-to-real scenario.

4.1 Results in the training environment

Fig. 2 shows the training curves of all policies for *object reaching* 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 *object reaching*, 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 not smaller than 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 pixels using pretrained representations. In particular, all policies on *object reaching* attain almost perfect scores. Unsurprisingly, the more complex *pushing* task requires significantly more training, and the variance across policies is larger. Nonetheless, almost all policies learn to solve the task satisfactorily.

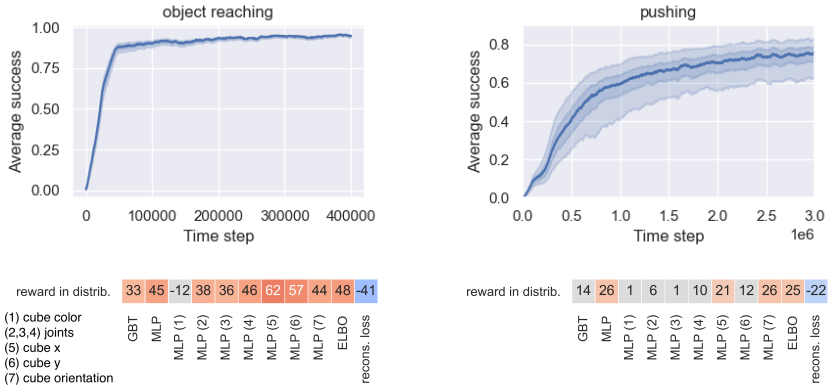

Figure 2: Top: Average training success, aggregated over *all* policies from the sweep (median, quartiles, 5th/95th percentiles). Bottom: Rank correlations between representation metrics and training reward, in the case without policy regularization. Correlations with p<0.05 are color-coded.

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 *object reaching*, the final reward correlates with the ELBO and the reconstruction loss. A simple supervised metric to evaluate a representation is how well a small downstream model can predict the ground-truth factors of variation. Following Dittadi et al. (2021), we use the MLP10000 and GBT10000 metrics (simply MLP and GBT in the following), where MLPs and GBTs are trained to predict the FoVs from 10,000 samples. The 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 architecture could in principle retrieve the FoVs relevant for the task. Interestingly, the correlation with the overall MLP metric mostly stems from the cube pose FoVs, i.e. those that are not included in the ground-truth 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.

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

4.2 Out-of-distribution generalization in simulation

In- and out-of-distribution rewards. In-distribution reward correlates with OOD1 performance on both tasks, especially with regularization, but not with OOD2 performance (see Fig. 3). Moreover, rewards in OOD1 and OOD2 environments are moderately correlated across tasks and regularization settings.

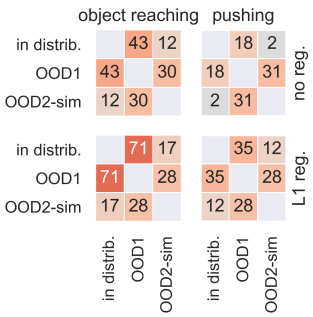

Figure 3: Correlations between training (in distrib.) and OOD rewards (p<0.05).

Unsupervised metrics and informativeness. In Fig. 4 (left) we assess the relation between OOD reward and in-distribution metrics (ELBO, reconstruction loss, 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 encoder 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 4.1, 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

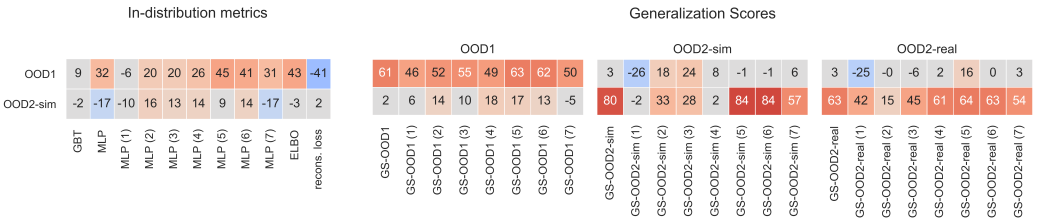

Figure 4: Rank correlations of representation properties with OOD1 and OOD2 reward on *object reaching* without regularization. Numbering when splitting metrics by FoV: (1) cube color; (2–4) joint angles; (5–7) cube position and rotation. Correlations with p<0.05 are color-coded.

results on *pushing* confirm these conclusions although correlations are generally weaker, presumably due to the more complicated nature of this task. An extensive discussion is provided in Appendix B.2.

Correlations with generalization scores. Here we analyze the link between generalization in RL and the generalization scores (GS) discussed in Section 2, which measure the generalization of downstream FoV predictors *out of distribution*, as opposed to the MLP and GBT metrics considered above. 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 *object reaching* task. Reward in the OOD1 setting is significantly correlated with the GS-OOD1 metric of the pretrained 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 measure a representation's generalization score from a few labeled images. This allows for selecting representations that yield more robust downstream policies.

Disentangled representations. Disentanglement has been shown to be helpful for downstream performance and OOD1 generalization even with MLPs (Dittadi et al., 2021). However, in *object reaching*, we only observe a weak correlation with some disentanglement metrics (Fig. 5). In agreement with (Dittadi et al., 2021), disentanglement does not correlate with OOD2 generalization. The same study observed that disentanglement correlates with the informativeness of a representation. To understand if these weak correlations originate from this common confounder, we investigate whether they persist after adjusting for MLP FoV prediction accuracy. Given two representations with similar 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., 2019) and compute the residual reward by subtracting the amount of reward explained by the MLP metric. Fig. 5 shows that this resolves the remaining correlations with disentanglement. Thus, for the RL downstream tasks considered here, disentanglement per se does not seem to be useful for 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. Although regularization has a positive effect on OOD1 and OOD2 generalization in general (Fig. 5, right), we see no link with disentanglement. Furthermore, in accordance with Dittadi et al. (2021), we find that observation noise when training representations is beneficial for OOD2 generalization. See Appendix B.2 for a detailed discussion.

Stronger OOD shifts: evaluating on a novel shape. On *object reaching*, we also test generalization w.r.t. a novel shape by replacing the cube with a sphere. This corresponds to a strong OOD2-type shift, since shape was never varied when training the representations. Surprisingly, the policies appear to be robust to the novel shape. In fact, when the sphere has the same colors that the cube had during policy training, *all* policies get closer than 5 cm to the sphere on average, with a mean success metric of 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%.

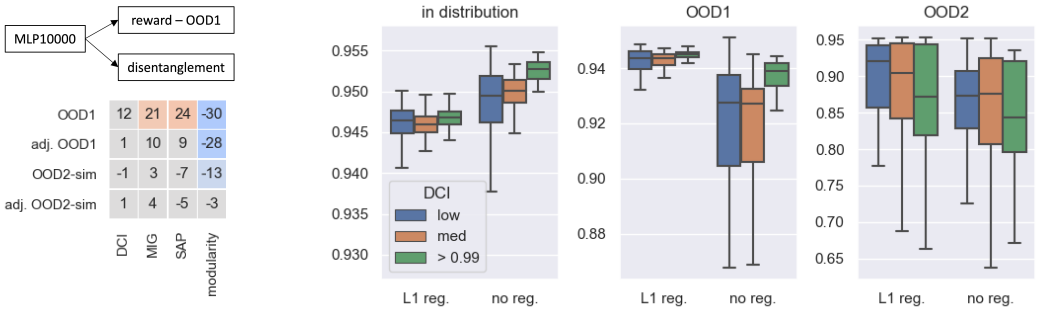

Figure 5: **Box plots**: fractional success on *object reaching* split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. L1 regularization in the first layer of the MLP policy has a positive effect on OOD1 and OOD2 generalization with minimal sacrifice in terms of training reward (see scale). **Correlation matrix** (left): although we observe a mild correlation (color-coded when p<0.05) between some disentanglement metrics and OOD1 (but not OOD2) generalization, this does not hold when adjusting for representation informativeness. We use disentanglement metrics from (Chen et al., 2018; Eastwood & Williams, 2018; Kumar et al., 2018; Ridgeway & Mozer, 2018)

Figure 6: **Zero-shot sim-to-real** on *object reaching* on over 2,000 episodes. **Left:** Rank-correlations on the real platform with a red cube (color-coded when p<0.05). **Middle**: Training encoders with additive noise improves sim-to-real generalization. **Right**: Histogram of fractional success in the more challenging OOD2-real-{green,blue} scenario from 50 policies across 4 different goal positions.

Summary. (1) In- and out-of-distribution rewards are correlated, as long as the representation remains in its training distribution (OOD1). (2) Similarly, in-distribution representation metrics (both unsupervised and supervised) predict OOD1 reward, but are not reliable when the representation is OOD (OOD2). (3) Disentanglement does not correlate with generalization in our experiments, while (4) input noise when training representations is beneficial for OOD2 generalization. (5) Most notably, the GS metrics, which measure generalization under distribution shifts, are significantly correlated with RL performance under similar distribution shifts. We thus recommend using these convenient proxy metrics for selecting representations that will yield robust downstream policies.

4.3 Deploying policies to the real world

We now evaluate a large subset of the agents on the real robot without fine-tuning, quantify their zero-shot sim-to-real generalization, and find metrics that correlate with real-world performance.

Reaching. We choose 960 policies trained in simulation, based on 96 representations and 10 random seeds, and evaluate them on two (randomly chosen, but far apart) goal positions using a red cube. While a red cube was in the training distribution in simulation, we consider this to be OOD2 because real-world images represent a strong distribution shift for the encoder (Dittadi et al., 2021; Djolonga et al., 2020). Although sim-to-real in robotics is considered to be very challenging without domain randomization or fine-tuning (Finn et al., 2017; Rusu et al., 2017; Tobin et al., 2017), many of our trained policies obtain a high fractional success score without resorting to these methods. In addition, in Fig. 6 (left) we observe consistent correlations between zero-shot real-world performance and some of the metrics previously discussed. First, there is a positive correlation with the OOD2-sim reward: Policies that generalize to unseen cube colors in simulation also generalize to the real world.

Second, representations with high GS-OOD2-sim and (especially) GS-OOD2-real scores are promising candidates for sim-to-real transfer. Third, if no labels are available, the weaker correlation with the reconstruction loss on the simulated images can be exploited for representation selection. Finally, as observed by Dittadi et al. (2021) for simple downstream tasks, input noise while learning representations is beneficial for sim-to-real generalization (see Fig. 6, middle).

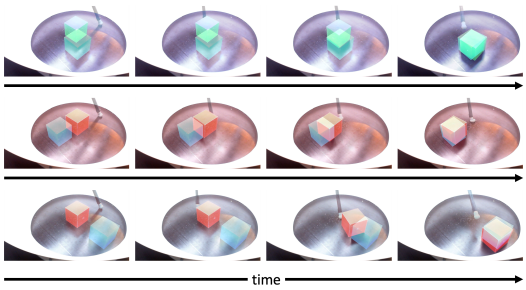

Figure 7: We select pushing policies with high GS-OOD2-real score. When deployed on the real robot without fine-tuning, they succeed in pushing the cube to a specified goal position (transparent blue cube).

Based on these findings, we select 50 policies with a high GS-OOD2-real score, and evaluate them on the real world with a green and a blue cube, which is an even stronger OOD2 distribution shift. In Fig. 6 (right), where metrics are averaged over 4 cube positions per policy, we observe that most policies can still solve the task: approximately 80% of them position the finger less than 5 cm from the cube. Lastly, we repeat the evaluations on the green sphere that we previously performed in simulation, and observe that many policies successfully reach this completely novel object. See Appendix B.3 and the project website for additional results and videos of deployed policies.

Pushing. We now test in practice whether our real-world findings on *object reaching* also hold for *pushing*. We again select policies with a high GS-OOD2-real score and encoders trained with additive input noise. We record episodes on diverse goal positions and cube colors to support our finding that it is also possible to train pushing policies in simulation that generalize to the real robot. In Fig. 7, we depict three representative episodes with successful task completions and refer to the project site for video recordings and further episodes.

Summary. Policies trained in simulation can solve the task on the real robot without domain randomization or fine-tuning. The reconstruction loss, the OOD2 robustness of the underlying encoder, and the OOD2 reward in simulation are all good predictors of real-world performance. For real-world applications, we recommend using GS-OOD2-sim and GS-OOD2-real for model selection, and training the encoder with additive noise.

5 Other related work

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 (Ahmed et al., 2021; Cobbe et al., 2019). 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 sim-to-real 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).

6 Conclusion

Robust out-of-distribution (OOD) generalization is still one of the key open challenges in machine learning. We attempted to answer central questions on the generalization of reinforcement learning agents in a robotics context, and how this is affected by pretrained representations. We presented a large-scale empirical study in which we trained over 10,000 downstream agents given pretrained representations, and extensively tested them under a variety of distribution shifts, including sim-to-real. We observed agents that generalize OOD, and found that some properties of the pretrained representations can be useful to predict which agents will generalize better. We believe this work brings us one step closer to understanding the generalization abilities of learning systems, and we hope that it encourages many further important studies in this direction.

ETHICS STATEMENT

Our study is based on synthetic and real data of a robotic setup where a robotic finger interacts with a cube. Our study does therefore not involve any human subjects leading to discrimination, bias or fairness concerns, or privacy and security issues. Representation learning and generalization is important across many disciplines and applications and could have harmful consequences without humans in the loop in safety-relevant settings. Having a sound understanding of the robustness of a given ML system based on such pretrained representations to distributions shifts is crucial to avoid harmful consequences in potential future high-stake applications to society, such as human-robot interaction (e.g. robotic surgery), autonomous driving, healthcare applications or other fairness-related settings. Here, we investigate a narrow aspect of this, that is, learning arguably harmless manipulation skills like reaching or pushing an object with a simple robotic finger. Our conclusions for OOD generalization are based on this setting and thus cannot be directly transferred to any given application setting of concern.

REPRODUCIBILITY STATEMENT

To make sure our experiments are fully reproducible, we provided a full account of all required setup and implementation details in Section 3 and Appendix A. Code will be shared upon request.

REFERENCES

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.

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.

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.

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

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.

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*, pp. 9448–9458, 2019.

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*, pp. 4055–4065, 2017.

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.

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.

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.

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

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.

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.

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

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

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.

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)*, pp. 512–519. IEEE, 2016.

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.

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

Niklas Funk, Charles Schaff, Rishabh Madan, Takuma Yoneda, Julen Urain De Jesus, Joe Watson, Ethan K. Gordon, Felix Widmaier, Stefan Bauer, Siddhartha S. Srinivasa, Tapomayukh Bhattacharjee, Matthew R. Walter, and Jan Peters. Benchmarking structured policies and policy optimization for real-world dexterous object manipulation. *IEEE Robotics and Automation Letters Special Issue: Robotic Grasping and Manipulation Challenges and Progress*, 2021.

Ali Ghadirzadeh, Atsuto Maki, Danica Kragic, and Mårten Björkman. Deep predictive policy training using reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2351–2358. IEEE, 2017.

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

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

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

Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*, 2018a.

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*, pp. 1861–1870. PMLR, 2018b.

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

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.

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.

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.

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*, pp. 12627–12637, 2019.

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.

Nan Rosemary Ke, Aniket Rajiv Didolkar, Sarthak Mittal, Anirudh Goyal, Guillaume Lajoie, Stefan Bauer, Danilo Jimenez Rezende, Michael Curtis Mozer, Yoshua Bengio, and Christopher Pal. Systematic evaluation of causal discovery in visual model based reinforcement learning, 2021. URL `https://openreview.net/forum?id=gp5Uzbl-9C-`.

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

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

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

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

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

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.

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*, pp. 14611–14624, 2019.

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

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

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.

Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, 2018.

Niklas Pfister, Stefan Bauer, and Jonas Peters. Learning stable and predictive structures in kinetic systems. *Proceedings of the National Academy of Sciences*, 116(51):25405–25411, 2019.

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

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

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

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*, pp. 262–270, 2017.

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.

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

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

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

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

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *arXiv preprint arXiv:1904.07854*, 2019.

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

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

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

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, pp. 1057–1063. Citeseer, 1999.

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)*, pp. 23–30. IEEE, 2017.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. *International Conference on Machine Learning (ICML)*, 2021.

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)*, pp. 3928–3934. IEEE, 2016.

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.

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.

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.

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

A IMPLEMENTATION DETAILS

Task definitions and rewards. We derive both tasks, *object reaching* and *pushing*, from the CausalWorld environments introduced by Ahmed et al. (2021). We pretrain representations on the dataset introduced by Dittadi et al. (2021), and allow only one finger to move in our RL experiments. We introduce the *object reaching* environment that involves an unmovable cube. We used reward structures similar to those in Ahmed et al. (2021):

- *object reaching*: $r_t = -750\left[d(g_t, e_t) - d(g_{t-1}, e_{t-1})\right]$
- *pushing*: $r_t = -750\left[d(o_t, e_t) - d(o_{t-1}, e_{t-1})\right] - 250\left[d(o_t, g_t) - d(o_{t-1}, g_{t-1})\right] + \rho_t$

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 *object reaching* is fixed, i.e. $o_t = g_t$ for all t. The time limit is 2 seconds in *object reaching* and 4 seconds in *pushing*.

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 *object reaching*, 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 very strongly correlated, thus allowing us 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 sampled from the 4 specified training colors from $\mathcal{D}_1$ that correspond 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 over 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 use 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 $5 \cdot 10^{-7}$ over 200,000 time steps in *object reaching*, and from 0 to $6 \cdot 10^{-8}$ over 3,000,000 time steps in *pushing*.

B ADDITIONAL RESULTS

B.1 TRAINING ENVIRONMENT

Fig. 2 in Section 4.1 shows correlations of unsupervised and supervised metrics with in-distribution reward for *object reaching* 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 4.2 we discussed rank-correlations of OOD rewards with unsupervised, informativeness and generalization scores on *object reaching* 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 *object reaching*, 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 L1 regularization. In contrast to our observations without regularization, we find that the correlation between GS-OOD1 and OOD1 generalization of the policy vanishes when adjusting for the MLP metric.

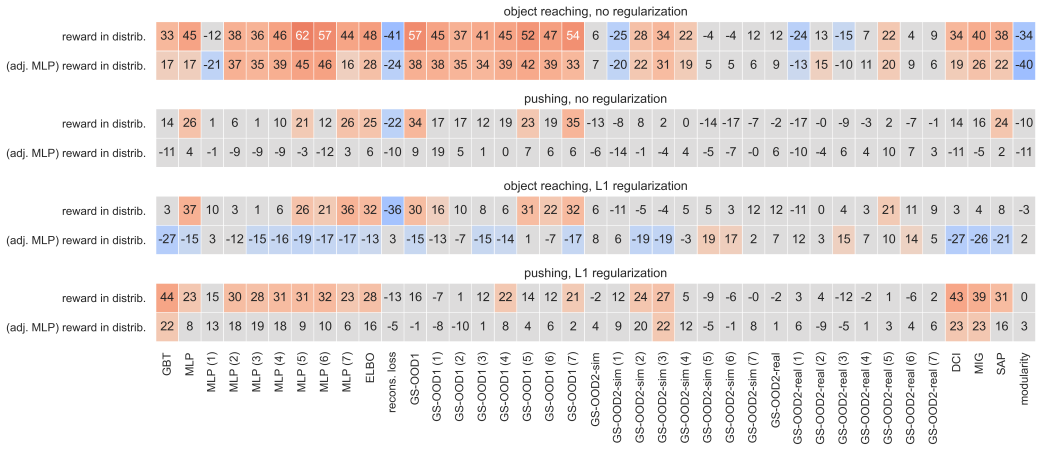

Figure 8: Rank correlations between metrics and in-distribution reward, with and without adjusting for informativeness.

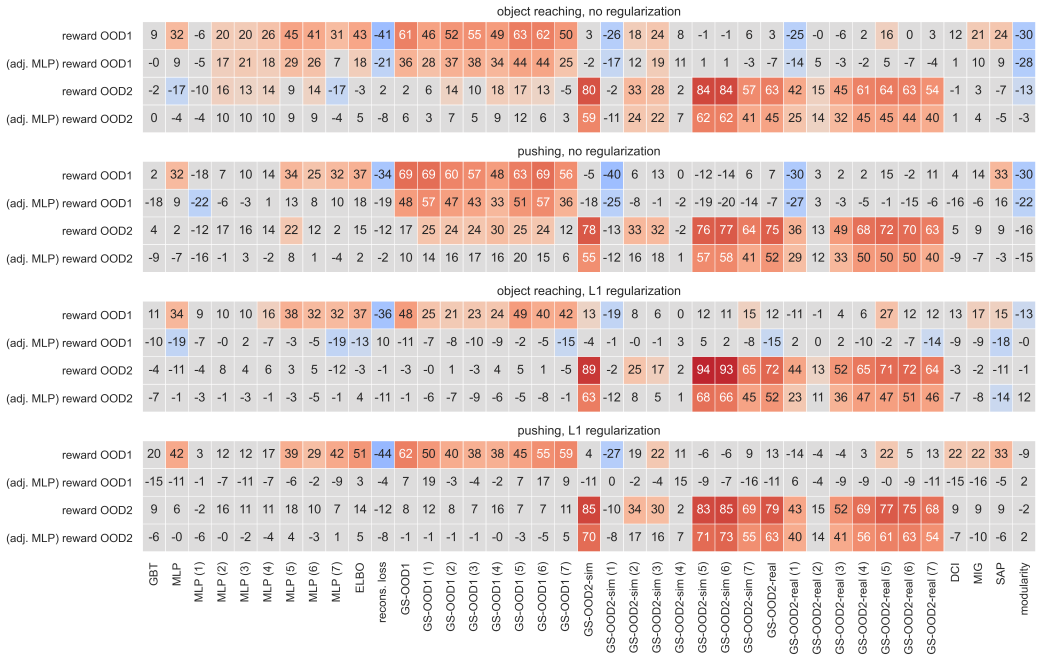

Figure 9: Rank correlations between metrics and OOD reward, with and without adjusting for informativeness.

B.2.1 Disentangled Representations

As discussed in Section 4.2 for *object reaching* 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.

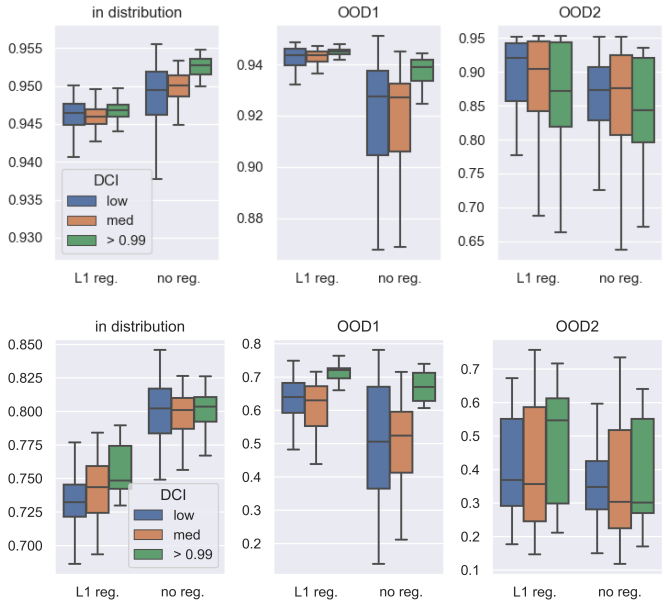

Figure 10: Fractional success on *object reaching* (**top**) and *pushing* (**bottom**), split according to low (blue), medium-high (orange), and almost perfect (green) disentanglement. Results for *object reaching* are also reported in Fig. 5 in Section 4.2.

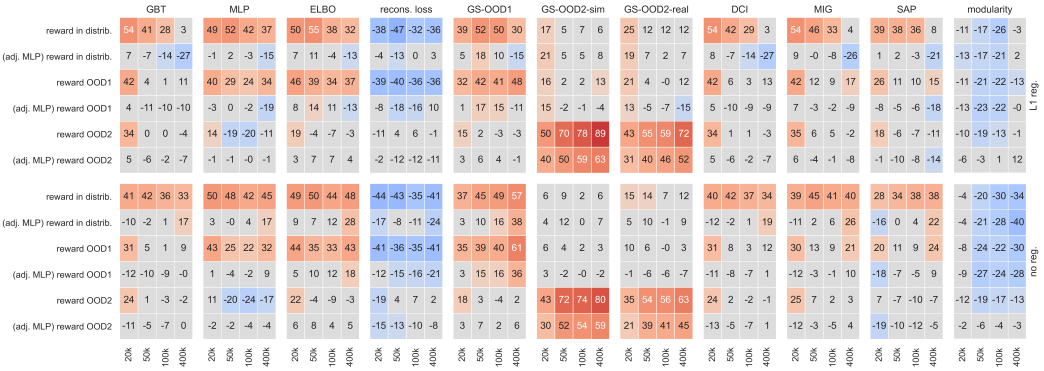

Figure 11: Sample efficiency analysis for *object reaching*. Rank correlations of rewards with relevant metrics along multiple time steps.

B.2.2 Regularization

As seen in Fig. 10, regularization generally has a positive effect on OOD1 and OOD2 generalization, which is particularly prominent in the OOD1 setting. On the other hand, it leads to lower training rewards both in *object reaching* and in *pushing*. In the latter, the performance drop is particularly significant, while in *object reaching* it is negligible.

B.2.3 Sample efficiency

In addition to the analysis reported in the main paper, we investigate how representation properties affect sample efficiency. Specifically, we store checkpoints of our policies at $t \in \{20\text{k}, 50\text{k}, 100\text{k}, 400\text{k}\}$ for *object reaching* and $t \in \{200\text{k}, 500\text{k}, 1\text{M}, 3\text{M}\}$ for *pushing*. We then evaluate policies at these time step on the same three environments as before: (1) on the cube colors from training; (2) on the OOD1 cube colors; and (3) on the OOD2-sim cube colors. Results are summarized in Fig. 11 for *object reaching* and Fig. 12 for *pushing*.

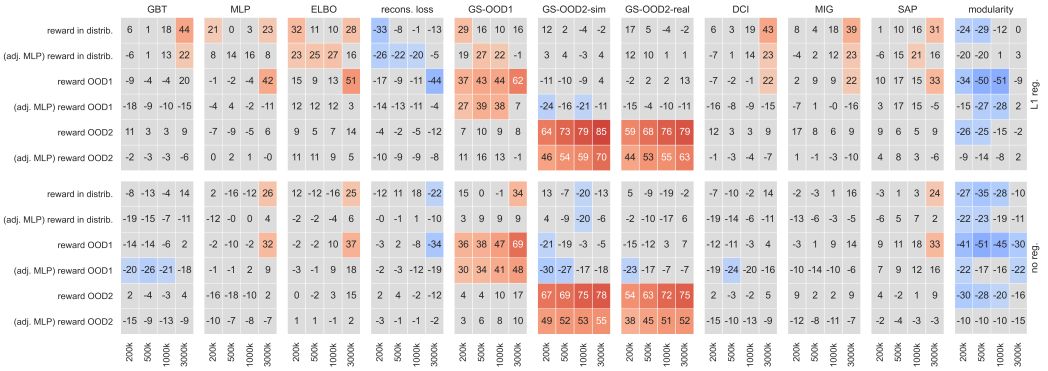

Figure 12: Sample efficiency analysis for *pushing*. Rank correlations of rewards with relevant metrics along multiple time steps.

On *object reaching* (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. Crucially, we see that the generalization scores (GS) are correlated with generalization of the corresponding policies under OOD1 and OOD2 shifts for all recorded time steps, confirming the results in the main text.

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 (GS) and policy generalization under the same distribution shifts remain strong and statistically significant, corroborating the analysis in the main text.

B.2.4 Generalization to a novel shape

As mentioned in Section 4.2, on the *object reaching* task, we also test 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 evaluate 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 5 cm 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%.

B.3 Deploying policies to the real world

In Fig. 14 we depict three representative episodes of testing a reaching policy on the real robot for the strong OOD shift with a novel sphere object shape instead of the cube from training. We present the respective videos in the project page. There we also present videos of additional real-world episodes on pushing and reaching cubes of different colors.

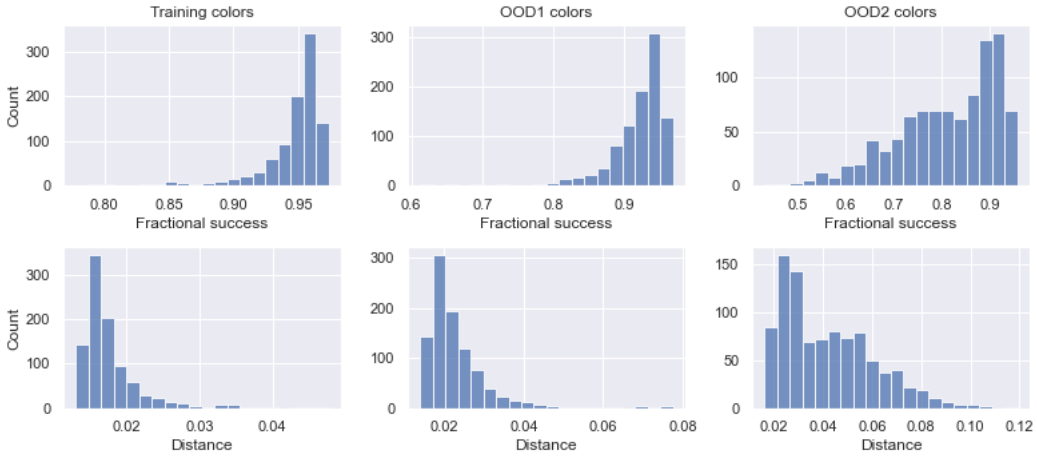

Figure 13: Testing policies for *object reaching* under the same in-distribution, OOD1, and OOD2 evaluation protocols regarding object color in simulation, but replacing the cube with a sphere, which was never used in training.

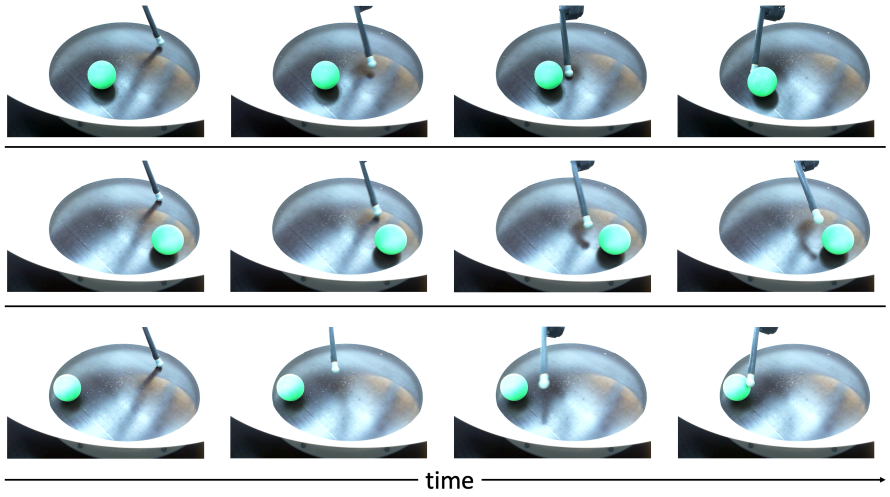

Figure 14: Transferring policies for *object reaching* to the real robot setup without any fine-tuning on a green sphere (unseen shape *and* color).