ON THE DATA-EFFICIENCY WITH CONTRASTIVE IMAGE TRANSFORMATION IN REINFORCEMENT LEARNING

Sicong Liu[1 2 3*] **Xi Sheryl Zhang**[2 3 5†] **Yushuo Li**[2] **Yifan Zhang**[2 3 5] **Jian Cheng**[2 3 4]
[1]NJUST, [2]Institute of Automation, Chinese Academy of Sciences,
[3]AIRIA, [4]School of Future Technology, University of Chinese Academy of Sciences,
[5]University of Chinese Academy of Sciences, Nanjing
{sicongliu1014;sheryl.zhangxi}@gmail.com,
yushuo.li@ia.ac.cn, {yfzhang;jcheng}@nlpr.ia.ac.cn

ABSTRACT

Data-efficiency has always been an essential issue in pixel-based reinforcement learning (RL). As the agent not only learns decision-making but also meaningful representations from images. The line of reinforcement learning with data augmentation shows significant improvements in sample-efficiency. However, it is challenging to guarantee the optimality invariant transformation, that is, the augmented data are readily recognized as a completely different state by the agent. In the end, we propose a contrastive invariant transformation (CoIT), a simple yet promising learnable data augmentation combined with standard model-free algorithms to improve sample-efficiency. Concretely, the differentiable CoIT leverages original samples with augmented samples and hastens the state encoder for a contrastive invariant embedding. We evaluate our approach on DeepMind Control Suite and Atari100K. Empirical results verify advances using CoIT, enabling it to outperform the new state-of-the-art on various tasks. Source code is available at https://github.com/mooricAnna/CoIT.

1 INTRODUCTION

Improving data-efficiency to accomplish sequential decisions has always been a crucial problem in pixel-based reinforcement learning. As the agent has to learn an optimal policy with a meaningful information abstraction from observations parallel. Unlike supervised representation learning with strong supervised high-dimensional signals, the training process in RL is fragile. It could be harmful to the training process and cause performance degradation consequently using inappropriate manners. Hence, it is an urgent request to seek subtle representation learning methods for visual RL.

Previous works have been proposed in the literature to demonstrate that introducing auxiliary loss functions such as pixel reconstruction (Yarats et al., 2019) and contrastive learning (Laskin et al., 2020b) alleviates this issue. In particular, data augmentations have already proven beneficial to data-efficiency. RAD (Laskin et al., 2020a) performs an extension of experiments and widely analyzes the impact of various techniques in data augmentation. DrQ (Yarats et al., 2020) and DrQ-v2 (Yarats et al., 2021) make use of appropriate image augmentation with great success. Also, previous works have carried out the potential of data augmentation in terms of generalization (Hansen et al., 2021; Raileanu et al., 2020; Zhang & Guo, 2021; Hansen & Wang, 2021; Fan et al., 2021).

Despite the mentioned efforts, it is pretty hard to guarantee that the augmented representations are sufficiently diverse yet semantically consistent. To this end, we explore the underlying condition for representation learning in RL. It is rational to hypothesize that there is an optimal transformation enabling an encoder to abstract informative latent space. This line of works belongs to the regime of state abstraction (Du et al., 2019; Zhang et al., 2020b; Tomar et al., 2021; Wang et al., 2022), which derives from grouping similar world-states for descriptions of the environment (Dietterich, 2000; Andre & Russell, 2002; Castro & Precup, 2010). Inspired by spatial transformer networks (STN)

*Work done during Institute of Automation, Chinese Academy of Sciences internship.
†Corresponding Author.

(Jaderberg et al., 2015), a data augmentation model in the vision domain, we consider that merging the parameterized transformation with visual RL could be beneficial. The designed transformation not only discovers the optimality of state abstraction but also produces diverse virtual samples for the agent. To do so, we enforce a learnable data augmentation that updates its parameters along with the RL objective.

To understand parameterized augmentation and its relation to representation learning in RL, we focus on fundamental data manipulation by generating augmented data from a learnable Gaussian distribution. To be clear, we present the image transformation to control the margin of the augmentation under an RL training-friendly data distribution. Since changed data distributions meanwhile being controlled by learning algorithms would be helpful in high-dimensional cases (Balestriero et al., 2021). Here we raise our idea:

Can we parameterize the data augmentation by sampling from a dynamic distribution to obtain a training-friendly state distribution along with model-free RL?

In light of this challenge, we present a contrastive invariant transformation (CoIT), a novel contrastive learning to ameliorate the data-efficiency for visual RL. CoIT integrates a learnable transformation for model-free methods with minimal modification to the architecture and training pipeline. Specifically, we parameterize the mean and variance of a Gaussian distribution for transforming data and update the parameters together with RL by using constraints to urge faster algorithm convergence empirically. As the learning goes on, the agent approximates the `TRANSFORM` distribution that is optimal for the task at hand to solve the task. In addition, we evaluate CoIT on DeepMind Control Suite and Atari100K, and experimental results demonstrate that the learnable transformation outperforms the current SOTA methods. Besides, our method does not claim any custom architecture choices and is essential for reproducing end-to-end training. Based on these results, we demonstrate that a learnable transformation improves data-efficiency effectively for visual RL.

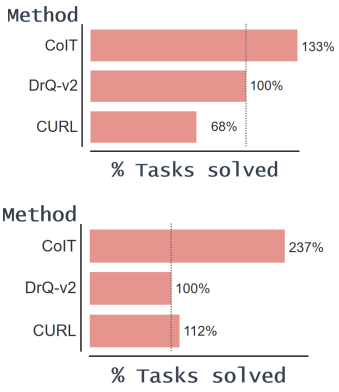

Figure 1: Percentage (%) of score solved in the DMC. We set the score of DrQ-v2 as 100% and report the result of CoIT and CURL: **Up**: in 500K steps. **Bottom**: in 100K steps. The task is solved when return nearly reaches the upper bound.

Key Contributions: (i) We present CoIT, a simple yet effective framework with a learnable image transformation that integrates invariant representations with model-free RL to improve data-efficiency. (ii) We propose a theoretical analysis of how our method can approximate a stationary distribution over the transformed data by the optimal invariant metric, thus learning better representations. (iii) We evaluate CoIT on popular benchmarks and show that our method outperforms previous state-of-the-art methods on data-efficiency and stability.

2 Related Work

Several concurrent methods have been proposed for improving data-efficiency whose common ingredients containing data augmentation and self-supervised learning are listed.

Data augmentation in RL. Like the success of data augmentation in computer vision (Zhong et al., 2020; DeVries & Taylor, 2017; Yun et al., 2019; Zhang et al., 2017), these methods have played a key role in improving the data-efficiency of visual RL problems Mnih et al. (2013); Yarats et al. (2019); Hafner et al. (2019); Lee et al. (2019). RAD (Laskin et al., 2020a) conducted mounts of experiments and finds out that different data augmentations lead to entirely different results. It provides a broader perspective for the follow-up study of data augmentation in RL. DrQ (Yarats et al., 2020) proposed an effective augmentation method called *random shift* and introduced a regularization term for Q-learning. Based on DrQ, the DrQ-v2 (Yarats et al., 2021) conducted minimal changes and demonstrated that merely a simple augmentation method could match the state-of-the-art model-based algorithm on data-efficiency and performance.

Self-supervised learning in RL. Motivated by the breakthrough in self-supervised learning (Chen et al., 2020; He et al., 2020; Caron et al., 2020; Grill et al., 2020), it is natural to combine these

methods with visual RL to learn rich representations. CURL (Laskin et al., 2020b) introduces a framework similar to SimCLR (che) into visual RL. CoBERL (Banino et al., 2021) also tried to minimize the consistency between positive samples by semantic-preserving data augmentation. Besides, STDIM (Mazoure et al., 2020) and PI-SAC (Lee et al., 2020) maximize the temporal mutual information (MI) between the nearby states. SPR (Schwarzer et al., 2020) and PlayVirtual (Yu et al., 2021) follow their idea, but they utilize the *dynamics model* to predict nearby states in latent space. DBC (Zhang et al., 2020b) and PSM (Agarwal et al., 2021) focus on learning task-relevant information. They utilize signals in the environment to achieve an invariant representation learning and thereby generalize the agent to unseen environments.

3 PRELIMINARIES

3.1 REINFORCEMENT LEARNING FROM OBSERVATIONS

Visual RL control is formulated as an infinite-horizon Markov Decision Process (MDP) (Bellman, 1957), as the observations can not fully describe the underlying state. To address this problem, we stack multiple consecutive frames together to represent the current underlying state $\mathbf{s}$ (Mnih et al., 2013). In this mind, the MDP $\mathcal{M}$ is a 5-tuple $\langle \mathcal{O}, \mathcal{S}, \mathcal{A}, r, \gamma \rangle$. Here, the observation space $\mathcal{O}$ generally consists of multiple-stack frames. The state space $\mathcal{S}$ is either observable or unobservable (Silver et al., 2017; Zhang et al., 2020a). The agent uses observations $\mathcal{O}$ to sample actions from the action space $\mathcal{A}$. Every time the agent interacts with the environment, it obtains a reward r. The end goal is to train an agent to maximize the cumulative reward $\mathcal{R}$. The policy evaluation used as estimating the performance of the policy π_ϕ is normally defined by rewards in infinite-horizon tasks,

$$\mathbb{E}[\mathcal{R}] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \,\middle|\, \pi_\phi\right]. \tag{1}$$

where $\gamma \in [0, 1)$ is the discount factor and r_t denotes the reward at time t.

3.2 Q LEARNING

The state-action value function Q_θ is trained by minimizing the Bellman error to estimate the cumulative reward at the current state:

$$J_\theta(\mathcal{D}) = \mathbb{E}_{e\sim\mathcal{D}}[(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - r_t - \gamma Q_{\bar{\theta}}(\mathbf{s}_{t+1}, \pi_\phi(\mathbf{s}_{t+1})))^2] \tag{2}$$

where e is a transition from the replay buffer $\mathcal{D}$. And $\bar{\theta}$ denotes an exponential moving average of θ.

For the continuous control tasks, we utilize an actor-critic algorithm called Deep Deterministic Policy Gradient (DDPG) (Silver et al., 2014; Lillicrap et al., 2015) which consists of the aforementioned state-action value function Q_θ and a deterministic policy π_ϕ. The policy π_ϕ aims at maximizing $J_\phi(\mathcal{D}) = \mathbb{E}_\mathcal{D}[Q_\theta(\mathbf{s}_t, \pi_\phi(\mathbf{s}_t)]$. Various effective improvements have also been lead to DDPG. The Q-learning process incorporates n-step returns (Watkins, 1989; Peng & Williams, 1994). The scheduled exploration noise is produced by a linear decay $\tilde{\sigma}(t)$ for the variance $\tilde{\sigma}^2$ which provides different levels of exploration at different training steps: $\tilde{\sigma}(t) = \tilde{\sigma}_{\text{init}} + (1 - \min(t/T, 1))(\tilde{\sigma}_{\text{final}} - \tilde{\sigma}_{\text{init}})$. The initial and final value for standard deviation are defined by $\tilde{\sigma}_{\text{init}}$ and $\tilde{\sigma}_{\text{final}}$, and the decay horizon T is related to the total training steps of the environment.

For the discrete control, we use the data-efficient Rainbow DQN (Van Hasselt et al., 2019) which applied multiple improvements on top of the original Nature DQN (Mnih et al., 2015).

3.3 STATE ABSTRACTION

While visual RL has achieved many successes in simulated tasks, it remains challenging to learn robust representations from real vision, where images reveal detailed scenes of a complex and unstructured world (Zhang et al., 2020b; Agarwal et al., 2021; Wang et al., 2022). Therefore, abstracting meaningful elements from the visual scene to present the underlying state is significantly important for visual RL.

We follow the Block Markov Decision Process (BMDP) (Du et al., 2019), which refers to episodic learning tasks via an unobservable latent space $\mathcal{S}$ and an observable context space $\mathcal{X}$. The environment generates a context by $\mathbf{x} \sim p(\cdot|\mathbf{s})$. They present a fundamental assumption as: each observation $\mathbf{x}$

uniquely determines its generating state $\mathbf{s}$. Similarly, in model-free RL without modeling dynamics, the manipulated context $\mathbf{x}$ can be conditioned on a certain probability given an environment transition $\mathbf{e}$ which is $\mathbf{x} \sim p(\cdot|\mathbf{e})$.

4 THE COIT

4.1 LEARNABLE INVARIANT TRANSFORMATION

Following the motivation of smoothing training experiences to stabilize the target Q network (Mnih et al., 2013; Lillicrap et al., 2015), the transformed $\mathbf{x}'$ is required to satisfy $\mathbf{x}' \sim p(\cdot|\mathbf{e})$, where environment transition $\mathbf{e}$ is ideally in the replay distribution $\mathcal{D}$. Note that $\mathbf{e}$ is a random variables. Formally, we are ready to introduce the optimal invariant metric to reach the stationary distribution $\mathcal{D}$ over the augmented context $\mathbf{x}'$, through the definition,

Definition 4.1. (Optimal Invariant Metric). Given a transition distribution $\mathcal{D}$ for tuples in the replay buffer, suppose the block structure assumption holds, the shift between transitions $\mathbf{x}$ and its context $\mathbf{x}'$ can be measured by a conditional divergence,

$$d(\mathbf{x}, \mathbf{x}'|\mathbf{e}) \triangleq \mathbb{E}_{e\sim\mathcal{D}}\left[d_{KL}\left(p(\mathbf{x}|\mathbf{e}=e)||p(\mathbf{x}'|\mathbf{e}=e)\right)\right] = \int_{\mathbf{e}} d_{KL}\left(p(\mathbf{x}|\mathbf{e}=e)||p(\mathbf{x}'|\mathbf{e}=e)\right) dp(\mathbf{e}) \quad (3)$$

where $d_{KL}(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence. It indicates the expected distance between $\mathbf{x}$ and $\mathbf{x}'$ conditioning on $\mathbf{e}$. One may argue that the dynamics cannot be assumed as a fixed distribution when it comes to new observations, especially after data manipulation. Nevertheless, it generally claims that the experiences are uniformly sampled in a replay memory (Mnih et al., 2013). Also, the fact that the given conditions of the transition are consistent for the observed data as well as the transformed data, makes the above definition reasonable. Next, we will show why the conditional divergence defined in Eq.(3) is an optimal invariant metric from theoretical perspectives.

We employ the Bayes' rule on the conditionally distribution $p(\mathbf{x}|\mathbf{e}) = p(\mathbf{e}|\mathbf{x})p(\mathbf{x})/p(\mathbf{e}), \forall \mathbf{x} \in \mathcal{O}, \mathbf{e} \in \mathcal{D}$. Then the transition operator $p(\mathbf{e}|\mathbf{x})$ can be further divided as $p(\mathbf{e}|\mathbf{s})p(\mathbf{s}|\mathbf{x})$ for any $\mathbf{x} \in \mathcal{O}$, if $\mathbf{e}$ and $\mathbf{x}$ are conditional independent given $\mathbf{s}$[1]. Eq.(3) is rewritten as,

$$\mathbb{E}_{\mathbf{e}}\left[d_{KL}\left(p(\mathbf{x}|\mathbf{e})||p(\mathbf{x}'|\mathbf{e})\right)\right] = \mathbb{E}_{\mathbf{e}|\mathbf{s}}\left[d_{KL}\left(p(\mathbf{s}|\mathbf{x})p(\mathbf{x})||p(\mathbf{s}|\mathbf{x}')p(\mathbf{x}')\right)\right] \quad (4)$$

Therefore, minimizing the conditional divergence leads to encoding the observation $\mathbf{x}$ and the transformed context $\mathbf{x}'$ into an invariant latent state space $\mathcal{S}$. As a consequence, the learnable pixel transformation is an optimality invariant combining a qualified encoder.

Now we have the observation encoder $g : \mathcal{O} \to \mathcal{S}$ mapping from the observed state $\mathcal{O}$ to the latent state $\mathcal{S}$ by a non-trivial function g such that $g(\mathbf{x}) = p(\mathbf{s}|\mathbf{x}), \forall \mathbf{x}$. Traditionally, there is only one encoder dubbed feature backbone in RL models. Since the pixel transformation could drift away, e.g., supported by different components with those supporting $\mathbf{x}$ (Du et al., 2019). To enforce the invariant hidden states, another state encoder g' that can map $\mathbf{x}'$ to $\mathbf{s}$ should exist, which is $g'(\mathbf{x}') = p(\mathbf{s}|\nu(\mathbf{x}, \cdot)), \forall \mathbf{x}'$, $\nu(\mathbf{x}, \cdot)$ is the transformation.

So far, the goal of learning the optimal transformed data and encoders boils down to minimizing the distance between representations $g(\mathbf{x})$ and $g'(\mathbf{x}')$. To tackle the issue, we first provide two definitions to introduce measurements as follows,

Definition 4.2. (ϵ-Approximation). Given a distance metric $d : \mathcal{O} \times \mathcal{S} \to \mathbb{R}_+$ satisfies $d(\mathbf{s}, \mathbf{s}) = 0, \forall \mathbf{s}$, and let $g, g' : \mathcal{O} \to \mathcal{S}$ be two functions. Let $\epsilon \geq 0$, given a distribution $\hat{\mathcal{D}}$ on $\mathcal{O}$, then g and g' are *ϵ-approximate* w.r.t. $(d, \mathcal{D})$ if $\mathbb{E}_{\mathbf{x}\sim\hat{\mathcal{D}}}\left[d(g(\mathbf{x}), g'(\mathbf{x}))\right] \leq \epsilon$.

Definition 4.3. (β-Similarity). Given a distance metric $d : \mathcal{O} \times \mathcal{S} \to \mathbb{R}_+$ satisfies $d(\mathbf{s}, \mathbf{s}) = 0, \forall \mathbf{s}$. There exists $g : \mathcal{O} \to \mathcal{S}$. Let $\beta \geq 0$, given distributions $\hat{\mathcal{D}}$ and $\hat{\mathcal{D}}'$ on $\mathcal{O}$, then $\mathbf{x}$ and $\mathbf{x}'$ are *β-similar* if $\mathbb{E}_{\mathbf{x}\sim\hat{\mathcal{D}}, \mathbf{x}'\sim\hat{\mathcal{D}}'}\left[d(g(\mathbf{x}), g(\mathbf{x}'))\right] \leq \beta$.

Without loss of generality, the distance between the encoded states $g(\mathbf{x})$ and $g'(\mathbf{x}')$ can be expressed as the following triangular inequality. To obtain a metric, Kullback-Leibler divergence is rewritten in a form of the square root of Jensen-Shannon divergence. Therefore, we have,

$$d(g(\mathbf{x}), g'(\mathbf{x}')) \leq \underbrace{d(g(\mathbf{x}), g'(\mathbf{x}))}_{\text{encoding: } \epsilon\text{-Approximation}} + \underbrace{d(g'(\mathbf{x}), g'(\mathbf{x}'))}_{\text{augmentation: } \beta\text{-Similarity}} \quad (5)$$

[1]The tuples in reply buffer can be written as $\mathbf{e} = (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ after encoding, which makes $\mathbf{s}$ an intermediate random variable.

From the view of invariant learning, minimizing the right side of the inequality can upper bound our problem. The first term on the right side is the so-called ϵ-approximation to measure the functional similarity after state abstraction, whereas the second term exists based on the procedure of data augmentation. Thus, we learn the encoders and shifted data simultaneously through the upper bound.

4.2 Optimal State Abstraction

To restrict the functional similarity of Eq.(5) from the perspective of learning a good encoding function with consistent semantics, the approaches formulating the main and momentum feature learning are utilized, motivated by contrastive learning (He et al., 2020). In particular, we enforce the encoding functions with exactly the same architecture, and use $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t$ at timestep t to update the parameters of momentum function $g_{\bar{\xi}}$ with g_ξ, where $\tau_m \in [0, 1]$ is the updating rate. Furthermore, we design a projection that is $f : \mathcal{S} \to \mathcal{Y}$ using a ReLU network (Petersen & Voigtlaender, 2018) to upper bound the divergence by minimizing the distance in the projected space $\mathcal{Y}$. Previously, the projection has been proposed by Chen et al. (2020), while the theoretical guarantees of the underlying mechanism with momentum updating for model-free RL are explained in this work.

Suppose the Markov chain $\mathcal{O} \xrightarrow{g} \mathcal{S} \xrightarrow{f} \mathcal{Y}$ holds. For two functions g and f in the compatible ranges, we use $f \circ g$ to denote the function composition $f(g(\cdot))$. Before showing the proposed data transformed method, we introduce technical lemmas to take advantage of the designable projection function by leveraging the convexity. The momentum updating paradigm is capable of turning into momentum feature updating through a convex function or an equivalence of the convex function.

Lemma 4.1. Assume that $h : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{Y}|}$ can be written as $h(\xi) = f(< \xi, \mathbf{s} >)$, for some $\mathbf{s} \in \mathbb{R}^{|\mathcal{S}|}$, and $f : \mathbb{R}^{|\mathcal{Y}|} \to \mathbb{R}^{|\mathcal{Y}|}$ with parameter ξ. Then, convexity of f implies the convexity of h.

Lemma 4.2. Given the dynamical updating: $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t$. By Lemma 4.1, $f_\xi = f_{\bar{\xi}}$ holds after convergence. As a result, the problem of $\min \mathbb{E}_{\mathbf{x}}[\|f_\xi \circ g_\xi(\mathbf{x}) - f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x})\|]$ is equivalent to the problem of $\min \mathbb{E}_{\mathbf{x}}[\|g_\xi(\mathbf{x}) - g_{\bar{\xi}}(\mathbf{x})\|]$.

To meet the requirement of a small upper bound, we state a theorem that provides some insights into why it is necessary to learn optimal transformed data together with the encoders.

Theorem 4.1. (CoIT) Suppose that Lipschitzness holds for functions g_ξ, $g_{\bar{\xi}}$, f_ξ and $f_{\bar{\xi}}$, respectively. The updating dynamics is: $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t, \tau_m \in [0, 1]$. For any input $\mathbf{x} \sim \hat{\mathcal{D}}$ and transformed $\mathbf{x}'$ obtained via the transform operator $\nu(\mathbf{x}, \cdot)$, optimizing the conditional divergence in Definition 4.1 means to minimize the upper bound as follows,

$$\mathbb{E}_{\mathbf{x}}\left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbf{x}'))\right] \leq \rho \mathbb{E}_{\mathbf{x}}\left[\|\mathbf{x} - \mathbf{x}'\|\right] \tag{6}$$

where $\rho = L_g\left(CL_f + \|\xi_f\|\right), C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$ are constants. L_f and L_g are Lipschitz constants of the functions $f(\mathbf{s})$ and $g(\mathbf{x})$, respectively.

The upper bound of the right side measures the margin of augmentation between original and transformed data. The left side measures the distribution changes in the latent space. Since both transformed data $\mathbf{x}'$ and encoder g are updating, incorporating augmentation directly cannot well meet the basic stationary environment. Hereby, the theorem suggests us that the automatic transformation is used to bound the representation learning so that the abstracted states enhance the stationary distribution of tuples e and facilitate efficient training. Proofs are given in Appendix A. The empirical comparisons of the projection network are also presented in Appendix D.

4.3 Parameterizable Observation

From the Theorem 4.1, we know the relation between the parameterized augmentation and the learnable latent state. That is, image transformation needs to be constrained by the distance between an observation and its associated augmentation. The optimal embedding can be obtained by minimizing of Eq.(5). Particularly, we parameterize the transformed data $\mathbf{x}'$ as $\nu(\mathbf{x}, \mathcal{G})$, where $\mathcal{G}$ is a Gaussian distribution dynamically changed along with the RL training.

In Algorithm 1, it defines the MDP $\mathcal{M}$ with Gaussian random variables $\mathcal{G}_0 \sim \mathcal{G}^{|\mathcal{O}|}$ for initialization. The `TRANSFORM` operator is fulfilled by the aforementioned pixel transformation ν which is a *shift*

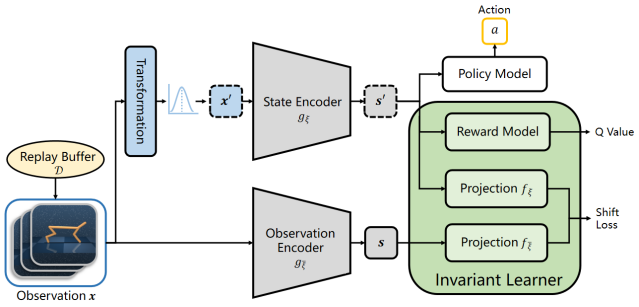

Figure 2: Overall architecture of CoIT. The observations are transformed following a Gaussian distribution $\mathcal{G}(\mu, \sigma)$ and encoded by the state encoder g_ξ. The observation encoder $g_{\bar{\xi}}$ and projection $f_{\bar{\xi}}$ are the exponentially moving average version of the state encoder g_ξ and projection f_ξ.

subject following Gaussian distribution $\mathcal{G}_t(\mu_t, \sigma_t)$ on the top of data interpolation. The transformation $\mathbf{x}' = \nu(\mathbf{x}, \mathcal{G})$ is normalized according to both $\mathbf{x}$ and learned distribution $\mathcal{G}$, and thereby contributes to the cumulative reward maximization in an interactive way. We use the scope to the observation sampling for an optimal state abstraction. It can be regarded as data sampling from the replay buffer to reach a training-friendly distribution, dubbed *Contrastive Invariant Transformation*.

4.4 Stabilizing Reward Function

It identifies one of the key impacts that CoIT in the RL training procedure as parametrizing the underlying invariant optimization to smooth the distribution $\mathcal{D}$ in the replay buffer. To further stabilize the reward function, we propose a mixed CoIT that samples multiple transformed data from the learned distribution $\mathcal{G}(\mu, \sigma)$, and then mix up the learned observation $\mathbf{x}'$. Similarly, we provide the invariant learning guarantee by optimizing the right side transformation in Theorem 4.2.

Algorithm 1 Parameterized Transformation in RL

1: **Initialization:** Draw distribution $\mathcal{G}_0 \sim \mathcal{G}^{|\mathcal{O}|}$ with given mean μ_0 and variance σ_0^2;
Set cumulative reward $\mathcal{R} = 0$.
2: **Training:**
3: **for** each timestep t in $0, \cdots, T$ **do**
4: $\quad \mathbf{x}'_t = \text{TRANSFORM}(\mathbf{x}_t, \mathcal{G}_t)$
5: $\quad \mathcal{R} = \mathcal{R} + \gamma^t r(\mathbf{x}'_t, \mathbf{a}_t)$
6: $\quad$ Adjust to an optimal $\mathcal{G}_t(\mu_t, \sigma_t)$.
7: **end for**

* Details about learning μ_t and σ_t are given in Algorithm 2 in Appendix B.

Theorem 4.2. (Mixed CoIT) Suppose that Lipschitzness holds for functions g_ξ, $g_{\bar{\xi}}$, f_ξ and $f_{\bar{\xi}}$, respectively. The updating dynamics is: $\bar{\xi}^t = (1 - \tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t$. For any input $\mathbf{x} \sim \hat{\mathcal{D}}$ and transformed $\mathbf{x}'$, the divergence with mixed transformed observation can be bound by,

$$\mathbb{E}_{\mathbf{x}} \left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbb{E}_{\mathbf{x}'}[\mathbf{x}'])) \right] \leq \rho \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{x}'} \left[||\mathbf{x} - \mathbf{x}'|| \right] \tag{7}$$

where $\rho = L_g \left(C L_f + \|\xi_f\| \right), C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$. L_f and L_g are Lipschitz constants of the functions $f(\mathbf{s})$ and $g(\mathbf{x})$ respectively. The proof of Theorem 4.2 is straightforward based on Theorem 4.1, and the details are presented in the supplement.

4.5 Learning Contrastive Invariant Transformation

With theoretical analysis of invariant transformations, we presented a new framework with normalization variants to ensure above discussed learning guarantees by optimizing parameters. We initialize a distribution $\mathcal{G}_t(\mu, \sigma)$ and use the TRANSFORM operator to produce different views of $\mathbf{x}_t$ (Algorithm 1). The transformed data $\mathbf{x}'_t$ is viewed as the positive pair of $\mathbf{x}_t$. We also utilize a similarity metric d to learn contrastive invariant transformation for the encoder $g_\xi(\cdot)$.

We first apply the bilinear interpolation to $\mathbf{x}_t$ and sample *shift* terms from $\mathcal{G}_t$ to produce multiple positive samples $\mathbf{x}_t^1, \mathbf{x}_t^2, .., \mathbf{x}_t^n$ and mix them together as $\mathbf{x}'_t$ following Theorem 4.2. To prevent the

distribution of transformed data from shifting too far away from the replay buffer $\mathcal{D}$, we borrow a similar idea from Yin et al. (2020) to regularize and smooth the distribution shift between the transformed and overall data. We use the statistical data stored in the BatchNorm layers to approximate the distribution of the overall data. In this way, the distribution shift between the transformed and overall data can be estimated by the following formulation,

$$\mathcal{K}_\omega(\mathbf{x}'_t) = \sum_l \left\|\tilde{\mu}(\mathbf{x}'_t) - \mathbb{E}(\tilde{\mu}_l(\mathbf{x})|\mathcal{O})\right\|_2 + \sum_l \left\|\tilde{\sigma}^2(\mathbf{x}'_t) - \mathbb{E}(\tilde{\sigma}^2_l(\mathbf{x})|\mathcal{O})\right\|_2 \tag{8}$$

where $\tilde{\mu}(\mathbf{x}'_t)$ and $\tilde{\sigma}^2(\mathbf{x}'_t)$ are the mean and variance of the transformed data and ω represents parameter collection $\{\mu_t, \sigma_t\}$ of the Gaussian distribution $\mathcal{G}_t(\mu, \sigma)$. The expectation terms $\mathbb{E}(\tilde{\mu}_l(\mathbf{x})|\mathcal{O})$ and $\mathbb{E}(\tilde{\sigma}^2_l(\mathbf{x})|\mathcal{O})$ respectively denote the estimation of the batch-wise mean and variance for the feature map corresponding to the l-th convolution layer, and $\mathcal{O}$ is the given observations.

Second, we utilize the similarity metric d proposed by Chen et al. (2020) for learning the encoder $g_\xi(\cdot)$ which maps high-dimensional observation to embeddings to meet the invariant transformation in Eq.5. Given a positive observation pair $(\mathbf{x}_t, \mathbf{x}'_t)$, the loss is given by,

$$\mathcal{L}_{\xi,\bar{\xi},\omega}(\mathcal{D}) \triangleq \left\|f_\xi(g_\xi(\mathbf{x}'_t)) - f_{\bar{\xi}}(g_{\bar{\xi}}(\mathbf{x}_t))\right\|_2^2 = 2 - 2 \cdot \frac{\left\langle f_\xi(g_\xi(\mathbf{x}'_t)), f_{\bar{\xi}}(g_{\bar{\xi}}(\mathbf{x}_t))\right\rangle}{\|f_\xi(g_\xi(\mathbf{x}'_t))\|_2 \cdot \left\|f_{\bar{\xi}}(g_{\bar{\xi}}(\mathbf{x}_t))\right\|_2} \tag{9}$$

Here $\bar{\xi}$ denotes the momentum version of parameters ξ and $f_\xi(\cdot)$ is a non-linear projection of the representations embedded by $g_\xi(\cdot)$. $\mathcal{D}$ indicates the tuples stored in the replay buffer.

Next, we update the critic network Q_θ with transformed data $\mathbf{x}'_t$ and $\mathbf{x}'_{t+n}$ to minimize the TD error for n-steps returns. This is regarded as a regularized Q learning by Yarats et al. (2020) where the regularized representation learning is beneficial for optimal actions (Zhang et al., 2020a). Specifically,

$$J_Q(\mathcal{D}; \theta, \omega, \xi) = \left(Q_\theta\left(g_\xi(\mathbf{x}'_t), \mathbf{a}_t\right) - \sum_{i=0}^{n-1} \gamma^i r_{t+i} - \gamma^n Q_{\bar{\theta}}\left(g_\xi(\mathbf{x}'_{t+n}), \pi(\cdot|g_\xi(\mathbf{x}'_{t+n}))\right)\right)^2 \tag{10}$$

Eventually, we give the unified objective function as the full version of the CoIT,

$$J_{\theta,\xi,\omega}(\mathcal{D}) = J_Q(\mathcal{D}) + \alpha\mathcal{L}_{\xi,\bar{\xi},\omega}(\mathcal{D}) + \lambda\mathcal{K}_\omega(\mathcal{D}) \tag{11}$$

where α and λ are hyper-parameters and the overall architecture is presented in Figure 2. We replace the vanilla Q-learning by $J_{\theta,\xi,\omega}(\mathcal{D})$ and the entire algorithm is presented in Algorithm 2 in Appendix B. Then, we evaluate CoIT on popular benchmarks to demonstrate the benefits of our method.

5 Experiments

In this section, we benchmark our method on the DeepMind control suite and Atari100K. We compare CoIT with prior model-free methods first, then we present ablation studies to show the details of our method. Implementation details can be found in Appendix C.

5.1 Environments

DMControl. DeepMind control suite (Tassa et al., 2018) is a widely used benchmark with several robot control tasks. Each episode is set to be $1,000$ frames and we use the total experienced frames to measure the data-efficiency. The per-frame reward is in the unit interval $[0, 1]$, so each episode contains a total reward of no more than $1,000$. Considering the different difficulties depending on tasks, we refer to setting more episodes with hard tasks for better evaluation.

Atari100K. There have been a number of prior papers that have benchmarked data-efficiency on the Atari 2600 Games for discrete control. Van Hasselt et al. (2019) and Kielak (2019) propose the data-efficient version of Rainbow DQN (Hessel et al., 2018) compared with human performance (Kaiser et al., 2019) within 100K time steps (400K frames, frame skip of 4). This sample-constrained evaluation is the so-called Atari100K and we benchmark CoIT on all 26 Atari Games.

5.2 Baselines

DMControl. For continuous control we present several baselines, including methods of using data augmentation and contrastive learning to improve data-efficiency: (i) DrQ-v2 (Yarats et al., 2021), (ii)

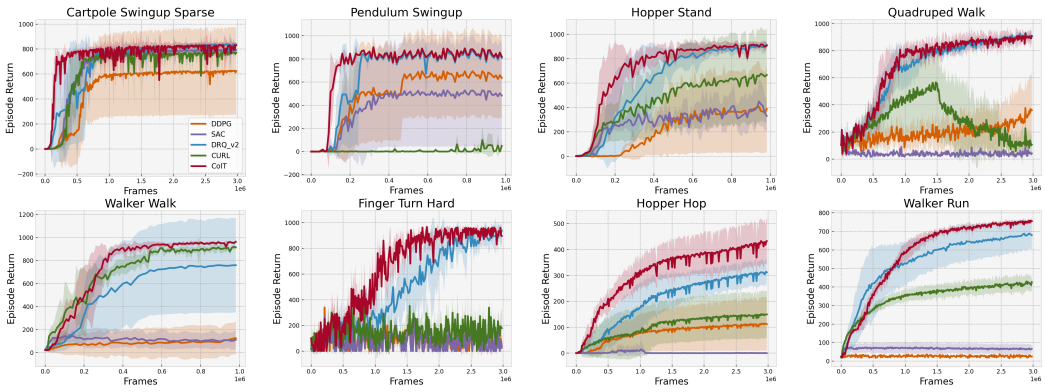

Figure 3: Results of complex tasks in DMControl. These tasks are chosen to offer multiple degrees of challenges, including complex dynamics, sparse rewards, hard exploration, and more.

CURL (Laskin et al., 2020b), (iii) Pixel SAC, and (iv) Pixel DDPG : Vanilla SAC and DDPG training directly from pixels. All methods are evaluated with the same periodicity of frames and average over 10 episodes return for evaluation query.

Atari100K. To benchmark the data-efficiency of CoIT for discrete control tasks, we compare our method to (i) DrQ (Yarats et al., 2020), (ii) CURL (Laskin et al., 2020b), (iii) SPR (Schwarzer et al., 2020), (iv) Random Agent, and Human Performance (Kaiser et al., 2019). All the algorithms are evaluated within 100K time steps for interaction. We average CoIT's performance over 10 random seeds and report the *best* score for each game following prior works.

5.3 Main Results

DMControl. We choose 8 complex tasks from the DMControl for evaluation and present the results in Figure 3 and Table 2 in Appendix C. We also report the percentage (%) of score solved in the DMControl for baselines and CoIT in 500K and 100K steps in Figure 1. Below are key findings: (i) CoIT outperforms vanilla DDPG and SAC in a wide range. (ii) We also compare CoIT with DrQ-v2, a remarkable method for continuous control, to better demonstrate our method's data-efficiency. (iii) From general trends of the learning curves, CoIT improves or keeps the data-efficiency in a more stable manner which is not trivial on DMControl tasks.

Atari100K. We present results for Atari100K in Table 1 and below are key findings: (i) CoIT achieves top-performance on 10 of 26 games while still being competitive in the rest. (ii) CoIT *surpasses superhuman performance on 6 games* on the basis of Rainbow DQN. (iii) We also report the *mean* and *std* of the scores achieved by CoIT in 10 of 26 games which are top-performance. We present the results of two versions of CoIT (mixed & no-mixed) in Figure 6 in the appendix. From the histogram, we find that CoIT has much better stability, which is similar to the observation in the DMControl.

5.4 Ablation Studies

We first visualize the TRANSFORM operator to demonstrate that there exists an invariant transformation for each task. We initialize the Gaussian distribution $\mathcal{G}_t(\mu, \sigma)$ based on the range of pixel shifts and plot the curves of the *mean* μ and *std* σ during training in the DMControl in Figure 4.

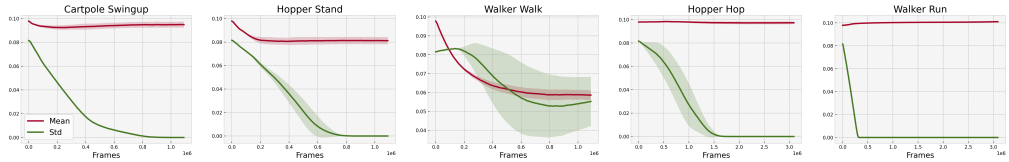

Figure 4: Visualization of the parameters of the Gaussian distribution for TRANSFORM.

Table 1: Mean episodic returns achieved by CoIT and baselines on 26 Atari games benchmarked at 100K environment steps. The results are recorded and averaged over 10 random seeds.

Game	Human	Random	SimPLe	CURL	DrQ	SPR	CoIT
Aline	7127.7	227.8	616.9	558.2	771.2	801.5	**1206.7**
Amidar	1719.5	5.8	88.0	142.1	102.8	176.3	**182.3**
Assault	742.0	222.4	527.2	600.6	452.4	571.0	**635.7**
Asterix	8503.3	210.0	1128.3	734.5	603.5	**977.8**	709.0
Bank Heist	753.1	14.2	34.2	131.6	168.9	**380.9**	124.8
Battle Zone	37187.5	2360.0	5184.4	14870.0	12954.0	**16651.0**	13760.0
Boxing	12.1	0.1	9.1	1.2	6.0	**35.8**	23.6
Breakout	30.5	1.7	16.4	4.9	16.1	**17.1**	16.1
Chopper Command	7387.8	811.0	1246.9	1058.5	780.3	974.8	**1338.0**
Crazy Climber	35829.4	10780.5	**62583.6**	12146.5	20516.5	42923.6	17538.0
Demon Attack	1971.0	152.1	208.1	817.6	**1113.4**	545.2	846.4
Freeway	29.6	0.0	20.3	26.7	9.8	24.4	**29.6**
Frostbite	4334.7	65.2	254.7	1181.3	331.1	1821.5	**2069.8**
Gopher	2412.5	257.6	**771.0**	669.3	636.3	715.2	746.8
Hero	30826.4	1027.0	2556.6	6279.3	3736.3	7019.2	**7572.8**
Jamesbond	302.8	29.0	125.3	**471.0**	236.0	349.0	336.0
Kangaroo	3035.0	52.0	323.1	872.5	940.6	3276.4	**4116.6**
Krull	2665.5	1598.0	**4539.9**	4229.6	4018.1	3688.9	3426.2
Kung Fu Master	22736.3	258.5	**17257.2**	14307.8	9111.0	13192.7	9250.0
Ms Pacman	6951.6	307.3	1480.0	1465.5	960.5	1313.2	**1509.6**
Pong	14.6	−20.7	**12.8**	−16.5	−8.5	−5.9	1.5
Private Eye	69571.3	24.9	58.3	**218.4**	−13.6	124.0	145.7
Qbert	13455.0	163.9	1288.8	1042.4	854.4	669.1	**2117.5**
Road Runner	7845.0	11.5	5640.6	5661.0	8895.1	**14220.5**	11758.5
Seaquest	42054.7	68.4	**683.3**	384.5	301.2	583.1	554.0
Up N Down	11693.2	533.4	3350.3	2955.2	3180.8	**28138.5**	4734.2

According to the curves below, we find that the mean and std converge to an interval as the training goes on. These results demonstrate that the gaussian distribution proposed in CoIT could automatically find a `TRANSFORM` to smooth the distribution shift between the different views of the same observation, therefore being beneficial to the representation learning.

Then, we study the effects of different components in Eq.(11). This object function is composed of two parts: $\mathcal{K}_\omega(\mathbf{x}'_t)$ for regularization and $\mathcal{L}_{\xi,\bar{\xi},\omega}(\mathcal{D})$ for similarity metric. On this basis, we divide CoIT into 4 versions: (i) *Critic*. Transformation is only updated with the critic. (ii) *X-stats & Critic*. Transformation is updated by critic and $\mathcal{K}_\omega(\mathbf{x}'_t)$ together. (iii) *H-dist & Critic*. Transformation is updated by critic and $\mathcal{L}_{\xi,\bar{\xi},\omega}(\mathcal{D})$ together. (iv) *Unified Objective*. We evaluate all of these versions on 8 representative tasks from the DMControl and present the results in Figure 9 in the appendix.

Compared *Critic* with other variants, we demonstrate that both of the components are beneficial to the performance. Though *Critic* is data-efficient on most tasks, it may fall into trivial solutions. To solve this issue, we utilize the regularization in Eq.(8) with the similarity metric in Eq.(9) to meet the invariant transformation. Thus the *Unified Objective*'s performance leads ahead of all tasks. See Appendix C for extra ablation studies.

6 Conclusion

A novel pixel transformation CoIT under model-free RL algorithms that significantly improves the data-efficiency and stability for visual tasks is introduced in this work. We theoretically analyze how the learnable transformation constrains the distribution of the abstracted data, and dissect its benefits to representation learning. CoIT is no need for any additional modifications to the backbone RL algorithm and is easy to implement. We compare CoIT to SOTA methods on popular benchmarks and certify that it gains promising performance with advanced stability. Hopefully, contrastive invariant transformation can lead to a new branch for representation learning in RL.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China No. 2020AAA0103400, National Key Research and Development Program of China No. 2021ZD0201504, National Natural Science Foundation of China under Grant 62273347, and CCF-Tencent Open Research Fund RAGR20220104. We thank anonymous reviewers for their discussions and feedback on the paper.

REFERENCES

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.

David Andre and Stuart J Russell. State abstraction for programmable reinforcement learning agents. In *Aaai/iaai*, pp. 119–125, 2002.

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

Randall Balestriero, Jerome Pesenti, and Yann LeCun. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*, 2021.

Andrea Banino, Adrià Puidomenech Badia, Jacob Walker, Tim Scholtes, Jovana Mitrovic, and Charles Blundell. Coberl: Contrastive bert for reinforcement learning. *arXiv preprint arXiv:2107.05431*, 2021.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5): 679–684, 1957.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Pablo Samuel Castro and Doina Precup. Using bisimulation for policy transfer in mdps. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

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

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

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*, pp. 1665–1674. PMLR, 2019.

Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Anima Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. *arXiv preprint arXiv:2106.09678*, 2021.

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.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

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

Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13611–13617. IEEE, 2021.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

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.

Kacper Piotr Kielak. Do recent advancements in model-based deep reinforcement learning really improve data efficiency? 2019.

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

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

Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019.

Kuang-Huei Lee, Ian Fischer, Anthony Liu, Yijie Guo, Honglak Lee, John Canny, and Sergio Guadarrama. Predictive information accelerates learning in rl. *Advances in Neural Information Processing Systems*, 33:11890–11901, 2020.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Bogdan Mazoure, Remi Tachet des Combes, Thang Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *arXiv preprint arXiv:2006.07217*, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Jing Peng and Ronald J Williams. Incremental multi-step q-learning. In *Machine Learning Proceedings 1994*, pp. 226–232. Elsevier, 1994.

Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*, 2020.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

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.

David Silver, Hado Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, et al. The predictron: End-to-end learning and planning. In *International Conference on Machine Learning*, pp. 3191–3199. PMLR, 2017.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Manan Tomar, Amy Zhang, Roberto Calandra, Matthew E Taylor, and Joelle Pineau. Model-invariant state abstractions for model-based reinforcement learning. *arXiv preprint arXiv:2102.09850*, 2021.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Hado P Van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32, 2019.

Tongzhou Wang, Simon S Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. *arXiv preprint arXiv:2206.15477*, 2022.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2020.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5276–5289, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

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*, pp. 11214–11224. PMLR, 2020a.

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

Hanping Zhang and Yuhong Guo. Generalization of reinforcement learning with policy-aware adversarial data augmentation. *arXiv preprint arXiv:2106.15587*, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13001–13008, 2020.

APPENDIX

A MISSING PROOFS

Lemma A.1. Assume that $h : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{Y}|}$ can be written as $h(\xi) = f(< \xi, \mathbf{s} >)$, for some $\mathbf{s} \in \mathbb{R}^{|\mathcal{S}|}$, and $f : \mathbb{R}^{|\mathcal{Y}|} \to \mathbb{R}^{|\mathcal{Y}|}$ with parameter ξ. Then, convexity of f implies the convexity of h.

Proof. Let $\xi_1, \xi_2 \in \mathbb{R}^{|\mathcal{S}|}$ and $\tau \in [0, 1]$. We have

$$\begin{aligned} &h(\tau\xi_1 + (1-\tau)\xi_2) = f(< \tau\xi_1 + (1-\tau)\xi_2, \mathbf{s} >) \\ &= f(< \tau\xi_1, \mathbf{s} > + < (1-\tau)\xi_2, \mathbf{s} >) = f(\tau < \xi_1, \mathbf{s} > + (1-\tau) < \xi_2, \mathbf{s} >) \\ &\leq \tau f(< \xi_1, \mathbf{s} >) + (1-\tau) f(< \xi_2, \mathbf{s} >) = \tau h(\xi_1) + (1-\tau) h(\xi_2) \end{aligned} \tag{12}$$

where the last inequality follows from the convexity of f. □

Lemma A.2. Given the updating dynamics: $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t$. By Lemma 4.1, $f_\xi = f_{\bar{\xi}}$ holds after convergence. As a result, the problem of $\min \mathbb{E}_{\mathbf{x}}[\|f_\xi \circ g_\xi(\mathbf{x}') - f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x})\|]$ is equivalent to the problem of $\min \mathbb{E}_{\mathbf{x}}[\|g_\xi(\mathbf{x}') - g_{\bar{\xi}}(\mathbf{x})\|]$.

Proof. Through Lemma A.1, we know the convexity of a designed function f can give rise to the convexity of the function h with the parameters as the input. Therefore, we design our projections f_ξ and $f_{\bar{\xi}}$ as $f(< \xi, \mathbf{s} >)$ and $f(< \bar{\xi}, \mathbf{s} >)$ respectively. For instance, ReLU and MLP can be adopted here. Using the dynamic: $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t$ together with $h(\xi)$ mentioned in Lemma A.1, we obtain the divergence of hidden representations $f_\xi \circ g_\xi(\mathbf{x}')$ and $f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x})$,

$$\begin{aligned} \mathbb{E}_{\mathbf{x}}[\|f_\xi \circ g_\xi(\mathbf{x}') - f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x})\|] &= \mathbb{E}_{\mathbf{x}}[\|f_\xi \circ (g_\xi(\mathbf{x}') - g_{\bar{\xi}}(\mathbf{x}))\| \\ &\leq \mathbb{E}_{\mathbf{x}}[\|\xi_f\| \|(g_\xi(\mathbf{x}') - g_{\bar{\xi}}(\mathbf{x}))\|] \\ &= \mathbb{E}_{\mathbf{x}}[\|(g_\xi(\mathbf{x}') - g_{\bar{\xi}}(\mathbf{x}))\|] \|\xi_f\| \end{aligned} \tag{13}$$

where $\|\xi_f\|$ is the parameter of the projection $\|f_\xi\|$. The first equality is determined by the approximation of convergence analysis, which is $f_\xi = f_{\bar{\xi}}$. We use Cauchy–Schwartz inequality here. Note that a premise in this lemma is that the momentum updating reached convergence, which means $f_\xi = f_{\bar{\xi}}$. Minimizing the right side bound equals optimizing the problem of the left side in Eq.(13). When the norm of $\|\xi_f\|$ is fixed, the proof completes. □

Assumption A.1. (L_f-Lipschitzness). Let the projection function $f(\mathbf{s}) : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{Y}|}$ is L_f-Lipschitz, we have that $\forall \mathbf{s}, \mathbf{s}'$, $|f(\mathbf{s}') - f(\mathbf{s})| \leq L_f \|\mathbf{s}' - \mathbf{s}\|$.

Assumption A.2. (L_g-Lipschitzness). Let the projection function $g(\mathbf{x}) : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{S}|}$ is L_g-Lipschitz, we have that $\forall \mathbf{x}, \mathbf{x}'$, $|g(\mathbf{x}') - g(\mathbf{x})| \leq L_g \|\mathbf{x}' - \mathbf{x}\|$.

Theorem A.1. (CoIT.) Suppose that Lipschitzness holds for functions g_ξ, $g_{\bar{\xi}}$, f_ξ and $f_{\bar{\xi}}$, respectively. The updating dynamics is: $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t, \tau_m \in [0, 1]$. For any input $\mathbf{x} \sim \hat{\mathcal{D}}$ and shifted $\mathbf{x}'$ obtained via the transform operator $\nu(\mathbf{x}, \cdot)$, optimizing the conditional divergence in Definition 4.1 means to minimize the upper bound as follows,

$$\mathbb{E}_{\mathbf{x}}\left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbf{x}'))\right] \leq \rho \mathbb{E}_{\mathbf{x}}\left[\|\mathbf{x} - \mathbf{x}'\|\right] \tag{14}$$

where $\rho = L_g\left(CL_f + \|\xi_f\|\right), C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$ are constants. L_f and L_g are Lipschitz constants of the functions $f(\mathbf{s})$ and $g(\mathbf{x})$ respectively.

Proof. By incorporating Lemma A.2 for the problem of $\min \mathbb{E}_{\mathbf{x}}[\|f_\xi \circ g_\xi(\mathbf{x}') - f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x})\|]$, it leads to a divergence with projections where a triangular inequality holds,

$$d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbf{x}')) \leq \underbrace{d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbf{x}))}_{\text{state } \epsilon\text{-approximation}} + \underbrace{d(f_\xi \circ g_\xi(\mathbf{x}), f_\xi \circ g_\xi(\mathbf{x}'))}_{L\text{-Lipschitzness}} \tag{15}$$

Set $\mathbf{s}' = g_\xi(\mathbf{x})$ and $\mathbf{s} = g_{\bar{\xi}}(\mathbf{x})$. Now we use Lemma A.1 and the updating dynamics for the designed projection f_ξ and $f_{\bar{\xi}}$, $\tau_m \in [0, 1]$, we can obtain,

$$h(\bar{\xi}^t) = h((1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t) \leq (1-\tau_m) h(\bar{\xi}^{t-1}) + \tau_m h(\xi^t) \tag{16}$$

By designing a projection that satisfies $h(\bar{\xi}^t) = f^t_{\bar{\xi}}(\mathbf{s})$ and $h(\xi^t) = f^t_{\xi}(\mathbf{s})$, we have

$$f^t_{\bar{\xi}}(\mathbf{s}) \leq (1-\tau_m) f^{t-1}_{\bar{\xi}}(\mathbf{s}) + \tau_m f^t_{\xi}(\mathbf{s}) \tag{17}$$

The goal is to minimize ϵ-approximation on latent distance $d(f_{\bar{\xi}}(\mathbf{s}), f_{\xi}(\mathbf{s}'))$ such that the left side of Eq.(15) is minimized. Particularly, given l_p-norm as distance $d(\cdot,\cdot)$ at the timestep t, and a ReLU network as function f, it leads to,

$$\begin{aligned}
\|f^t_{\bar{\xi}}(\mathbf{s}) - f^t_{\xi}(\mathbf{s}')\| &\leq \|\tau f^{t-1}_{\bar{\xi}}(\mathbf{s}) + (1-\tau) f^t_{\xi}(\mathbf{s}) - f^t_{\xi}(\mathbf{s}')\| \\
&= \|\tau f^{t-1}_{\bar{\xi}}(\mathbf{s}) - \tau f^t_{\xi}(\mathbf{s}) + f^t_{\xi}(\mathbf{s}) - f^t_{\xi}(\mathbf{s}')\| \\
&\leq \tau \|f^{t-1}_{\bar{\xi}}(\mathbf{s}) - f^t_{\xi}(\mathbf{s})\| + \|f^t_{\xi}(\mathbf{s}) - f^t_{\xi}(\mathbf{s}')\| \\
&\leq \tau \|f^{t-1}_{\bar{\xi}}(\mathbf{s}) - f^{t-1}_{\xi}(\mathbf{s}')\| + \tau \|f^{t-1}_{\xi}(\mathbf{s}') - f^t_{\xi}(\mathbf{s})\| + \|f^t_{\xi}(\mathbf{s}) - f^t_{\xi}(\mathbf{s}')\|
\end{aligned} \tag{18}$$

where $\tau = 1 - \tau_m$. L_f-Lipschitzness and L_g-Lipschitzness assumptions are employed as stated in Assumption A.1 and A.2. Suppose the updating has achieved convergence, Eq.(18) turns to the following inequality,

$$\|f_{\bar{\xi}}(\mathbf{s}) - f_{\xi}(\mathbf{s}')\| \leq \frac{1+\tau}{1-\tau} \|f_{\xi}(\mathbf{s}) - f_{\xi}(\mathbf{s}')\| \leq \frac{(1+\tau) L_f}{1-\tau} \|g_{\xi}(\mathbf{x}) - g_{\xi}(\mathbf{x}')\| \leq \frac{(1+\tau) L_f L_g}{1-\tau} \|\mathbf{x} - \mathbf{x}'\| \tag{19}$$

On the other hand, the second term on the right side of Eq.(15) can be rewritten as,

$$\|f_{\xi} \circ g_{\xi}(\mathbf{x}) - f_{\xi} \circ g_{\xi}(\mathbf{x}')\| \leq \|\xi_f\| \|g_{\xi}(\mathbf{x}) - g_{\xi}(\mathbf{x}')\| \leq L_g \|\xi_f\| \|\mathbf{x} - \mathbf{x}'\| \tag{20}$$

Altogether, we substitute Eq.(19) and Eq.(20) in Eq.(15), we finally obtain,

$$\mathbb{E}_{\mathbf{x}} \left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_{\xi} \circ g_{\xi}(\mathbf{x}')) \right] \leq \left(L_g \|\xi_f\| + \frac{(1+\tau) L_f L_g}{1-\tau} \right) \mathbb{E}_{\mathbf{x}} \left[\|\mathbf{x} - \mathbf{x}'\| \right] \tag{21}$$

The proof is finished. □

Theorem A.2. (Mixed CoIT.) Suppose that Lipschitzness holds for functions g_{ξ}, $g_{\bar{\xi}}$, f_{ξ} and $f_{\bar{\xi}}$, respectively. The updating dynamics is: $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t$. For any input $\mathbf{x} \sim \hat{\mathcal{D}}$ and shifted $\mathbf{x}' \sim \mathcal{G}$, the divergence with mixed augmented states can be bound by,

$$\mathbb{E}_{\mathbf{x}} \left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_{\xi} \circ g_{\xi}(\mathbb{E}_{\mathbf{x}'}[\mathbf{x}'])) \right] \leq \rho \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{x}'} \left[||\mathbf{x} - \mathbf{x}'|| \right] \tag{22}$$

where $\rho = L_g \left(C L_f + \|\xi_f\| \right), C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$. L_f and L_g are Lipschitz constants of the functions $f(\mathbf{s})$ and $g(\mathbf{x})$ respectively

Proof. Based on Theorem A.1, we can view the mixup $\mathbb{E}_{\mathbf{x}'}[\mathbf{x}']$ as a sort of transformed data, and thereby we have,

$$\mathbb{E}_{\mathbf{x}} \left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_{\xi} \circ g_{\xi}(\mathbb{E}_{\mathbf{x}'}[\mathbf{x}'])) \right] \leq L_g \left(C L_f + \|\xi_f\| \right) \mathbb{E}_{\mathbf{x}} \left[\|\mathbf{x} - \mathbb{E}_{\mathbf{x}'}[\mathbf{x}']\| \right] \tag{23}$$

Since $\mathbb{E}_{\mathbf{x}} \left[\|\mathbf{x} - \mathbb{E}_{\mathbf{x}'}[\mathbf{x}']\| \right] = \mathbb{E}_{\mathbf{x}} \left[\|\mathbb{E}_{\mathbf{x}'}[\mathbf{x} - \mathbf{x}']\| \right] \leq \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{x}'} \left[\|\mathbf{x} - \mathbf{x}'\| \right]$, we can finally obtain,

$$\mathbb{E}_{\mathbf{x}} \left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_{\xi} \circ g_{\xi}(\mathbb{E}_{\mathbf{x}'}[\mathbf{x}'])) \right] \leq L_g \left(C L_f + \|\xi_f\| \right) \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{x}'} \left[\|\mathbf{x} - \mathbf{x}'\| \right] \tag{24}$$

which completes the proof. □

B PSEUDOCODE

Algorithm 2 CoIT. Similarity metric. Learnable data transformation.

Inputs:
STATE encoder g_ξ, policy π_ϕ, Q-functions $Q_{\theta_1}, Q_{\theta_2}$, OBSERVATION encoder $g_{\bar{\xi}}$, PROJECTION f_ξ, $f_{\bar{\xi}}$
$\mu, \sigma \sim \mathcal{G}$ for TRANSFORM.
Scheduled standard deviation $\tilde{\sigma}(t)$ for the exploration noise
Training steps T, mini-batch size N, learning rate δ, target update rate τ, clip value c, TRANSFORM learning rate δ_{aug}, MOMENTUM update rate τ_m
Training:
for each timestep t in $1..T$ **do**
  $\tilde{\sigma}_t \leftarrow \tilde{\sigma}(t)$
  $\mathbf{x}'_t \leftarrow$ TRANSFORM$(\mathrm{x}_t, \mathcal{G}_t)$ and $\sigma_t \leftarrow 0$
  $\mathbf{a}_t \leftarrow \pi_\phi(g_\xi(\mathbf{x}'_t)) + \tilde{\epsilon}$ and $\tilde{\epsilon} \sim \mathcal{G}(0, \tilde{\sigma}_t)$
  $\mathbf{x}_{t+1} \sim P(\cdot|\mathbf{x}_t, \mathbf{a}_t)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}_t, \mathbf{a}_t, \mathcal{R}(\mathbf{x}_t, \mathbf{a}_t), \mathbf{x}_{t+1})$
  UPDATECRITIC$(\mathcal{D}, \tilde{\sigma}_t)$
  UPDATEACTOR$(\mathcal{D}, \tilde{\sigma}_t)$
end for
procedure UpdateCritic$(\mathcal{D}, \tilde{\sigma})$
  $\{(\mathbf{x}_t, \mathbf{a}_t, r_{t:t+n-1}, \mathbf{x}_{t+n})\}_{i=1}^N \sim \mathcal{D}$
  $\mathbf{x}'_t, \mathbf{x}'_{t+n} \leftarrow$ TRANSFORM$(\mathrm{x}_t, \mathcal{G}_t)$, TRANSFORM$(\mathrm{x}_{t+n}, \mathcal{G}_t)$
  $\mathbf{s}_t, \mathbf{s}_{t+n} \leftarrow g_\xi(\mathbf{x}'_t), g_\xi(\mathbf{x}'_{t+n})$
  $\mathbf{s}_{\bar{\xi}} \leftarrow g_{\bar{\xi}}(\mathbf{x}_t)$
  Measure similarity by $\mathcal{L}_{\xi,\bar{\xi},\omega}$
  $\mathbf{a}_{t+n} \leftarrow \pi_\phi(\mathbf{s}_{t+n}) + \tilde{\epsilon}$ and $\tilde{\epsilon} \sim \mathcal{G}(0, \tilde{\sigma}_t)$
  Compute $J_{\theta,\omega}(\mathcal{D})$ for Q_θ and TRANSFORM updating
  $\mu \leftarrow \mu - \delta_{\text{aug}} \nabla_\mu (J_{\theta,\omega,\xi}(\mathcal{D}))$ `# Learning` μ_t, `Line 6 in Algorithm 1.`
  $\sigma \leftarrow \sigma - \delta_{\text{aug}} \nabla_\sigma (J_{\theta,\omega,\xi}(\mathcal{D}))$ `# Learning` σ_t, `Line 6 in Algorithm 1.`
  $\xi \leftarrow \xi - \delta \nabla_\xi J_{\theta,\omega,\xi}(\mathcal{D})$
  $\theta \leftarrow \theta - \delta \nabla_\theta J_{\theta,\omega,\xi}(\mathcal{D})$
  $\hat{\theta} \leftarrow (1-\tau)\hat{\theta} + \tau\theta$
  $\bar{\xi} \leftarrow (1-\tau_m)\bar{\xi} + \tau_m \xi$
end procedure
procedure UpdateActor$(\mathcal{D}, \tilde{\sigma})$
  $\{\mathbf{x}_t\}_{i=1}^N \sim \mathcal{D}$
  $\mathbf{s}_t \leftarrow g_\xi(\text{TRANSFORM}(\mathbf{x}_t, \mathcal{G}_t))$
  $\mathbf{a}_t \leftarrow \pi_\phi(\mathbf{s}_t) + \tilde{\epsilon}$ and $\tilde{\epsilon} \sim \text{clip}(\mathcal{G}(0, \tilde{\sigma}))$
  Update the actor using the sampled policy gradient
  $\nabla_\phi J \approx \frac{1}{N} \sum_i \nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})|_{\mathbf{s}=\mathbf{s}_t, \mathbf{a}=\mathbf{a}_t} \nabla_\phi \pi(\mathbf{s})|_{\mathbf{s}_t}$
  $\phi \leftarrow \phi - \delta \nabla_\phi J$
end procedure

Table 2: We evaluate CoIT on the DMControl at 500k and 100k steps, compared with 4 baselines. CoIT outperfoms on 8 of 7 tasks both in 500k and 100k steps.

500K Steps Scores	CoIT	DrQ-v2	CURL	DDPG	SAC
Cartpole swingup sparse	731 ± 95	733 ± 47	$\mathbf{753 \pm 38}$	584 ± 328	714 ± 25
Pendulum swingup	$\mathbf{828 \pm 2}$	814 ± 12	53 ± 56	641 ± 352	488 ± 445
Hopper stand	$\mathbf{907 \pm 29}$	886 ± 44	663 ± 377	370 ± 340	390 ± 152
Walker walk	$\mathbf{958 \pm 10}$	758 ± 410	916 ± 17	118 ± 146	107 ± 74
Quadruped walk	$\mathbf{754 \pm 56}$	657 ± 133	272 ± 215	126 ± 70	32 ± 12
Finger turn hard	$\mathbf{509 \pm 126}$	271 ± 240	121 ± 88	73 ± 46	60 ± 42
Hopper hop	$\mathbf{386 \pm 81}$	176 ± 97	135 ± 88	79 ± 72	11 ± 24
Walker run	$\mathbf{585 \pm 47}$	524 ± 118	394 ± 35	28 ± 3	71 ± 28
100K Steps Scores					
Cartpole swingup sparse	$\mathbf{697 \pm 108}$	277 ± 380	480 ± 293	16 ± 36	21 ± 18
Pendulum swingup	$\mathbf{786 \pm 24}$	426 ± 392	3 ± 1	332 ± 358	221 ± 271
Hopper stand	$\mathbf{627 \pm 291}$	305 ± 339	460 ± 280	3 ± 3	252 ± 41
Walker walk	$\mathbf{491 \pm 258}$	321 ± 228	72 ± 93	465 ± 250	74 ± 74
Quadruped walk	$\mathbf{176 \pm 68}$	160 ± 66	147 ± 122	131 ± 28	54 ± 22
Finger turn hard	$\mathbf{223 \pm 87}$	58 ± 50	179 ± 135	73 ± 46	71 ± 36
Hopper hop	$\mathbf{179 \pm 35}$	34 ± 41	6 ± 10	2 ± 3	0 ± 0
Walker run	167 ± 11	201 ± 125	$\mathbf{257 \pm 15}$	26 ± 3	66 ± 24

C EXPERIMENTS

In this section, we explain the implementation details for CoIT in the DMControl setting. We use the DDPG as the backbone RL algorithm objective and build on top of the implementation from Yarats et al. (2021). We present in detail the hyperparameters for the architecture and optimization.

C.1 ACTOR AND CRITIC NETWORKS

The clipped double Q-learning (Van Hasselt et al., 2016; Fujimoto et al., 2018) is applied for the critic, where each Q-function is parametrized as a 3-layer MLP with `ReLU` activations after each layer except for the last. The actor is also a 3-layer MLP with `ReLU`s that outputs mean for the action. The hidden dimension is set to 1024 for both the critic and the actor.

C.2 ENCODER NETWORK

The architecture of the encoder is based on the work of Yarats et al. (2019), which has four convolutional layers with 3×3 kernels and 32 channels. The `ReLU` activation is applied after each conv layer. We also utilize `BatchNorm` (Ioffe & Szegedy, 2015) after each activation rather than `LayerNorm` (Ba et al., 2016) after a single fully-connected layer. The stride for the first conv layer is 2 while 1 for the rest. `BatchNorm` is also applied to normalize the fully-connected layer where the output of the convent is fed into. Finally, the use of `tanh` nonlinearity and the initialization of weight are consistent with the prior work. The actor and critic share the same encoder, although the encoder only uses the gradients from the critic for updating.

C.3 PROJECTION NETWORK

To encode the feature map into a latent space for similarity metrics, we introduce a projection network with its momentum version. The projection network is a 2-layer MLP with `ReLU` activations after the first layer. We set the hidden dimension as 1024 for each layer and the output dimension for contrastive loss is 128.

C.4 TRANSFORMATION DETAILS

Following prior works, we stack n 84×84 RGB images as observations that involve the temporal information to present the underlying state, where n is 4 in Atari100K and n is 3 in DMControl. For transformation, we first expand the images to 92×92 by padding 4 pixels at each side and then use `grid_sample` with the TRANSFORM operator to interpolate them to the original size.

C.5 HYPERPARAMETERS

For fair comparisons, our hyper-parameters are as consistent with Yarats et al. (2021) as possible. CoIT introduces two new hyperparameters α and λ to scale $\mathcal{K}_\omega(\mathbf{x}'_t)$ and $\mathcal{L}_{\xi,\bar{\xi},\omega}(\mathcal{D})$ in magnitude. The overview of used hyper-parameters in Atari100K and DeepMind Control Suite is shown in Table 4 and Table 5. We also present the learning rate of `TRANSFORM` and hyper-parameters of Eq. 11 for Atari100K in Table 3.

Table 3: An overview of used hyper-parameters for TRANSFORM of CoIT in Atari100K.

Game	TRANSFORM-lr	α	λ
Alien	$5e-07$	0.01	$5e-03$
Amidar	$2e-06$	0.01	$5e-03$
Assault	$2e-05$	$2e-03$	$5e-03$
Asterix	$2e-05$	$1e-03$	$5e-03$
Bank Heist	$2e-05$	0.01	$5e-03$
Battle Zone	$1e-05$	$1e-03$	$5e-03$
Boxing	$5e-07$	$1e-03$	$1e-03$
Breakout	$1e-05$	0.01	$2e-03$
Chopper Command	$1e-05$	0.01	$5e-03$
Crazy Climber	$2e-05$	$2e-03$	0.01
Demon Attack	$1e-06$	$1e-03$	$5e-03$
Freeway	$5e-06$	0.01	$5e-03$
Frostbite	$1e-05$	$2e-03$	$5e-03$
Gopher	$5e-07$	0.01	$5e-03$
Hero	$5e-05$	0.01	$5e-03$
Jamesbond	$2e-04$	0.01	$5e-03$
Kangaroo	$5e-06$	0.01	$5e-03$
Krull	$2e-06$	$5e-04$	$5e-03$
Kung Fu Master	$1e-04$	$1e-03$	0.05
Ms Pacman	$5e-07$	0.01	$5e-03$
Pong	$5e-07$	$5e-03$	0.02
Private Eye	$5e-07$	$1e-03$	$5e-03$
Qbert	$1e-05$	0.02	$5e-03$
Road Runner	$5e-06$	0.01	$5e-03$
Seaquest	$1e-06$	$5e-04$	0.01
Up N Down	$5e-06$	0.01	$5e-03$

C.6 MIXED CONTRASTIVE INVARIANT TRANSFORMATION

To demonstrate the effect of stabilizing the reward function by the Theorem 4.2, we present the results in Figure 5 and Figure 6 for comparisons between two versions of CoIT: (i) *w/o mix*. CoIT with single transformed data and (ii) *mix=2*. We use the transformation to produce 2 transformed data $\mathbf{x}_t^1, \mathbf{x}_t^2$, and mix them after encoding:

$$\mathbf{x}'_t = \eta \cdot \mathbf{x}_t^1 + (1-\eta) \cdot \mathbf{x}_t^2 \tag{25}$$

where η is a parameter randomly sampled from $(0, 1)$ and g_ξ denotes the online encoder.

In most scenarios, they gain similar performance with excellent stability. However, in the situation with sparse reward settings like Cartpole Swingup Sparse, CoIT without mixed data becomes extremely unstable, while CoIT for mixing 2 transformed data still gains outstanding performance.

Table 4: An overview of used hyper-parameters in Atari100K experiments.

Hyperparameter	**Setting**
Image size	$(84, 84)$
Replay buffer capacity	10^5
Training frames	4×10^5
Training steps	10^5
Frame skip	4
Stacked frames	4
Action repeat	4
Replay period every	1
Encoder: channels	32, 64
Encoder: filter size	$5 \times 5, 5 \times 5$
Encoder: stride	$5, 5$
Encoder: hidden dim	256
Momentum (EMA for CoIT)	0.001
Non-linearity	ReLU
Reward Clipping	$[-1, 1]$
Multi-step return	20
Minimum replay size for sampling	1600
Max frames per episode	$108K$
Update	Distributional Double Q
Target Network Update Period	every 2000 updates
Support-of-Q-distribution	51 bins
Mini-batch size	32
Discount γ	0.99
Optimizer	Adam
Optimizer: learning rate	10^{-4}
Optimizer: $\beta 1$	0.9
Optimizer: $\beta 2$	0.999
Optimizer ϵ	0.000015
Max gradient norm	10
Exploration	Noisy Nets
Noisy nets parameter	0.1
Priority exponent	0.5
Priority correction	$0.4 \rightarrow 1$
Critic Q-function soft-update rate τ	0.01
Similarity dim.	128

Table 5: An overview of used hyper-parameters in DeepMind Control Suite experiments.

Hyperparameter	**Setting**
Image size	$(84, 84)$
Replay buffer capacity	10^6
Stacked frames	3
Action repeat	Hopper Hop:4
	2
Seed frames	4000
Exploration steps	2000
n-step returns	3
Mini-batch size	256
Discount γ	0.99
Optimizer	Adam
Learning rate	10^{-4}
Augmentation learning rate	Hopper Hop: 1×10^{-6}
	2×10^{-6}
Agent update frequency	2
Critic Q-function soft-update rate τ	0.01
Momentum τ_m	0.0001
α	0.01
λ	0.005
Features dim.	50
Hidden dim.	1024
Similarity dim.	128
Exploration stddev. clip	0.3
Exploration stddev. schedule	linear$(1.0, 0.1, 100000)$ for 1M frames
	linear$(1.0, 0.1, 500000)$ for 3M frames

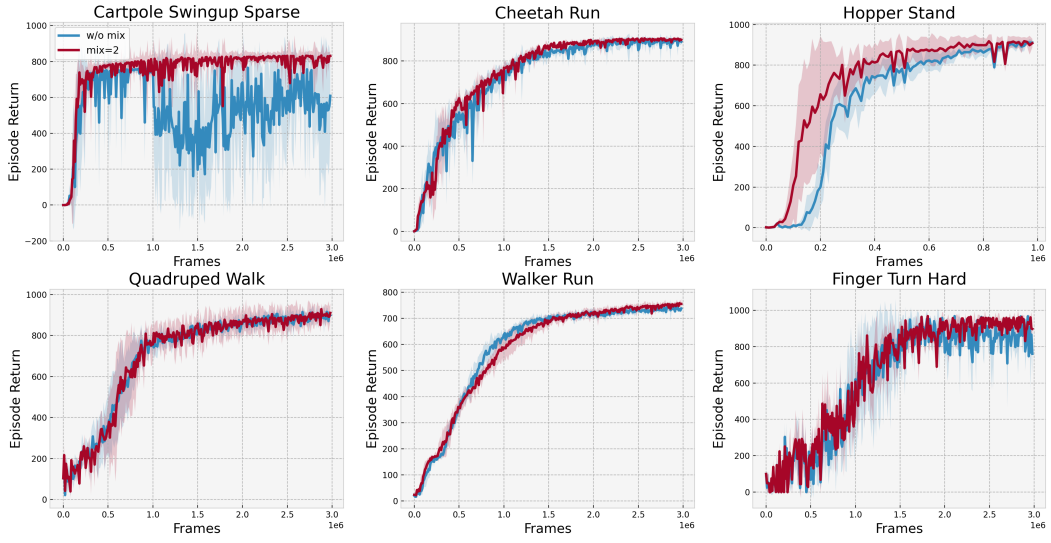

Figure 5: Comparisons between CoIT and its variants of without mix on the DMControl.

This appearance demonstrates that our mixed CoIT greatly stabilizes the training process, especially in sparse reward settings.

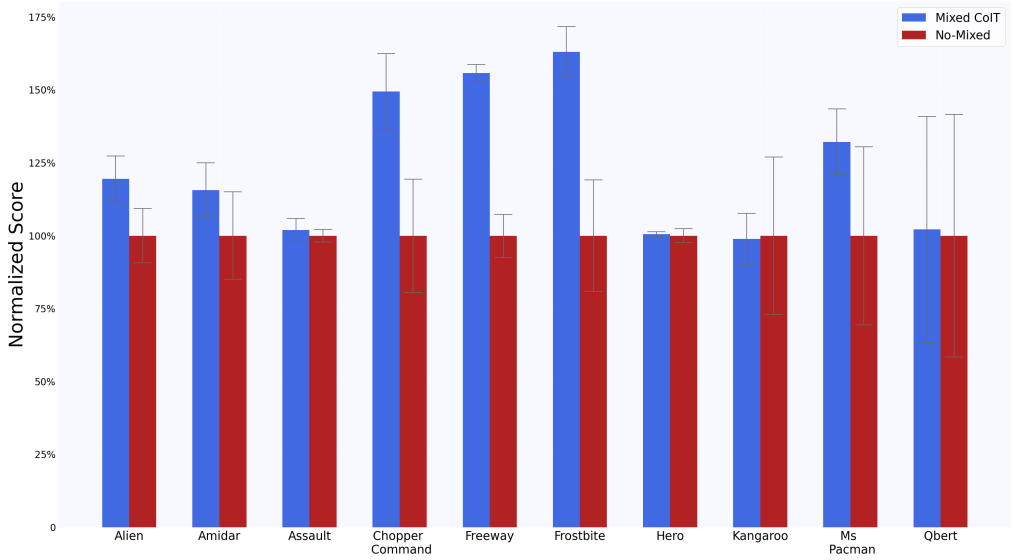

Figure 6: Comparisons between CoIT and its variants of without mix on Atari100K. To show the improvement of data mixing, we normalized the score of the no-mixed variants and plot the *mean* and *std*. The results are based on 10 runs of the outperformed tasks via CoIT.

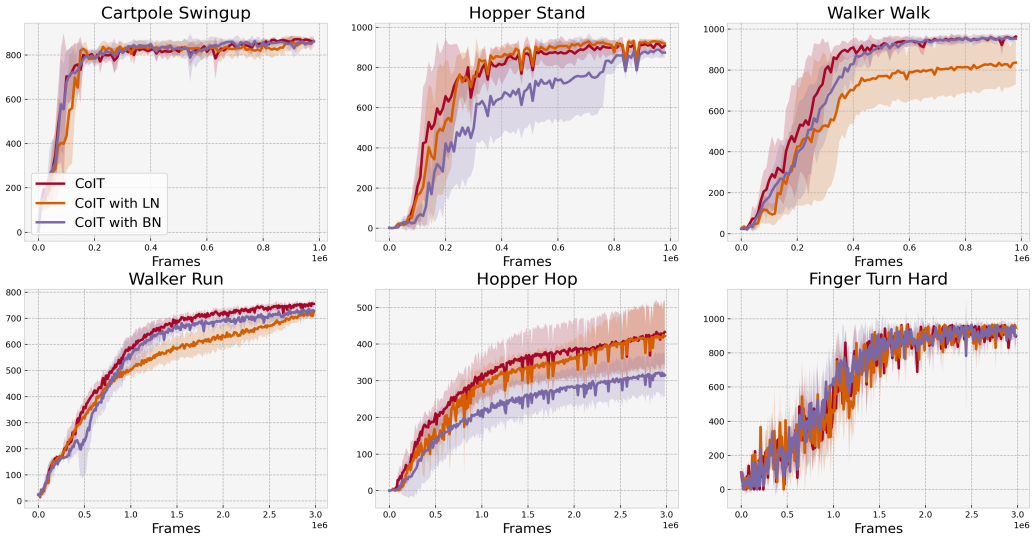

Figure 7: Comparisons between CoIT and its variants with different normalization for projection.

C.7 Types of Normalization for Projection

In this section, we compare CoIT to its two variants with different types of normalization in the projection network for similarity metrics, (i) *CoIT with LN* which using the `LayerNorm` after the first fully-connected layer, (ii) *CoIT with BN* which utilizing the `BatchNorm` after every fully-connected layer. We conduct experiments on 6 pixel-based continuous control tasks in the DMControl and present the results in Figure 7. Here we see that the original CoIT outperforms on each task and the performance of CoIT with BN is pretty close to the original CoIT. There is a score gap between the CoIT with LN and the original CoIT. This indicates that the specifically designed network architecture is beneficial to the RL performance.

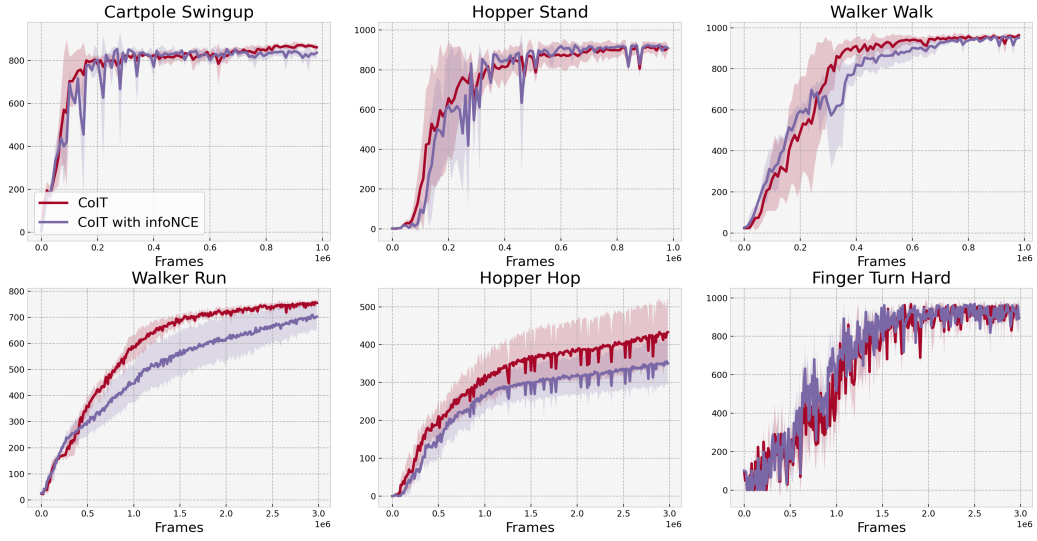

Figure 8: Comparisons between CoIT and its variants with different contrastive loss.

C.8 Types of Similarity Metric

We choose different types of contrastive loss for similarity metric and present results in Figure 8. We utilize the infoNCE loss which is widely used in self-supervised learning (He et al., 2020; Chen et al., 2020) and conduct a variant called *CoIT with infoNCE*. As shown in Figure 8, the original which optimizes the contrastive loss proposed in Grill et al. (2020) outperforms on each task in the DMControl. This indicates that naively making negative samples dissimilar may not be soundness.

C.9 Effects of Different Components in Object Function

Here we conduct experiments to study the effects of different components in object function for CoIT on extra 4 tasks in Figure 9. As mentioned before, we divide CoIT into 4 versions and show the performance impact of two auxiliary losses in CoIT separately: (i) *Critic*. Transformation is only updated with the critic. (ii) *X-stats & Critic*. Transformation is updated by critic and $\mathcal{K}_\omega(\mathbf{x}'_t)$ together. (iii) *H-dist & Critic*. Transformation is updated by critic and $\mathcal{L}_{\xi,\bar{\xi},\omega}$ together. (iv) *Unified Objective*. The transformation will be updated by all components.

From the curves, we demonstrate that both $\mathcal{K}_\omega(\mathbf{x}'_t)$ and $\mathcal{L}_{\xi,\bar{\xi},\omega}$ are sufficient for data-efficiency improvement and combining them together achieves remarkable performance and stability. We also find that *Critic* is facing performance degradation and is hard to improve the asymptotical performance in some domains. We consider that during the end of training the Gaussian distribution $\mathcal{G}_t(\mu_t, \sigma_t)$ converges to a small boundary, thus the produced virtual data is augmented in a slight amplitude. This makes the CoIT run into a situation similar to vanilla SAC & DDPG: updating by reward signal purely is quite unfavorable for representation learning as the agent focuses on slight reward-relevant information. For this reason, we introduce $\mathcal{K}_\omega(\mathbf{x}'_t)$ and $\mathcal{L}_{\xi,\bar{\xi},\omega}$ to assist the agent to learn meaningful representations and stabilize the reward function.

C.10 Saliency Maps

To better present the representations learned by CoIT, we present the saliency maps on the augmented data (e.g., *Random Shift*) of the encoder for 6 tasks on the DMControl in Figure 10.

These saliency maps demonstrate that CoIT is beneficial for the agent to focus on task-relevant elements like the whole robot body and ignore the task-irrelevant information like the floor and background. What's more, the agent trained with CoIT pays high attention to the reward-relevant signals while still taking note of other components that are useful for the task at hand. In Finger Turn Hard, for instance, the lightest part in the saliency map is a red ball in the observation, which

Figure 9: Experiments for studying the effects of different components in object function.

is highly related to the reward. However, the agent also focuses on the robot's body for reasonable action taking. This appearance inspires us that excellent representation learning is a trade-off: the agent needs to figure out which elements are highly related to the reward while still concerned with as much information as possible.

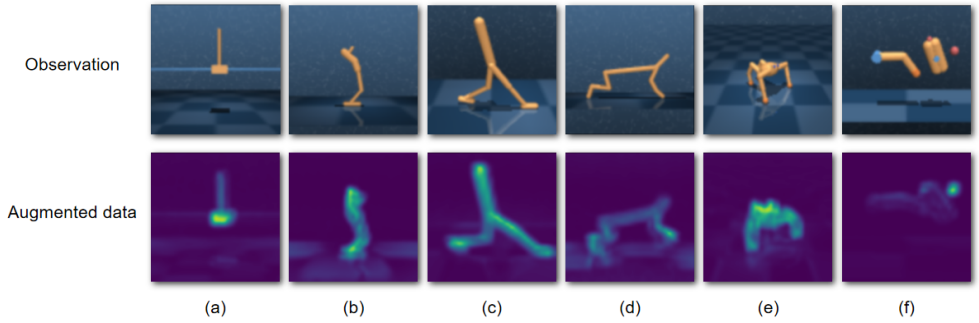

Figure 10: Saliency maps for CoIT on 6 tasks: (a) Cartpole Swingup Sparse, (b) Hopper Stand, (c) Walker Run, (d) Cheetah Run, (e) Quadruped Walk, and (f) Finger Turn Hard.

D Connection to Batch Normalization

Batch normalization (BatchNorm) aims to alleviate the shift caused by the randomness of input data for neural network layers, which is called as internal covariate shift. In particular, the change of *means* and *variances* of the distributions in each layer during mini-batch training creates the problem that parameters are sensitive to the initialization and required to consistently adapt to the distributions. Normalization by batch statistics employs the empirical mean and variance on the feature scalar,

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i, \quad \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{x}_i - \mu_B\right)^2 \tag{26}$$

where B denotes a mini-batch of size m in training. Then, the feature dimension of each layer is normalized by,

$$\hat{\mathbf{x}}_i^{(k)} = \frac{\mathbf{x}_i^{(k)} - \mu_B^{(k)}}{\sqrt{{\sigma_B^{(k)}}^2 + \epsilon}} \tag{27}$$

where ϵ is a small constant added for numerical stability, $k \in [1, d]$ is the index of feature dimensions d, and i is the index of data points. A counterpart procedure of the inner batch normalization is

whitening which is non-differential everywhere making the mean of 0 and the variance of 1 for the distribution. In turn, given the distribution with mean μ_B and variance σ_B^2, it is straightforward that the deviation given by $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|$ is associated with the value of $\mathbf{x}_i$ and the collection of $w_B = \{\mu_B, \sigma_B\}$.

Different with the goal of whitening procedure with 0-mean and 1-variance, the automatic data augmentation needs to find an optimal interpolation $\mathbf{x}'$ obey certain distribution with unknown mean μ and variance σ^2. As the given environment is non-stationary, the distribution for data generation is shifted with respect to the parameter sets $w_t = \{\mu_t, \sigma_t\}$, where t denotes a timestep. To cope with the assumption of a stationary environment using the replay buffer in off-policy methods, the automatic data augmentation allows a new dynamic distribution controlling by the model-free RL. In other words, the dynamics of the normalization derive from the observation distribution conditioned on the changed environment.

Concretely, the augmented data $\mathbf{x}' \sim \hat{\mathcal{D}}'$ is parameterized through the Gaussian distribution $\mathcal{G}(\mu, \sigma)$, as the Algorithm 1 shown. Similar to BatchNorm, $\|\mathbf{x} - \mathbf{x}'\|$ is determined by the value of the original image $\mathbf{x}$ and the dynamically changed $\{\mu_t, \sigma_t\}$. As a consequence, a normalized shift distribution conditioned on the values of the replay buffer is obtained for the current training round.