# Learning Domain Invariant Representations in Goal-conditioned Block MDPs


## Abstract

Deep Reinforcement Learning (RL) is successful in solving many real-world complex Markov Decision Processes (MDPs) problems. However, agents often face unanticipated environmental changes after deployment, especially in robotics. These changes are often spurious and unrelated to the underlying problem, such as background shifts for visual input agents. Unfortunately, Deep RL policies are usually sensitive to these changes and fail to act robustly against them. This resembles the problem of domain generalization in supervised learning. In this work, we study this problem for goal-conditioned RL agents. We propose a theoretical framework in the Block MDP setting that characterizes the generalizability of goal-conditioned policies in a new environment. In addition, we develop a practical method called PA-SkewFit under the framework to enhance domain generalization. The empirical evaluation shows that our goal-conditioned RL agent can behave well in various unseen environments during testing with an overall 50% performance increase and roughly the same self-learning efficiency.


## 1 Introduction

Deep Reinforcement Learning (RL) has achieved remarkable success in solving high-dimensional Markov Decision Processes (MDPs) problems, e.g., Alpha Zero [1] for Go, DQN [2] for Atari games and SAC [3] for locomotion control. However, current RL algorithms requires massive amounts of trial and error to learn [1-3]. They also tend to overfit to specific environments and often fail to generalize beyond the environment they were trained on [4]. Unfortunately, this characteristic limits the applicability of RL algorithms for many real world applications. Deployed RL agents, e.g. robots in the field, will often face environment changes in their input: different weather, lighting conditions or varying objects [5]. Many of these changes are often spurious and unrelated to the underlying task, e.g. control. However, RL agents trained without experiencing these changes are sensitive to the changes and often perform poorly in practice [5-7].

In our work, we seek to tackle changing, diverse problems with goal-conditioned RL agents. Goal-conditioned Reinforcement Learning is a popular research topic as its formulation and method is practical for many robot learning problems [8, 9]. In goal-conditioned MDPs, the agent has to achieve a desired goal state $g$ which is sampled from a prior distribution. The agent should be able to achieve not only the training goals but also new test-time goals. Moreover, in practice, goal-conditioned RL agents often receive high-dimensional inputs for both observations and goals [10, 11]. Thus, it is important to ensure that the behaviour of goal-conditioned RL agents is invariant to any irrelevant environmental changes in the input at test time. Previous work [6] tries to address these problems via model bi-simulation metric [12]. These methods aim to acquire a minimal representation which is invariant to irrelevant environment factors. However, as goal-conditioned MDPs are a family of MDPs indexed by the goals, it is inefficient for these methods to acquire the model bi-simulation representation for every possible goal, especially in high-dimensional continuous goal spaces (such as images).

In our work, we instead choose to optimize a surrogate objective to learn the invariant policy. Our main contributions are:

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

1. We formulate the Goal-conditioned Block MDPs (GBMDPs) to study domain generalization in the goal-conditioned reinforcement learning setting (Section 2), and propose a general theory characterizing how well a policy generalizes to unseen environments (Section 3.1).
2. We propose a theoretically-motivated algorithm based on optimizing a surrogate objective, *perfect alignment*, for learning domain invariant representations (Section 3.2). We then describe a practical implementation via *aligned data* to achieve the objective (Section 3.3).
3. Empirically, our experiments for a sawyer arm robot simulation with visual observations and goals demonstrates that our proposed method achieves state-of-the-art performance compared to data augmentation and bi-simulation baselines at generalizing to unseen test environments in goal-conditioned tasks (Section 4).

## 2 Problem Formulation

In this section, we formulate the domain invariant learning problem as solving Goal-conditioned Block MDPs (GBMDPs). This extends previous work on learning invariances [6, 13] to the goal-conditioned setting [14, 15, 8].

We consider a family of Goal-conditioned Block MDP environments $M^{\mathcal{E}} = \{(\mathcal{S}, \mathcal{A}, \mathcal{X}^e, \mathcal{T}^e, \mathcal{G}, \gamma) | e \in \mathcal{E}\}$ where $e$ stands for the environment index. Each environment consists of shared state space $\mathcal{S}$, shared action space $\mathcal{A}$, observation space $\mathcal{X}^e$, transition dynamic $\mathcal{T}^e$, shared goal space $\mathcal{G} \subset \mathcal{S}$ and the discount factor $\gamma$.

Moreover, we assume that $M^{\mathcal{E}}$ follows the generalized Block structure [6]. The observation $x^e \in \mathcal{X}^e$ is determined by state $s \in \mathcal{S}$ and the environmental factor $b^e \in \mathcal{B}^e$, i.e., $x^e(s, b^e)$ (Figure 1). For brevity, we use $x^e_t(s)$ to denote the observation for domain $e$ at state $s$ and step $t$. We may also omit $t$ as $x^e(s)$ if we do not emphasize on the step $t$ or the exact environmental factor $b^e_t$. The transition function is thus consists of state transition $p(s_{t+1}|s_t, a_t)$ and environmental factor transition $q^e(b^e_{t+1}|b^e_t)$. In our work, we assume the state transition is nearly deterministic which is quite common in most RL benchmarks and applications [2, 16, 17].

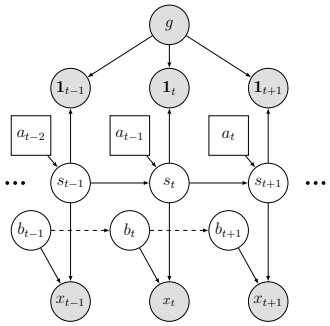



Figure 1: Graphical model for Goal-conditioned Block MDPs (GBMDPs) setting. The agent takes in the goal $g$ and observation $x_t$, which is produced by the domain invariant state $s_t$ and environmental state $b_t$, and acts with action $a_t$. Note that $b_t$ may have temporal dependence indicated by the dashed edge.

Most importantly, $\mathcal{X} = \cup_{e \in \mathcal{E}} \mathcal{X}^e$ satisfies the disjoint property [13], i.e., each observation $x \in \mathcal{X}$ uniquely determines its underlying state $s$. Thus, the observation space $\mathcal{X}$ can be partitioned into disjoint blocks $\mathcal{X}(s), s \in \mathcal{S}$. This assumption prevents the partial observation problem.

The objective function in GBMDP is to learn a goal-conditioned policy $\pi(a|x^e, g)$ that maximizes the discounted state density function $J(\pi)$ [9] across all domains $e \in \mathcal{E}$. In our theoretical analysis, we do not assume the exact form of $g$ to the policy. You can regard $\pi(\cdot|x^e, g)$ as a group of RL policies indexed by the goal state $g$.

$$J(\pi) = \mathbb{E}_{e \sim \mathcal{E}, g \sim \mathcal{G}, \pi} \left[ (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p^e_\pi(s_t = g | s_0, g) \right] = \mathbb{E}_{e \sim \mathcal{E}}[J^e(\pi)] \tag{1}$$

$p^e_\pi(s_t = g|s_0, g)$ denotes the probability of achieving goal $g$ under policy $\pi(\cdot|x^e, g)$ at step $t$ in domain $e$. Besides $e \sim \mathcal{E}$ and $g \sim \mathcal{G}$ refers to uniform samples from each set. As $p^e_\pi$ is defined over state space, it may differs among environments since policy $\pi$ takes $x^e$ as input. Fortunately, in an GBMDP, there exist optimal policies $\pi_G(\cdot|x^e, g)$ which are invariant over all environments, i.e., $\pi_G(a|x^e(s), g) = \pi_G(a|x^{e'}(s), g), \forall a \in \mathcal{A}, s \in \mathcal{S}, e, e' \in \mathcal{E}$.

During training, the agent has access to training environments $\{e_i\}_{i=1}^N = \mathcal{E}_{\text{train}} \subset \mathcal{E}$ with their environment indices. However, we do not assume that $\mathcal{E}_{\text{train}}$ is i.i.d sampled from $\mathcal{E}$. Thus, we want the goal-conditioned RL agent acquires the ability to neglect the spurious and unrelated environmental factor $b^e$ and capture the underlying invariant state information. This setup is adopted in many recent literature such as in [6] and in domain generalization [18, 19] of supervised learning.

# 3 Method

In this section, we propose a novel learning algorithm to solve GBMDPs. First, we propose a general theory to characterize how well a policy $\pi$ generalizes to a unseen test environment after training on $\mathcal{E}_{\text{train}}$. Then, we introduce *perfect alignment* as a surrogate objective for learning. This objective is supported by the generalization theory. Finally, we propose a practical method to acquire perfect alignment.

## 3.1 Domain Generalization Theory for GBMDP

In a seminal work, Ben-David et al. [20] show it is possible to bound the error of a classifier trained on a source domain on a target domain with a different data distribution. Follow-up work extends the theory to the domain generalization setting [21, 22]. In GBMDP, we can also derive similar theory to characterize the generalization from training environments $\mathcal{E}_{\text{train}}$ to target test environment $t$. The theory relies on the Total Variation Distance $D_{\text{TV}}$ [23].

$D_{\text{TV}}$ of two policies $\pi_1, \pi_2$ with input $(x^e, g)$ is defined as follows.

$$D_{\text{TV}}(\pi_1(\cdot|x^e, g) \parallel \pi_2(\cdot|x^e, g)) = \sup_{A' \subset \mathcal{A}} |\pi_1(A'|x^e, g) - \pi_2(A'|x^e, g)|$$

In the following statements, we denote $\rho(x, g)$ as some joint distributions of goals and observations that $g \sim \mathcal{G}$ and $x$ is determined by $\rho(x|g)$. Additionally, we use $\rho^e_\pi(x^e|g)$ to denote the occupancy measure of $x^e$ in environment $e$ under policy $\pi(\cdot|x^e, g)$ and also $\rho^e_\pi(x^e)$ as the marginal distribution of $\rho^e_\pi(x^e, g)$.

To begin with, we denote $\epsilon^{\rho(x,g)}(\pi_1 \parallel \pi_2)$ as the average $D_{\text{TV}}$ between $\pi_1$ and $\pi_2$, i.e., $\epsilon^{\rho(x,g)}(\pi_1 \parallel \pi_2) = \mathbb{E}_{\rho(x,g)}[D_{\text{TV}}(\pi_1(\cdot|x, g) \parallel \pi_2(\cdot|x, g))]$. This quantity is crucial in our theory as it can characterize the performance gap between two policies (see Appendix A).

Then, similar to the famous $\mathcal{H}\Delta\mathcal{H}$-divergence [20, 21] in domain adaptation theory, we define $\Pi\Delta\Pi$-divergence of two joint distributions $\rho(x, g)$ and $\rho(x, g)'$ in terms of the policy class $\Pi$:

$$d_{\Pi\Delta\Pi}(\rho(x, g), \rho(x, g)') = \sup_{\pi,\pi' \in \Pi} |\epsilon^{\rho(x,g)}(\pi \parallel \pi') - \epsilon^{\rho(x,g)'}(\pi \parallel \pi')|$$

On one hand, $d_{\Pi\Delta\Pi}$ is a distance metric which reflects the distance between two distributions w.r.t function class $\Pi$. On the other hand, if we fix these two distributions, it also reveals the quality of the function class $\Pi$, i.e., smaller $d_{\Pi\Delta\Pi}$ means more invariant to the distribution change.

Finally, we state the following Proposition in which $\pi_G$ is some optimal and invariant policy.[1]

**Proposition 1** (Informal)**.** *For any $\pi \in \Pi$, we consider the occupancy measure $\{\rho^{e_i}_\pi(x^{e_i}, g)\}_{i=1}^N$ for training environments and $\rho^t_{\pi_G}(x^t, g)$ for the target environment. For simplicity, we use $\epsilon^{e_i}$ as the abbreviation of $\epsilon^{\rho^{e_i}_\pi(x^{e_i},g)}$, $\epsilon^t$ as $\epsilon^{\rho^t_{\pi_G}(x^t,g)}$ and $\delta = \max_{e_i,e'_i \in \mathcal{E}_{train}} d_{\Pi\Delta\Pi}(\rho^{e_i}_\pi(x^{e_i}, g), \rho^{e'_i}_\pi(x^{e'_i}, g))$. Considering*

$$\lambda = \frac{1}{N}\sum_{i=1}^N \epsilon^{e_i}(\pi^* \parallel \pi_G) + \epsilon^t(\pi^* \parallel \pi_G), \quad \pi^* = \arg\min_{\pi' \in \Pi} \sum_{i=1}^N \epsilon^{e_i}(\pi' \parallel \pi_G)$$

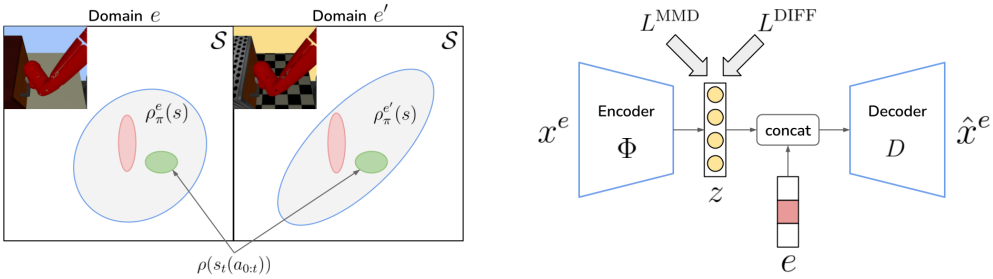
*Then, we have*

$$J^t(\pi_G) - J^t(\pi) \leq \frac{1}{N}\sum_{i=1}^N \epsilon^{e_i}(\pi \parallel \pi_G) + \lambda + \delta + \min_{\rho(x,g) \in B} d_{\Pi\Delta\Pi}(\rho(x, g), \rho^t_{\pi_G}(x^t, g)) \tag{2}$$

*where $B$ is a characteristic set of joint distributions determined by $\mathcal{E}_{train}$ and policy class $\Pi$.*

The formal statement and the proof are shown in Appendix A. Generally speaking, the first term of the right hand side in Eq. (2) quantifies the performance of $\pi$ in the $N$ training environments. $\lambda$ quantifies the optimality of the policy class $\Pi$ over all environments. $\delta$ reflects how the policy class $\Pi$ can reflect the difference among $\{\rho^{e_i}_\pi(x^{e_i}, g), e_i \in \mathcal{E}_{\text{train}}\}$, which should be small if the policy

[1] If there are multiple optimal policies, we can choose one that minimizes the left hand side of Eq. (2).




(a) Illustration of Aligned Sampling

(b) Overall structure

Figure 2: **(a)**: Illustration of Aligned Sampling. Square represents the whole state space $\mathcal{S}$, gray area represents the distribution $\rho_\pi^e(s)$ in two different environments. Small colored areas are the aligned state distribution generated by aligned sampling in Section 3.2. **(b)**: Overall VAE structure in our PA-SF. Encoder maps $x^e$ to the latent embedding $z$ and decoder $D$ reconstructs the observations with $z$ and index $e$. $L^{\text{MMD}}$ and $L^{\text{DIFF}}$ denote the two losses in Section 3.2.

class is invariant. The last term characterizes the distance between training environment and target environment and will be small if the training environments are diversely distributed.

Many works on domain generalization of supervised learning [20, 24, 21, 22, 25] spend much effort in discussing the trade-offs among different terms similar to the ones in Eq. (2), e.g., minimizing $\delta$ may increase $\lambda$ [25], and in developing sophisticated techniques to optimize the bound, e.g. distribution matching [26–28] or adversarial learning [24].

Different from their perspectives, in GBMDPs, we propose a simple but effective criteria to minimize the bound. From now on, we only consider the policy class $\Pi = \Pi_\Phi = \{w \circ (\Phi(x), g), \forall w\}$. Usually, $\Phi$ will be referred as an encoder which maps $x \in \mathcal{X}$ to some latent representation $z = \Phi(x)$. We will also use the notation $z(s) = \Phi(x(s))$ if we do not emphasize on the specific environment.

**Definition 1** (**Perfect Alignment**). *An encoder is called a* perfect alignment *encoder $\Phi$ w.r.t environment set $E$ if $\forall e, e' \in E$ and $\forall s, s' \in \mathcal{S}$, $\Phi(x^e(s)) = \Phi(x^{e'}(s'))$ if and only if $s = s'$.*

As illustrated in Figure 5, an encoder is in perfect alignment if it maps two observations of the same invariant state $s$ to the same latent encoding $z(s)$ while also preventing meaningless embedding, i.e., mapping observations of different states to the same $z$. We believe perfect alignment plays an important role in domain generalization for goal-conditioned RL agents. Specifically, it can minimize the bound of Eq. (2) as follows.

**Proposition 2.** *If the encoder $\Phi$ is a perfect alignment over $\mathcal{E}_{train}$, then*

$$J^t(\pi_G) - J^t(\pi) \leq \underbrace{\frac{1}{N}\sum_{i=1}^{N} \epsilon^{e_i}(\pi \parallel \pi_G)}_{(E)} + \delta + \underbrace{\epsilon^t(\pi^* \parallel \pi_G) + d_{\Pi_\Phi \Delta \Pi_\Phi}(\tilde{\rho}(x, g), \rho^t_{\pi_G}(x^t, g))}_{(t)} \quad (3)$$

*where $\tilde{\rho}(x, g)$ denotes the minimal in $B$ as Eq. (2). Besides, under mild assumptions, if $(E) \to 0$ then $\delta \to 0$.*

Theoretically speaking, when $\Phi$ is a perfect alignment encoder, an goal-conditioned RL policy trained over the encoded space $\{z(s)\}_{s \in \mathcal{S}}$ will minimize $(E)$ to 0. As a consequence, the only term left in Eq. (3) is $(t)$ when training converges.

Though it is hard to conduct task-agnostic analysis on $(t)$ term, we believe that perfect alignment is still an adequate criteria in minimizing $(t)$ term. For example, if $\Phi$ is a perfect alignment encoder over the diverse $\mathcal{E}_{\text{train}}$ and the environment factor of $t$ is similar as the ones in training , it is reasonable to assume that $\Phi(x^t(s)) \approx \Phi(x^{e_i}(s))$, i.e., $\Phi$ is a perfect alignment encoder over both $\mathcal{E}_{\text{train}}$ and $t$. Consequently, we have $J^t(\pi) \to J^t(\pi_G)$ when $(E) \to 0$. Please refer to Appendix A for more discussions. In the following sections, we will focus on acquiring a perfect alignment encoder $\Phi$ as our surrogate objective.

### 3.2 Learning Domain Invariant via Aligned Sampling

First, we discuss about the *if* condition on perfect alignment encoder $\Phi$, i.e., $\forall s, \Phi(x^e(s)) = \Phi(x^{e'}(s))$. The proposed method is based on *aligned sampling*.

Most RL algorithms use observation-dependent sampling from the environment, e.g., $\epsilon$-greedy or Gaussian distribution policies [3, 29, 30, 2]. However, with observation-dependent sampling, occupancy measures $\rho^e_\pi(s), \forall e \in \mathcal{E}_{\text{train}}$s will be different. Thus, simply aligning the latent representation of these observations will fail to produce a perfect alignment encoder $\Phi$.

Thus, we propose a novel strategy for data collection called *aligned sampling*. First, we randomly select a trajectory (e.g., from replay buffer etc.), denoted as $\{x^e_0, a_0, x^e_1, a_1, \ldots, x_T\}$ from environment $e$. The set of correspondent states along this trajectory are denoted as $\{s^e_t(a_{0:t})\}^T_{t=0}$. Second, we take the same action sequence $a_{0:T}$ in another domain $e'$ to get another trajectory $\{x^{e'}_0, a_0, x^{e'}_1, a_1, \ldots, x^{e'}_T\}$ (so as $\{s^{e'}_t(a_{0:t})\}^T_{t=0}$). We refer to the data collected by aligned sampling from all training environments as aligned data. These aligned observations $\{x^{e_i}_t(a_{0:t})\}, \forall e_i \in \mathcal{E}_{\text{train}}$ are stored in a aligned buffer $\mathcal{R}_{\text{align}}$ corresponding to the aligned action sequence $a_{0:t}$.

Under the definition of GBMDP, we have $\forall t \in [0 : T], s \in \mathcal{S}$, $\rho^e(s_t(a_{0:t})) = \rho^{e'}(s_t(a_{0:t}))$. Therefore, we use MMD loss [31] to match distribution of $\Phi(x^e(s))$ for the aligned data. To be specific, in each iteration, we sample a mini-batch of $B$ aligned observations of every training environment $e_i \in \mathcal{E}_{\text{train}}$ from $\mathcal{R}_{\text{align}}$, i.e., $\mathcal{B}_{\text{align}} = \{x^{e_i}(s_t(a^b_{0:t})), \forall e_i \in \mathcal{E}_{\text{train}}\}^B_{b=1}$. Then we use the following loss as a computationally efficient approximate of the MMD metric[32, 26].

$$L^{\text{MMD}}(\Phi) = \mathbb{E}_{e,e' \sim \mathcal{E}_{\text{train}}, \mathcal{B}_{\text{align}} \sim \mathcal{R}_{\text{align}}}[\| \frac{1}{B}\sum_{b=1}^{B} \psi(\Phi(x^e(s_t(a^b_{0:t}))) - \frac{1}{B}\sum_{b=1}^{B} \psi(\Phi(x^{e'}(s_t(a^b_{0:t})))) \|^2_2]$$

where $\psi$ is a random expansion function. The following Proposition justifies that $L^{\text{MMD}}(\Phi)$ is an effective indicator.

**Proposition 3.** *Under mild assumptions, if* $\mathbb{E}_{e,e' \sim \mathcal{E}_{train}, \mathcal{B}_{align} \sim \mathcal{R}_{align}}[\| \Phi(x^e(s)) - \Phi(x^{e'}(s)) \|^2_2] \geq \delta$, *then* $L^{MMD}(\Phi) \geq O(\delta)$.

In Figure 2(a), we illustrate the intuition of the above approach. When the transition is nearly deterministic, the entropy for $\rho(s_t(a_{0:t}))$ is much smaller, i.e., $\mathcal{H}(\rho(s_t(a_{0:t}))) \ll \mathcal{H}(\rho^e_\pi(s_t))$. Thus, $\rho(s_t(a_{0:t}))$ can be regard as small patches in $\mathcal{S}$ (Figure 2(a)). We use the MMD loss $L^{\text{MMD}}$ to match the latent representation $\{\Phi(x^e(s)), s \sim \rho(s_t(a_{0:t}))\}, \forall e \in \mathcal{E}_{\text{train}}$ together. As a consequence, we should achieve an encoder $\Phi$ that is more aligned.

However, simply minimizing $L^{\text{MMD}}$ may violate the *only if* condition for perfect alignment. For example, a trivial solution for $L^{\text{MMD}} = 0$ is mapping all observations to some constant latent. To ensure that $\Phi(x^e(s)) = \Phi(x^{e'}(s'))$ *only if* $s = s'$, we additionally use the difference loss $L^{\text{DIFF}}$ as follows.

$$L^{\text{DIFF}}(\Phi) = -\mathbb{E}_{e \sim \mathcal{E}_{\text{train}}, x^e, \tilde{x}^e \in \mathcal{R}^e} \| \Phi(x^e) - \Phi(\tilde{x}^e) \|^2_2$$

where $\mathcal{R}^e$ refers to the replay buffer of environment $e$. Clearly, minimizing $L^{\text{DIFF}}$ encourages dispersed latent representations over all states $s \in \mathcal{S}$.

We refer the combination $L^{\text{MMD}} + \alpha L^{\text{DIFF}}$ as our perfect alignment loss $L^{\text{PA}}$. Note that $L^{\text{PA}}$ resembles contrastive learning [33, 34]. Namely, observations of aligned data from $\mathcal{R}_{\text{align}}$ are positive pairs while observations sampled randomly from a big replay buffer are negative pairs. We match the latent embedding of positive pairs via the MMD loss while separate negative pairs by the difference loss. As discussed in Section 3.1, we believe this latent representation will improve generalization to unseen target environments.

### 3.3 Perfect Alignment for Skew-Fit

In Section 4, we will train goal-conditioned RL agents with perfect alignment encoder using the Skew-Fit algorithm [30]. Skew-Fit is typically designed for visual-input agents which learn a goal-conditioned policy via purely self-supervised learning.

First, Skew-Fit trains a $\beta$-VAE with observations collected online to acquire a compact and meaningful latent representation for each state, i.e., $z(s)$ from the image observations $x(s)$. Then, Skew-Fit optimizes a SAC [3] agent in the goal-conditioned setting over the latent embedding of the image observation and goal, $\pi(a|z, g)$. The reward function is the negative of $l_2$ distance between the two latent representation $z(s)$ and $z(g)$, i.e., $r(s, g) = - \parallel z(s) - z(g) \parallel_2^2$. Furthermore, to improve sample efficiency, Skew-Fit proposes skewed sampling for goal-conditioned exploration.

In our algorithm, *Perfect Alignment for Skew-Fit* (PA-SF), the encoder $\Phi$ is optimized via both $\beta$-VAE losses as [30, 35] and $L^{\text{PA}}$ loss to ensure meaningful and perfectly aligned latent representation.

$$L(\Phi, D) = L^{\text{RECON}}(x^e, \hat{x^e}) + \beta D_{\text{KL}}(q_\Phi(z|x^e) \parallel p(z)) + \alpha_{\text{MMD}} L^{\text{MMD}} + \alpha_{\text{DIFF}} L^{\text{DIFF}} \tag{4}$$

Besides, we use both aligned sampling and observation-dependent sampling. Aligned sampling provides aligned data but hurts sample-efficiency while observation-dependent sampling is exploration-efficient but fails to ensure alignment. In practice, we find that collecting a small portion ($15\%$ of all data collected) of aligned data in $\mathcal{R}_{\text{align}}$ is enough for perfect alignment via $L^{\text{PA}}$.

Additionally, inspired by [26], we also change the $\beta$-VAE structure to what is shown in Figure 2(b), since in GBMDPdata are collected from $N$ training environments and thus, identity Gaussian distribution is no longer a proper fit for prior. The encoder $\Phi$ maps $x^e(s)$ to some latent representation $z(s)$ while the decoder $D$ takes both $z(s)$ and the environment index $e$ as input to reconstruct $\hat{x^e}(s)$.

Note that by using both $L^{\text{PA}}$ and $L^{\text{RECON}}$, we require static environmental factors in $\mathcal{E}_{\text{train}}$ (unnecessary for testing environments) for a stable optimization. In future work, we will address the limit from $\beta$-VAE by training two latent representations simultaneously to stabilize the optimization for generality.

# 4 Experiments

In this section, we conduct experiments to evaluate our PA-SF algorithms. The experiments are based on multiworld [17]. Our empirical analysis tries to answer the following questions: (1) How well does PA-SF perform in solving GBMDP problems? (2) How does each component proposed in Section 3 contribute to the performance?

## 4.1 Comparative Evaluation

In this subsection, we aim to answer the question (1) by comparing our proposed PA-SF method with vanilla Skew-Fit and several other baselines that attempt to acquire invariant policies for RL agents.

**Baselines** Current methods for obtaining robust policies can be characterized into two categories: (1) data augmentation and (2) model bi-simulation.

1. **Data Augmentation**. Recent work [36] tries to use data augmentation to prevent the RL agents from distractions. We implement the most widely accepted data augmentation methods RAD [37] upon Skew-Fit (Skew-Fit + RAD) as a baseline. Note that our PA-SF method does not use any data augmentation and is parallel with this kind of techniques.
2. **Model Bi-simulation** [12]. These methods utilize bi-simulation metrics to learn a minimal but sufficient representation which will neglect irrelevant features of Block MDPs. We include MISA [6] and DBC [7] in our comparison as they are the most successful implementations for high-dimensional tasks. Moreover, in the goal-conditioned setting, we use an oracle state-goal distance $- \parallel s - g \parallel_2$ as rewards for these two algorithms in GBMDP. In contrast, our PA-SF method does not have such information.

**Environments** We evaluate PA-SF and all baselines on a set of GBMDP tasks based on multiworld benchmark [17], which is widely used to evaluate the performance of visual input goal-conditioned algorithms. We use the following four basic tasks [35, 30]: *Reach*, *Door*, *Pickup* and *Push*. In GBMDP, we create different environments with various backgrounds, desk surfaces, and object appearances. During testing, in addition unseen static backgrounds, we also create environments with unseen video backgrounds to mimic environmental factor transitions $q^e(b^e_{t+1}|b^e_t)$. This makes policy generalization more challenging. Please refer to Appendix D for a full desciption of our experiment setup and implementation details of the baselines and our algorithm.

Table 1: Evaluation of PA-SF and baselines on four control tasks. We report the mean and one standard deviation on each task (lower metric is better).

| | Algorithm | Reach | Door | Push | Pickup |
|---|---|---|---|---|---|
| | | Hand distance (35K) | Angle difference (150K) | Puck distance (400K) | Object distance (250K) |
| Test Avg | Skew-Fit | $0.111 \pm 0.010$ | $0.194 \pm 0.018$ | $0.083 \pm 0.004$ | $0.036 \pm 0.003$ |
| | Skew-Fit + RAD | $0.152 \pm 0.030$ | $0.229 \pm 0.061$ | $0.094 \pm 0.006$ | $0.039 \pm 0.002$ |
| | MISA | $0.239 \pm 0.0142$ | $0.255 \pm 0.027$ | $0.099 \pm 0.006$ | $0.042 \pm 0.003$ |
| | DBC | $0.185 \pm 0.037$ | $0.320 \pm 0.033$ | $0.095 \pm 0.006$ | $0.041 \pm 0.003$ |
| | PA-SF(**Ours**) | $\mathbf{0.076} \pm 0.005$ | $\mathbf{0.106} \pm 0.015$ | $\mathbf{0.069} \pm 0.005$ | $\mathbf{0.030} \pm 0.004$ |
| Train Avg | PA-SF(**Ours**) | $0.067 \pm 0.005$ | $0.058 \pm 0.074$ | $0.060 \pm 0.005$ | $0.020 \pm 0.002$ |
| | Oracle Skew-Fit | $0.055 \pm 0.010$ | $0.057 \pm 0.012$ | $0.054 \pm 0.006$ | $0.020 \pm 0.006$ |

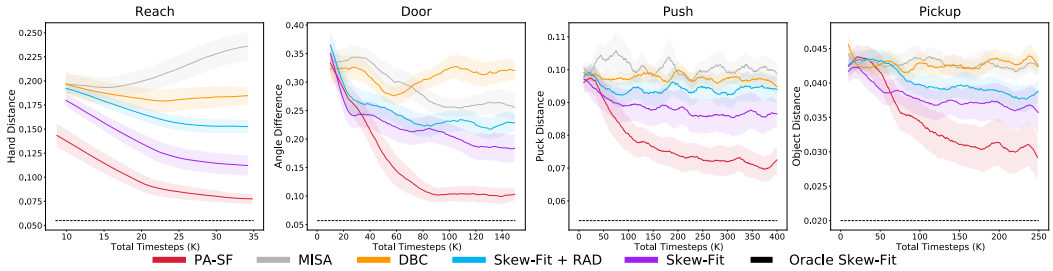



Figure 3: Learning curve of all algorithms on averaged across test environments for each task. All curves are drawn with the mean and one standard deviation of 7 seeds.

**Results** In Table 1, we show the final average performance of each algorithm on *unseen* test environments $\mathcal{E}_{\text{test}}$. The corresponding learning curves are shown in Figure 3. This metric shows the generalizability of each RL agent. All these agents are allowed to collect data from $\mathcal{E}_{\text{train}}$ ($N = 3$) with static environmental factors. Our PA-SF achieves SOTA performance on all tasks. On testing environments, we achieve a relative reduction around 40% to 65% of the corresponding metrics over vanilla Skew-Fit w.r.t the optimal metric possible (Oracle Skew-Fit) Oracle Skew-Fit refers to the performance of a Skew-Fit algorithm trained directly on the single environment (and *not* simultaneously on all $\mathcal{E}_{\text{train}}$).

Other invariant policy learning methods perform sluggishly on all tasks. For DBC and MISA, we hypothesize that they struggle for goal-conditioned problem since the model bi-simulation metric is defined for a single MDP. In GBMDPs, this means acquiring a set of encoders $\Phi_g$ that achieves model bi-simulation for every possible $g$ and is thus inefficient for learning. By design, our method is not susceptible to this issue as the perfect alignment is a universal invariant representation for all goals. For RAD, we suspect that it is incompatible with the VAE-encoder in Skew-Fit. This suggests a future research problem in developing data augmentation algorithms compatible with goal-conditioned RL agents.

In addition, we also show the performance of PA-SF on the training environments in Table 1. PA-SF is still comparable and as sample-efficient as Skew-Fit in the training environments.

### 4.2 Design Evaluation

In this subsection, we conduct comprehensive analysis on the design of PA-SF to interpret how well it carries out the theoretical framework discussed in Section 3.1 and Section 3.2.

To begin with, we show the learning curves in Figure 4 of different ablations of PA-SF in the *Door* environment during both training and testing. To understand the roles of $\mathcal{L}^{\text{DIFF}}$ and $L^{\text{MMD}}$, PA-SF (w/o D) excludes $L^{\text{DIFF}}$ and PA-SF (w/o MD) excludes both losses[2]. Noticing that PA-SF (w/o MD) is equivalent to the Skew-Fit algorithm with our proposed VAE structure (Figure 2(b)). We also add PA-SF (w/o AS) which excludes aligned sampling.

[2] A single $L^{\text{DIFF}}$ is not useful here.

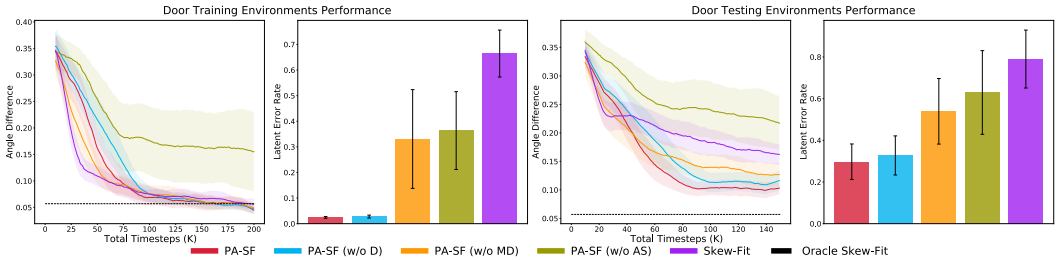



Figure 4: Ablation of our design and visualization of the latent representation via LER metric. All curves represent the mean and one standard deviation across 7 seeds.

Additionally, we also quantify the quality of the latent representation $\Phi(x^e(s))$ in Fig.. The metric *Latent Err Rate* (LER) is defined as the average over environment set $\hat{\mathcal{E}} \in \{\mathcal{E}_{\text{train}}, \mathcal{E}_{\text{test}}\}$ as follows:

$$\text{Err}(\Phi) = \mathbb{E}_{e\sim\hat{\mathcal{E}}, s\sim\mathcal{S}} \left[ \frac{\| \Phi(x^e(s)) - \Phi(x^{e_0}(s)) \|_2}{\| \Phi(x^e(s) \|_2} \right]$$

In general, the smaller $\text{Err}(\Phi)$ is, the closer the encoder $\Phi$ is to perfect alignment over the environments $\hat{\mathcal{E}}$. We first focus on the discussion about training performance.

1. $\Phi$ achieves perfect alignment over $\mathcal{E}_{\text{train}}$ via $L^{\text{MMD}}$ as the LER value of PA-SF and PA-SF (w/o D) is almost 0. While without MMD loss, PA-SF (w/o MD) and Skew-Fit struggle with large LER value despite achieving good training performance. The comparison between PA-SF and PA-SF (w/o AS) demonstrates the importance of using aligned data in the MMD loss (Otherwise, the matching is inherently erroneous). Also, the only-if condition, i.e., $\Phi(x^e(s)) = \Phi(x^{e'}(s'))$ only if $s = s'$, is satisfied by visualizing the reconstruction of the VAE, we believe this is ensured by both the difference loss and the reconstruction loss.
2. Policy $\pi$ converges to near optimal performance on training environments, i.e. comparable with Oracle Skew-Fit. This indicates that the $(E)$ term in Proposition 2 asymptotically achieves 0 as expected. Together with the previous argument, we can conclude that $(E) + \delta$ is near 0 in Eq. (3) on Door environment.

Second, we focus on the generalization performance on target domains $t$, i.e., term $(t)$ in Eq. (3). We observe the following:

1. As shown by the learning curve of test environments, the target domain performance of different ablations match that of the LER metric: SkewFit, PA-SF (w/o AS) > PA-SF (w/o MD) > PA-SF (w/o D) > PA-SF. During training, these ablations have almost the same performance. This indicates that the increased test performance indeed comes from the improved representation quality of the encoder $\Phi$, i.e., more aligned.
2. As is minimized by $L^{\text{MMD}}$, the LER of testing environments is reduced significantly, as is the evaluation distance metric. This supports our claim that a perfect alignment encoder on training environments encourages generalization to unseen test environments. In addition, by encouraging dispersed latent representation, the difference loss $L^{\text{DIFF}}$ also plays a role in reducing LER during testing. This supports the necessity of both losses for generalization.

We observe the similar results in other tasks as well (Appendix C). Here, we also visualize the latent space by t-SNE plot to illustrate the perfect alignment on task *Push*. Dots in training environments are matched perfectly and the corresponding test environment dot is approximately near as expected.

## 5 Related Work

**Goal-conditioned RL:** Goal-conditioned RL [14, 15] removes the need for complicated reward shaping by only rewarding agents for reaching a desired set of goal states. RIG [35] is the seminal work for visual-input, reward-free learning in goal-conditioned MDPs. Skew-Fit [30] improves over RIG [35] in training efficiency by ensuring the behavioural goals used to explore are diverse and have wide state coverage. However, Skew-Fit has its own limitation in understanding the semantic meaning of the goal-conditioned task. To acquire more meaningful goal's and observation's latent representation, several approaches apply inductive biases or seek human feedback. ROLL [38] applies object extraction methods under strong assumptions, while WSC [39] uses weak binary

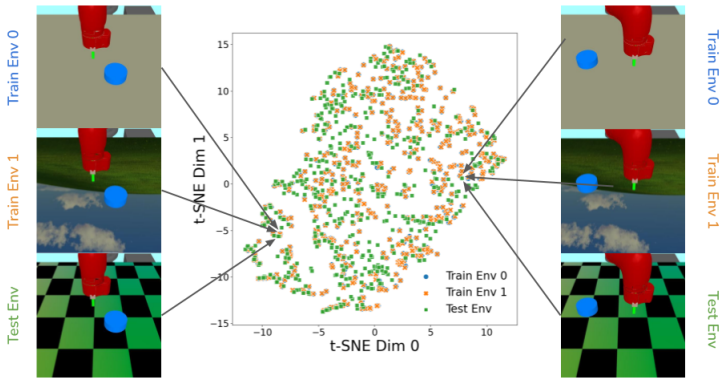



Figure 5: t-SNE visualization of the latent space $\Phi(x^e)$ trained with our method for three environments: 2 training and 1 testing of *Push* as well as instances visualization.

labeled data as the reward function. Others explore the same goal-conditioned RL problem via hindsight experience replay [8, 40, 41], unsupervised reward learning [11], inverse dynamics models [10], C-learning [9], goal generation [42-44], goal-conditioned forward models [45], and hierarchical RL [46-49]. Our study focus on learning goal-conditioned policies that is invariant of spurious environmental factors. We aim to learn a policy that can generalize to visual goals in unseen test environments.

**Learning Invariants in RL:** Robustness to domain shifts is crucial for real-world applications of RL. [6, 7, 50] implement the model-bisimulation framework [12] to acquire a minimal but sufficient representation for solving the MDP problem. However, model-bisimulation for high-dimension problems typically requires domain-invariant and dense rewards. These assumptions do not hold in GBMDPs. Contrastive Metric Embeddings (CME) [51] instead uses $\pi^*$-bisimulation metric but it also requires extra information of the optimal policy. Another line of work tries to address these issues via self-supervised learning. [36] tests multiple data augmentation methods including RAD [37] and DrQ [52] to boost the robustness of the representation as well as the policy. Our work can also apply data augmentation in practice. However, we find that RAD is not very helpful in the Skew-Fit framework. Additionally, [53, 54] use self-supervised correction during real-world adaptation like sim2real transfer but these methods are incompatible for domain generalization.

## 6 Conclusion

In this paper, we study the problem of learning invariant policies in Goal-conditioned RL agents. The problem is formally formulated as a GBMDP, which is an extension of Goal-conditioned MDPs and Block MDPs where we want the agent's policy to generalize to unseen test environments after training on several training environments.

As supported by the generalization bound for GBMDP, we propose a simple but effective heuristic, i.e., perfect alignment which we can minimize the bound asymptotically and benefit the generalization. To learn a perfect alignment encoder, we propose a practical method based on aligned sampling. The method resembles contrastive learning: matching latent representation of aligned data via MMD loss and dispersing the whole latent representations via the DIFF loss. Finally, we propose a practical implementation Perfect Alignment for Skew-Fit (PA-SF) by adding the perfect alignment loss to Skew-Fit and changing the VAE structure to handle GBMDPs.

The empirical evaluation shows that our method is the SOTA algorithm and achieves a remarkable increase in test environments' performance over other methods. We also compare our algorithm with several ablations and analyze the representation quantitatively. The results support our claims in the theoretical analysis that perfect alignment criteria is effective and that we can effectively optimize the criteria with our proposed method. We believe the perfect alignment criteria will enable applications in diverse problem settings and offers interesting directions for future work, such as extensions to other goal-conditioned learning frameworks [9, 10].

# References


[1] David Silver, Julian Schrittwieser, Karen Simonyan, Aj Antonoglou, Ioannis abd Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550, 2017.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg strovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan umaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518, 2015.

[3] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[4] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *CoRR*, abs/1810.12282, 2018. URL http://arxiv.org/abs/1810.12282.

[5] Ryan Julian, Benjamin Swanson, Gaurav S Sukhatme, Sergey Levine, Chelsea Finn, and Karol Hausman. Efficient adaptation for end-to-end vision-based robotic manipulation. *arXiv preprint arXiv:2004.10190*, 2020.

[6] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pages 11214–11224. PMLR, 2020.

[7] 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.

[8] Andrychowicz Marcin, Wolski Filip, Ray Alex, Schneider Jonas, Fong Rachel, Welinder Peter, McGrew Bob, Tobin Josh, Abbeel Pieter, and Zaremba Wojciech. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 2017.

[9] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*, 2020.

[10] Keiran Paster, Sheila A McIlraith, and Jimmy Ba. Planning from pixels using inverse dynamics models. *arXiv preprint arXiv:2012.02419*, 2020.

[11] Alexandre Péré, Sébastien Forestier, Olivier Sigaud, and Pierre-Yves Oudeyer. Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations*, 2018.

[12] Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

[13] Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

[14] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, pages 1094–1099. Citeseer, 1993.

[15] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

[16] Brockman Greg, Cheung Vicki, Pettersson Ludwig, Schneider Jonas, Schulman John, Tang Jie, and Zaremba Wojciech. Openai gym, 2016.

[17] Vitchyr Pong, Murtaza Dalal, Steven Lin, and Ashvin Nair. multiworld. https://github.com/vitchyr/multiworld, 2018.

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

[19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[20] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1): 151–175, 2010.

[21] Anthony Sicilia, Xingchen Zhao, and Seong Jae Hwang. Domain adversarial neural networks for domain generalization: When it works and how to improve. *arXiv preprint arXiv:2102.03924*, 2021.

[22] Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2019.

[23] Wikipedia. Total variation distance of probability measures — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Total%20variation%20distance%20of%20probability%20measures&oldid=1024556616, 2021. [Online; accessed 24-May-2021].

[24] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pages 4013–4022. PMLR, 2019.

[25] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 315–331. Springer, 2019.

[26] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S Zemel. The variational fair autoencoder. In *ICLR*, 2016.

[27] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018.

[28] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Feature alignment and restoration for domain generalization and adaptation. *arXiv preprint arXiv:2006.12009*, 2020.

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

[30] Vitchyr Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. In *International Conference on Machine Learning*, pages 7783–7792. PMLR, 2020.

[31] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel method for the two-sample problem. *Journal of Machine Learning Research*, 1:1–10, 2008.

[32] Ji Zhao and Deyu Meng. Fastmmd: Ensemble of circular discrepancy for efficient two-sample test. *Neural computation*, 27(6):1345–1372, 2015.

[33] 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.

[34] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[35] Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9209–9220, 2018.

[36] Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

[37] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020.

[38] Yufei Wang, Gautham Narayan Narasimhan, Xingyu Lin, Brian Okorn, and David Held. Roll: Visual self-supervised reinforcement learning with object reasoning. *arXiv preprint arXiv:2011.06777*, 2020.

[39] Lisa Lee, Benjamin Eysenbach, Ruslan Salakhutdinov, Chelsea Finn, et al. Weakly-supervised reinforcement learning for controllable behavior. *arXiv preprint arXiv:2004.02860*, 2020.

[40] Zhizhou Ren, Kefan Dong, Yuan Zhou, Qiang Liu, and Jian Peng. Exploration via hindsight goal generation. *Advances in Neural Information Processing Systems*, 32:13485–13496, 2019.

[41] Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. *arXiv e-prints*, pages arXiv–1912, 2019.

[42] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pages 1515–1528. PMLR, 2018.

[43] Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *International Conference on Learning Representations*, 2019.

[44] Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International Conference on Machine Learning*, pages 7750–7761. PMLR, 2020.

[45] Suraj Nair, Silvio Savarese, and Chelsea Finn. Goal-aware prediction: Learning to model what matters. In *International Conference on Machine Learning*, pages 7207–7219. PMLR, 2020.

[46] Siyuan Li, Lulu Zheng, Jianhao Wang, and Chongjie Zhang. Learning subgoal representation with slow dynamics.

[47] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Near-optimal representation learning for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2018.

[48] Lunjun Zhang, Ge Yang, and Bradly C Stadie. World model as a graph: Learning latent landmarks for planning. *arXiv preprint arXiv:2011.12491*, 2020.

[49] Zhimin Hou, Jiajun Fei, Yuelin Deng, and Jing Xu. Data-efficient hierarchical reinforcement learning for robotic assembly control applications. *IEEE Transactions on Industrial Electronics*, 2020.

[50] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

[51] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

[52] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[53] Nicklas Hansen, Yu Sun, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020.

[54] Cristian Bodnar, Karol Hausman, Gabriel Dulac-Arnold, and Rico Jonschkowski. A geometric perspective on self-supervised policy adaptation. *arXiv preprint arXiv:2011.07318*, 2020.

# Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See appendix.
   (b) Did you include complete proofs of all theoretical results? [Yes] See appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited the multiworld environment [17], which we extended, in Section 4.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See citation
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] . We used a simulated robotics environment with no crowdsourcing nor human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]