SHIFTNORM: ON DATA EFFICIENCY IN REINFORCEMENT LEARNING WITH SHIFT NORMALIZATION

Sicong Liu[1,2,3], **Xi Sheryl Zhang**[2,3], **Yushuo Li**[2], **Yifan Zhang**[2,3] **and Jian Cheng**[2,3]
[1]Nanjing University of Science and Technology
[2]Institute of Automation, Chinese Academy of Sciences
[3]Nanjing Artificial Intelligence Research of IA
{sicongliu1014;sheryl.zhangxi}@gmail.com
yushuo.li@ia.ac.cn
{yfzhang;jcheng}@nlpr.ia.ac.cn

ABSTRACT

We propose ShiftNorm, a simple yet promising data augmentation that can be applied to standard model-free algorithms to improve data-efficiency in high-dimensional observation-based reinforcement learning (RL). Concretely, the differentiable ShiftNorm leverages original samples with reparameterized virtual samples, and hasten the image encoder to generate invariant representations. Our approach demonstrates certify substantial advances, enabling it to outperform the new state-of-the-art on 8 of 9 tasks on the DeepMind Control Suite at 500k steps.

1 INTRODUCTION

Improving agent's performance and data-efficiency have always been a key problem in visual reinforcement learning (RL). Unlike the representation learning in supervised learning, as the model has a strongly supervised singal and various methods could be applied to learn representations useful to the task, there are not enough supervised signals in visual RL, thus the trainging process is quite fragile. The network need to learn useful representations for performance improvement, while inappropriate methods may do damage to the training process, causing performance degradation. In this case, we urgently need representation learning methods suitable for reinforcement learning. Previous works have demonstrated that data augmentation could better this situation, both for sample-efficiency (Laskin et al., 2020a; Yarats et al., 2020; 2021) and generalization (Hansen et al., 2021; Raileanu et al., 2020; Zhang & Guo, 2021; Hansen & Wang, 2021; Fan et al., 2021). This method without further modifications to the backbone of RL algorithms is being noticed by others.

Inspired by Spatial Transformer Networks (STN) (Jaderberg et al., 2015), we hold the point that the observation transformation process can be parameterized. We focus on transforming handcrafted perturbations into an optimizeable process and propose ShiftNorm to improve the data-efficiency at pixel-based tasks. We hypothesize that there is a suitable transformation of the observation for the agent, thus the learned transformation can enable the encoder to abstract more useful semantic representations from high-dimensional obsevations, and control algorithms based on these representations should be more sample-efficiency. Following this way, we propose ShiftNorm to improve the data-efficiency at pixel-based tasks. Through this process, the pixel shift in different degrees and directions will occur in the whole image and the *shift* controls the level of perturbation of the image. Here we raise our ideas:

Can we reparameterize this shift procedure by sampling it from a dynamic distribution to cope with the assumption of stationary environment in model-free RL training?

To this aim, we parameterize the mean and variance of the distribution and update them with the RL process. Two constraints are also proposed to ensure that the automatic augmentation will not ruin the convergence of the RL algorithm. As the iterations increase, the agent will find the shift distribution suitable for the task at hand.

Key Contributions: We introduce ShiftNorm, a reparameterized data shift method to integrate invariant representations with model-free RL methods. We evaluate our algorithm on 9 control tasks from the DM control suite and show that the optimizable transformation is effective to improve data-efficiency for visual RL.

2 THE SHIFTNORM

2.1 PROBLEM FORMULATION

We formulate the visual RL as an infinite-horizon Markov Decision Process (MDP) (Bellman, 1957). The MDP $\mathcal{M}$ can be described as a 5-tuple $\langle \mathcal{O}, \mathcal{S}, \mathcal{A}, r, \gamma \rangle$. Here $\mathcal{O}$ consists of a stack of images (Mnih et al., 2013). The state space $\mathcal{S}$ is either observable or unobservable (Silver et al., 2017; Zhang et al., 2020). $\mathcal{A}$ is the action space for the agent. The goal is to maximize the cumulative rewards $\mathcal{R} = \sum_t \gamma^t r_t$, where $\gamma \in [0, 1)$ is the discount factor and r_t denotes the reward at time t.

2.2 LEARNABLE INVARIANT TRANSFORMATION

We first introduce the optimal invariant metric to reach the stationary distribution $\mathcal{D}$ over the augmented context $\mathbf{x}'$, where $\mathbf{x}'$ is required to satisfy $\mathbf{x}' \sim q(\cdot|\mathbf{e})$. Environment transition $\mathbf{e}$ belongs to the replay distribution $\mathcal{D}$. Below are the definition:

Definition 2.1. (Optimal Invariant Metric). Given a transition distribution $\mathcal{D}$, suppose the block structure assumption holds, the shift between $\mathbf{x}$ and $\mathbf{x}'$ can be measured by a conditional divergence:

$$d(\mathbf{x}, \mathbf{x}'|\mathbf{e}) \triangleq \mathbb{E}_{e\sim\mathcal{D}} \left[d_{KL} \left(q(\mathbf{x}|\mathbf{e} = e), q(\mathbf{x}'|\mathbf{e} = e) \right) \right] = \sum_{\mathbf{e}} p(\mathbf{e}) \sum_{\mathbf{x}} \sum_{\mathbf{x}'} q(\mathbf{x}|\mathbf{e}) \log \frac{q(\mathbf{x}|\mathbf{e})}{q(\mathbf{x}'|\mathbf{e})} \tag{1}$$

Following the Bayes' rule on the conditional distribution, Eq.(1) can be rewritten as:

$$\mathbb{E}_{\mathbf{e}} \left[d_{KL} \left(q(\mathbf{x}|\mathbf{e}), q(\mathbf{x}'|\mathbf{e}) \right) \right] = \mathbb{E}_{\mathbf{e}|\mathbf{s}} \left[d_{KL} \left(p(\mathbf{s}|\mathbf{x}) p(\mathbf{x}), p(\mathbf{s}|\mathbf{x}') p(\mathbf{x}') \right) \right] \tag{2}$$

where $d_{KL}(\cdot, \cdot)$ is the Kullback–Leibler (KL) divergence. Therefore, minimizing the conditional divergence leads to encoding $\mathbf{x}$ and $\mathbf{x}'$ into an invariant latent state space $\mathcal{S}$.

Now we define a non-trivial function $g : \mathcal{O} \to \mathcal{S}$ mapping from the observed state $\mathcal{O}$ to the latent state $\mathcal{S}$ such that $g(\mathbf{x}) = p(\mathbf{s}|\mathbf{x}), \forall \mathbf{x}$. Since the pixel transformation is reparameterized as $\nu(\mathbf{x}, \mathcal{G})$ and could drift away, it is natural to find another state encoder $g'(\mathbf{x}') = p(\mathbf{s}|\nu(\mathbf{x}, \mathcal{G})), \forall \mathbf{x}'$, and this encoding function g' should be different with the original observation encoder g. So far, the learning goal boils down to minimizing the distance between $g(\mathbf{x})$ and $g'(\mathbf{x}')$,

Definition 2.2. (State 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

$$\mathbb{E}_{\mathbf{x}\sim\hat{\mathcal{D}}} \left[d(g(\mathbf{x}), g'(\mathbf{x})) \right] \leq \epsilon \tag{3}$$

Define the distance $d(\cdot, \cdot)$ as l_p-norm. If g' satisfies L_p-Lipschitzness according to Assumption B.1, the distance between the $\mathbf{s}$ and $\mathbf{s}'$ can be expressed as the following triangular inequality,

$$d(g(\mathbf{x}), g'(\mathbf{x}')) \leq \underbrace{d(g(\mathbf{x}), g'(\mathbf{x}))}_{\text{state } \epsilon\text{-approximation}} + \underbrace{d(g'(\mathbf{x}), g'(\mathbf{x}'))}_{L_g\text{-Lipschitzness}} \tag{4}$$

Minimizing the right side of the inequality is able to upper bound our problem.

To restrict the functional similarity, we combine the momentum updating (He et al., 2020) with a projection $f : \mathcal{S} \to \mathcal{Y}$ (Chen et al., 2020a) and minimize the distance in the projected space $\mathcal{Y}$ for model-free RL. Suppose the Markov chain $\mathcal{O} \xrightarrow{g} \mathcal{S} \xrightarrow{f} \mathcal{Y}$ holds. We use $f \circ g$ to denote the function composition $f(g(\cdot))$. Also, there exists the assumption of Lipschitzness for the projection f. Before present data shifted method, we leverage the convexity in momentum updating paradigm.

Lemma 2.1. 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.

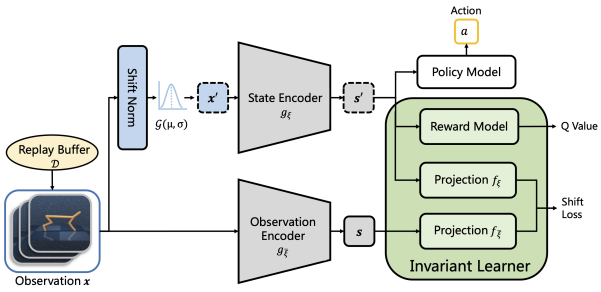

Figure 1: Overall architecture of ShiftNorm. Observations are augmented that follows a Gaussian distribution $\mathcal{G}(\mu, \sigma)$. $g_{\bar{\xi}}$ and $f_{\bar{\xi}}$ are the momentum averaged version of g_ξ and f_ξ.

Lemma 2.2. Given the dynamical updating: $\bar{\xi}^t = (1-\tau_m)\bar{\xi}^{t-1} + \tau_m \xi^t$. By Lemma 2.1, $f_\xi = f_{\bar{\xi}}$ 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 $\min \mathbb{E}_{\mathbf{x}}[\|g_\xi(\mathbf{x}) - g_{\bar{\xi}}(\mathbf{x})\|]$.

We then provide helpful insight for learning an optimal shifted data together with the encoders.

Theorem 2.1. (Shift Normalization.) Suppose Assumption B.1 hold 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 $\nu(\mathbf{x}, \mathcal{G})$, optimizing the conditional divergence in Definition 2.1 means to minimize the upper bound as follows, where $C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$ is a constant.

$$\mathbb{E}_{\mathbf{x}\sim\hat{\mathcal{D}}}\left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbf{x}'))\right] \leq L_g \left(CL_f + \|\xi_f\|\right) \mathbb{E}_{\mathbf{x}\sim\hat{\mathcal{D}}}\left[\|\mathbf{x} - \mathbf{x}'\|\right] \tag{5}$$

The above theorem suggests two analysis results. First, minimizing the representations of the projected layer $\in \mathcal{Y}$ is useful for encoding the optimal invariant state. Second, the pixel transformation needs to be regularized with the observation such that the divergence in upper bounded. Next, we will place emphasis on the illustration of this shift via a reparameterization method.

2.3 Reparameterizable Observation

For the bound of Theorem2.1, the Algorithm 1 in appendix is a procedure defined on the MDP $\mathcal{M}$ with Gaussian random variables $\mathcal{G}_0 \sim \mathcal{G}^{|\mathcal{O}|}$ for initialization. The TRANSFORM is fulfilled by aforementioned pixel transformation ν. To parameterize the augmentation, we add a shift subject to $\mathcal{G}_t(\mu_t, \sigma_t)$ on the bilinear interpolation. Therefore, the output will have different levels of pixel shift.

2.4 Stabilizing Reward Function

Reparametrizing the underlying invariant optimization plays a key role to smooth the distribution $\mathcal{D}$ in the replay buffer. We sample multiple augmentation data following $\mathcal{G}(\mu, \sigma)$ and then mixup the learned hidden state $\mathbf{s}'$ for further stabilization.

Theorem 2.2. (Mixed Shift Normalization.) Suppose Assumption B.1 hold for functions g_ξ, $g_{\bar{\xi}}$, f_ξ, and $f_{\bar{\xi}}$, respectively. 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, where $C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$.

$$\mathbb{E}_{\mathbf{x}\sim\hat{\mathcal{D}}}\left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbb{E}_{\mathbf{x}'\sim\mathcal{G}}[\mathbf{x}']))\right] \leq L_g \left(CL_f + \|\xi_f\|\right) \mathbb{E}_{\mathbf{x}\sim\hat{\mathcal{D}}}\mathbb{E}_{\mathbf{x}'\sim\mathcal{G}}\left[\|\mathbf{x} - \mathbf{x}'\|\right] \tag{6}$$

The proof of Theorem 2.2 is straightforward based on Jensen's inequality and the Theorem 2.1.

2.5 Automatic Manipulated Observation

With theoretical analysis on invariant transformations, we presented a new framework with normalization variants to ensure our discussed learning guarantees.

Critic oriented. We switch the shift into a parameterized Gaussian distribution. The mean and variance will participate in optimizing the objective function of the critic network, where $s(\mathbf{x}'_t)$ and $s(\mathbf{x}'_{t+1})$ are states embedded by encoder and ω represents the transformation.

$$J_Q(\theta, \omega) = \left(Q_\theta(\mathbf{x}'_t, \mathbf{a}_t) - r - \gamma Q_{\bar{\theta}}(\mathbf{x}'_{t+1}, \pi(\cdot|\mathbf{x}'_{t+1}))\right)^2 \tag{7}$$

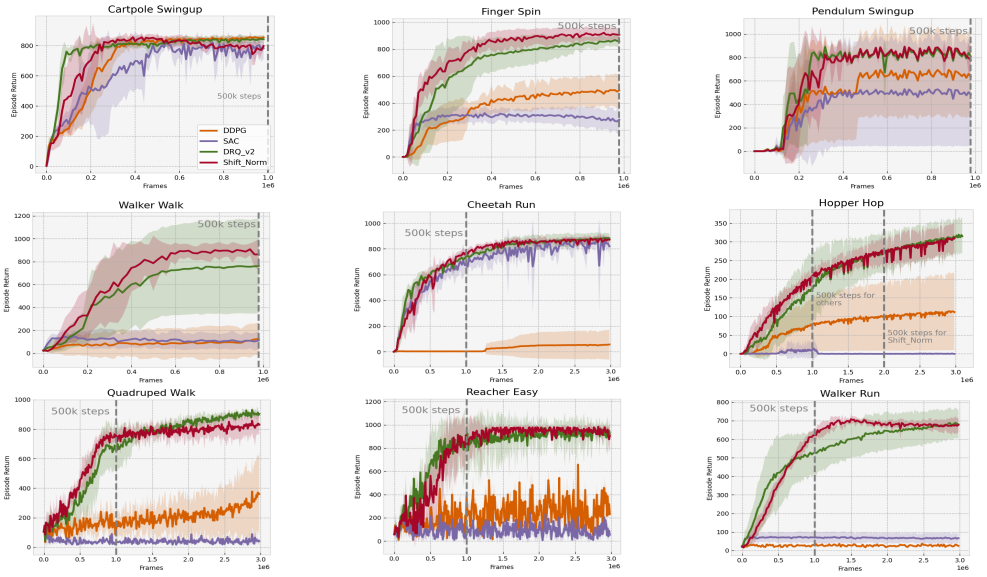

Figure 2: Results of 9 complex tasks in DM control suite. Our method demonstrates improvement on sample-efficiency and performance over tasks on 8 out 9 selected tasks

Hidden representation oriented. For similarity measurement, we use the loss proposed in BYOL. ξ and $\bar{\xi}$ represent the parameters of online encoder $g_\xi \circ \mathrm{f}_\xi$ and momentum encoder $g_{\bar{\xi}} \circ \mathrm{f}_{\bar{\xi}}$ respectively.

$$\mathcal{L}_{\xi,\bar{\xi},\omega}(\mathcal{D}) \triangleq \left\| f_\xi(g_\xi(\mathbf{x}'_t)) - f_{\bar{\xi}}(g_{\bar{\xi}}(\mathbf{x}_t)) \right\|_2^2 = 2 - 2 \cdot \frac{\left\langle f_\xi(g_\xi(\mathbf{x}'_t)), f_{\bar{\xi}}(g_{\bar{\xi}}(\mathbf{x}_t)) \right\rangle}{\left\| f_\xi(g_\xi(\mathbf{x}'_t)) \right\|_2 \cdot \left\| f_{\bar{\xi}}(g_{\bar{\xi}}(\mathbf{x}_t)) \right\|_2} \tag{8}$$

Batch statistics oriented. The distribution shift between the transformed samples and the overall data can be measured by the following formulation, where $\tilde{\mu}(\mathbf{x}'_t)$ and $\tilde{\sigma}^2(\mathbf{x}'_t)$ is the mean and variance corresponding to the l-th convolution layer, X is the overall observations.

$$\mathcal{R}_\omega(\mathbf{x}'_t) = \sum_l \left\| \tilde{\mu}_l(\mathbf{x}'_t) - \mathbb{E}(\tilde{\mu}_l(\mathbf{x})|X) \right\|_2 + \sum_l \left\| \tilde{\sigma}_l^2(\mathbf{x}'_t) - \mathbb{E}(\tilde{\sigma}_l^2(\mathbf{x})|X) \right\|_2 \tag{9}$$

Architectural overview. We summarize the objective function of the transformation below. α and λ as hyperparameters represent the magnitude of the constraints.

$$J_{\theta,\xi,\omega}(\mathcal{D}) = J_Q(\mathcal{D}) + \alpha \mathcal{R}_\omega(\mathcal{D}) + \lambda \mathcal{L}_{\theta,\xi,\omega}(\mathcal{D}) \tag{10}$$

3 Experiments

We compare ShiftNorm with prior model-free methods on 9 visual tasks from the DM control suite. and results are presented in Figure 2. Below are key findings: (i) Compared to vanilla DDPG and SAC, ShiftNorm gains outstanding results and outperform in a wide range. (ii) When compared with DrQ-v2 which has already performed remarkable for continuous control, we improve the sample-efficiency on multiple tasks. (iii) We also find that ShiftNorm has better stability during training while still keep convincing performance. More details and ablation studies can be found in appendix.

4 Conclusion

We introduce a simple automatic transformation for model-free RL algorithm for visual continuous control tasks. We proposed an auxiliary loss to improve the performance and training stability at the same time. Our method achieves convincing performance without specific architectural selection compared to SOTA approachs on DeepMind control suite. We hope that our method can promote the progress of automatic augmentation and representation learning in RL.

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.

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

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.

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, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

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.

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

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.

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.

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.

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.

Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 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.

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.

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.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

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.

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.

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, 2020.

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

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

$$\begin{aligned} &h(\tau\xi_1 + (1-\tau)\xi_2) = f(< \tau\xi_1 + (1-\tau)\xi_2, \mathbf{s} >) \\ &= f(< \tau\xi_1, \mathbf{s} > + < (1-\tau)\xi_2, \mathbf{s} >) = f(\tau < \xi_1, \mathbf{s} > +(1-\tau) < \xi_2, \mathbf{s} > \\ &\leq \tau f(< \xi_1, \mathbf{s} >) + (1-\tau) f(< \xi_2, \mathbf{s} >) = \tau h(\xi_1) + (1-\tau) h(\xi_2) \end{aligned} \tag{11}$$

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 2.1, $f_\xi = f_{\bar{\xi}}$ 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 $f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x})$],

$$\begin{aligned} \mathbb{E}_{\mathbf{x}}[\|f_\xi \circ g_\xi(\mathbf{x}') - f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x})\|] &= \mathbb{E}_{\mathbf{x}}[\|f_\xi \circ (g_\xi(\mathbf{x}') - g_{\bar{\xi}}(\mathbf{x}))\| \\ &\leq \mathbb{E}_{\mathbf{x}}[\|\xi_f\| \|(g_\xi(\mathbf{x}') - g_{\bar{\xi}}(\mathbf{x}))\|] \\ &= \mathbb{E}_{\mathbf{x}}[\|(g_\xi(\mathbf{x}') - g_{\bar{\xi}}(\mathbf{x}))\|] \|\xi_f\| \end{aligned} \tag{12}$$

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–Schwarz 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.(12). When the norm of $\|\xi_f\|$ is fixed, the proof completes. □

Theorem A.1. (Shift Normalization.) Suppose Assumption B.1 hold 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 $\mathbf{x}'$, optimizing the conditional divergence in Definition 2.1 means to minimize the upper bound as follows,

$$\mathbb{E}_{\mathbf{x}}\left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_\xi \circ g_\xi(\mathbf{x}'))\right] \leq L_g \left(C L_f + \|\xi_f\|\right) \mathbb{E}_{\mathbf{x}}\left[\|\mathbf{x} - \mathbf{x}'\|\right] \tag{13}$$

where $C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$ is a constant.

Proof. According to the following triangular inequality,

$$d(g_{\bar{\xi}}(\mathbf{x}), g_\xi(\mathbf{x}')) \leq \underbrace{d(g_{\bar{\xi}}(\mathbf{x}), g_\xi(\mathbf{x}))}_{\text{state } \epsilon\text{-approximation}} + \underbrace{d(g_\xi(\mathbf{x}), g_\xi(\mathbf{x}'))}_{L_g\text{-Lipschitzness}} \tag{14}$$

The distance $d(\cdot, \cdot)$ is set as l_p-norm for simplicity. By incorporating Lemma A.2 into the left side of Eq.(14), it leads to a divergence with projections where a new triangular inequality holds,

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

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 assumption is employed. Suppose the updating has achieved convergence, Eq.(18) turns to the following inequality,

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

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 Shift Normalization.) Suppose Assumption B.1 hold 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,

$$\mathbb{E}_{\mathbf{x} \sim \hat{\mathcal{D}}} \left[d(f_{\bar{\xi}} \circ g_{\bar{\xi}}(\mathbf{x}), f_{\xi} \circ g_{\xi}(\mathbb{E}_{\mathbf{x}' \sim \mathcal{G}}[\mathbf{x}'])) \right] \leq \quad L_g \left(C L_f + \|\xi_f\| \right) \mathbb{E}_{\mathbf{x} \sim \hat{\mathcal{D}}} \mathbb{E}_{\mathbf{x}' \sim \mathcal{G}} \left[\|\mathbf{x} - \mathbf{x}'\| \right] \tag{22}$$

where $C = \frac{1+\tau}{1-\tau}, \tau = 1 - \tau_m$.

Proof. Based on Theorem A.1, we can view the mixup $\mathbb{E}_{\mathbf{x}'}[\mathbf{x}']$ as a sort of augmented 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 TECHNICAL TOOLS

Assumption B.1. (Lipschitzness). Let the encoding function $g(\mathbf{x}) : \mathbb{R}^{|\mathcal{O}|} \to \mathbb{R}^{|\mathcal{S}|}$ is L_g-Lipschitz, we have that $\forall \mathbf{x}, \mathbf{x}'$

$$|g(\mathbf{x}') - g(\mathbf{x})| \leq L_g \|\mathbf{x}' - \mathbf{x}\|$$

C Related Work

Data augmentation. Training agents from high-dimensional images data has always been highly concerned (Mnih et al., 2013; Yarats et al., 2019; Hafner et al., 2019; Lee et al., 2019). Previous work has shown the great potential of solving the visual RL tasks by data augmentation. Many results show that even the simplest augmentation method can greatly improve the agent's sample efficiency and asymptomatic performance (Laskin et al., 2020a; Yarats et al., 2020; 2021) without further modifications to the backbone RL algorithm.

Self-supervised learning. Recent years have witnessed the self-supervised learning methods achieving huge success in representation learning (Chen et al., 2020a; He et al., 2020; Caron et al., 2020; Grill et al., 2020; Chen et al., 2020b). For SSL combined with RL, CURL (Laskin et al., 2020b), ReLIC (Mitrovic et al., 2020) and CoBERL (Banino et al., 2021) focus on compute the consistent between positive samples. On the other hand, Mazoure et al. (2020) maximized the mutual information between the nearby states, while PSM (Agarwal et al., 2021) measures behavioral similarity between states through optimal policies similarity with future state transition probability.

D Benchmarks.

We follow the settings in DrQ-v2 (Yarats et al., 2021) and classify these tasks into *easy* and *challenging* and provide a summary for each task in Table 1.

Table 1: A detailed description of each tasks in our *easy*, and *challenging* benchmarks.

Task	Traits	Difficulty	Allowed Steps	$\dim(\mathcal{S})$	$\dim(\mathcal{A})$
Cartpole Swingup	swing, dense	easy	1×10^6	4	1
Finger Spin	rotate, dense	easy	1×10^6	6	2
Pendulum Swingup	swing, sparse	easy	1×10^6	2	1
Walker Walk	walk, dense	easy	1×10^6	18	6
Cheetah Run	run, dense	challenging	3×10^6	18	6
Hopper Hop	move, dense	challenging	3×10^6	14	4
Quadruped Walk	walk, dense	challenging	3×10^6	56	12
Reacher Easy	reach, dense	challenging	3×10^6	4	2
Walker Run	run, dense	challenging	3×10^6	18	6

E PSEUDOCODE

Algorithm 1 Reparameterized Data Manipulation

Initialization: Generate an initial distribution $\mathcal{G}_0 \sim \mathcal{G}^{|\mathcal{O}|}$ with given mean μ_0 and variance σ_0, $\mathcal{R} = 0$.
Training:
for each timestep t in $0, \cdots, T$ **do**
  $\mathbf{x}'_t = \text{TRANSFORM}(\mathbf{x}_t, \mathcal{G}_t)$
  $\mathcal{R} = \mathcal{R} + \gamma^t r(\mathbf{x}'_t, \mathbf{a}_t)$
  Adjust to an optimal $\mathcal{G}_t(\mu_t, \sigma_t)$
end for

Algorithm 2 ShiftNorm
Similarity metric. Learnable transformation.

Inputs:
Encoder g_ξ, policy π_ϕ, Q-functions Q_{θ_1}, Q_{θ_2}, OBSERVATION encoder $g_{\bar{\xi}}$, MLP f_ξ, OBSERVATION MLP $f_{\bar{\xi}}$
$\mu, \sigma \sim \mathcal{G}$ 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**
  $\tilde{\sigma}_t \leftarrow \tilde{\sigma}(t)$
  $\mathbf{x}'_t \leftarrow \text{TRANSFORM}(\mathrm{x}_t, \mathcal{G}_t)$ 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$(\mathcal{D}, \tilde{\sigma})$
  $\{(\mathbf{x}_t, \mathbf{a}_t, r_{t:t+n-1}, \mathbf{x}_{t+n})\}_{i=1}^N \sim \mathcal{D}$
  $\mathbf{x}'_t, \mathbf{x}'_{t+n} \leftarrow \text{TRANSFORM}(\mathrm{x}_t, \mathcal{G}_t), \text{TRANSFORM}(\mathrm{x}_{t+n}, \mathcal{G}_t)$
  $\mathbf{s}_t, \mathbf{s}_{t+n} \leftarrow g_\xi(\mathbf{x}'_t), g_\xi(\mathbf{x}'_{t+n})$
  $s_{\bar{\xi}} \leftarrow g_{\bar{\xi}}(x_t)$
  Measure similarity by $\mathcal{L}_{\xi,\bar{\xi},\omega}$
  $\mathbf{a}_t \leftarrow \pi_\phi(\mathbf{s}_{t+n}) + \tilde{\epsilon}$ and $\tilde{\epsilon} \sim \mathcal{G}(0, \tilde{\sigma}_t)$
  Compute $J_{\theta,\omega}(\mathcal{D})$ for Q_θ and TRANSFORM updating
  $\mu \leftarrow \mu - \delta_{\text{aug}} \nabla_\mu (J_{\theta,\omega}(\mathcal{D}))$
  $\sigma \leftarrow \sigma - \delta_{\text{aug}} \nabla_\sigma (J_{\theta,\omega}(\mathcal{D}))$
  $\xi \leftarrow \xi - \delta \nabla_\xi J_{\theta,\omega}(\mathcal{D})$
  $\theta \leftarrow \theta - \delta \nabla_\theta J_{\theta,\omega}(\mathcal{D})$
  $\hat{\theta} \leftarrow (1-\tau)\hat{\theta} + \tau\theta$
  $\bar{\xi} \leftarrow (1-\tau_m)\bar{\xi} + \tau_m \xi$
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

F EXPERIMENTS

F.1 SETUP

Environments. The DeepMind control suite Tassa et al. (2018) is a popular benchmark that has been widely used in prior algorithms. This benchmark is built on MuJoCo Todorov et al. (2012) and contained several robot control tasks, which provides different difficulties.

Following previous works, we set our observations as stacks of 3 consecutive images. The size of the RGB images is 84×84, and we concatenate 3 of them at the dimension of the color channel to ensure that the dynamic and temporal information is fitted into the agent.

Details. We select nine tasks from DM Control with different difficulties (as mentioned by Drq-v2) to test the performance of ShiftNorm both on *sample-efficiency* and *asymptotical performance*. Since each episode is set to a total of 1000 frames in all tasks of DM Control, it is quite reasonable to set the total number of frames experienced during training as x-Axis, so that we can refer to the number of episodes we have gone through to evaluate the tested algorithms. We will also use the settings in DM Control to calculate the reward: a per-frame reward is in the unit interval $[0, 1]$, so each episode will get an episode reward of no more than 1000. For a fair comparison, we refer to the settings of the training episodes for different tasks in DrQ-v2, e.g., more episodes will be trained for hard tasks to facilitate asymptotical performance comparison.

Baselines. We present several baselines, including methods of using data augmentation to improve performance and sample-efficiency to benchmark performance for continuous control on DM Control suite: (i) DrQ-v2 Yarats et al. (2021) where the authors change the backbone RL algorithm from SAC Haarnoja et al. (2018) to a better designed DDPG. In addition, they optimized the details in DrQ Yarats et al. (2020), (ii) Pixel SAC and (iii) Pixel DDPG: Vanilla SAC and DDPG operating purely from images.

Evaluation. To facilitate fair performance comparison all algorithms will be evaluated with the same periodicity of 20000 environment steps and we average over 10 episodes return for each evaluation query. we also use environment steps to measure sample complexity for a well-defined comparison with action repeat of 2.

Hyperparameters. We set the parameters as consistent as possible with the baselines. To prevent premature convergence, the mean and standard deviation in automatic shift augmentation, the learning rate is 2e-6 and the momentum tau is set to 0.0001. Both policy and Q-function networks are trained using Adam optimizer, and the batch size is the same as DrQ-v2 of 256. SAC and DDPG's parameters are following prior algorithms. The detailed description of all tasks will be shown in table 1.

F.2 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 of 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 actor.

F.3 ENCODER NETWORKS

The architecture of encoder is based on 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 use `BatchNorm` Ioffe & Szegedy (2015) after each activations 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 feed 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.

F.4 RESULTS

The curves of 9 complex tasks are in Table 2

Table 2: We evaluate **ShiftNorm** on 9 tasks from the DeepMind control suite at 100k and 500k environment steps, compared with 3 baselines.

500K STEPS SCORES	SHIFTNORM	DRQ-V2	SAC	DDPG
CARTPOLE SWINGUP	783 ± 43	$\mathbf{842 \pm 25}$	776 ± 78	853 ± 12
FINGER SPIN	$\mathbf{910 \pm 49}$	859 ± 43	276 ± 81	488 ± 116
PENDULUM SWINGUP	$\mathbf{831 \pm 24}$	826 ± 12	496 ± 452	652 ± 355
WALKER WALK	$\mathbf{886 \pm 78}$	758 ± 410	106 ± 72	110 ± 137
CHEETAH RUN	$\mathbf{760 \pm 36}$	739 ± 43	687 ± 86	3 ± 2
HOPPER HOP	$\mathbf{271 \pm 37}$	169 ± 96	9 ± 21	75 ± 69
QUADRUPED WALK	$\mathbf{749 \pm 45}$	668 ± 167	36 ± 10	153 ± 80
REACHER EASY	$\mathbf{849 \pm 177}$	835 ± 218	94 ± 57	191 ± 65
WALKER RUN	$\mathbf{616 \pm 31}$	524 ± 118	70 ± 28	26 ± 3
100K STEPS SCORES				
CARTPOLE SWINGUP	607 ± 234	$\mathbf{774 \pm 37}$	447 ± 177	388 ± 120
FINGER SPIN	$\mathbf{620 \pm 103}$	467 ± 317	291 ± 63	228 ± 132
PENDULUM SWINGUP	228 ± 284	$\mathbf{321 \pm 305}$	152 ± 204	172 ± 210
WALKER WALK	$\mathbf{288 \pm 184}$	207 ± 145	130 ± 54	72 ± 93
CHEETAH RUN	247 ± 103	$\mathbf{424 \pm 62}$	301 ± 65	3 ± 2
HOPPER HOP	$\mathbf{91 \pm 37}$	21 ± 29	0 ± 0	2 ± 3
QUADRUPED WALK	$\mathbf{187 \pm 80}$	150 ± 65	47 ± 22	108 ± 31
REACHER EASY	202 ± 91	$\mathbf{255 \pm 90}$	167 ± 80	175 ± 81
WALKER RUN	89 ± 45	$\mathbf{156 \pm 102}$	71 ± 18	29 ± 5

F.5 ABLATION STUDIES

In this section, we present an ablation study to discuss the effects of different terms in shift loss. As mentioned in previous sections, the shift loss function is composed of two parts: $\mathcal{R}_\omega(\mathbf{x}'_t)$ for normalization with observation statistics, and $\mathcal{L}_{\xi,\bar{\xi},\omega}$ for similarity measure. On this basis, we can divide ShiftNorm into 4 versions: (i) **with critic**. Transformation is only updated with critic. (ii) **with xstats & critic**. Transformation is updated by critic and $\mathcal{R}_\omega(\mathbf{x}'_t)$ together. (iii) **with h-dist & critic**. Transformation is updated by critic and $\mathcal{L}_{\xi,\bar{\xi},\omega}$ together. (iv) **with all**. Transformation will be updated by all components. We evaluate all of these versions on 5 tasks from the DeepMind control suite and present the results in Figure 3.

Compared **with critic** with other methods, we demonstrate that both of the components in shift loss has improved the performance. Though the version of **with critic** has the ability to solve most of the tasks and improve the data-efficiency, it may suffer from weak supervision signals and sink into suboptimal cases. However, once the constraints of shiftnorm are added, the performance has been improved in a degree compared with the version of **with critic**. If both terms are used, the performance can lead ahead of all tasks, rather than task-specific. We can also figure out from the shades that the curve with complete constraints achieves better stability (i.e. smaller variance), which shows that representation learning may be closely related to policy learning. This result demonstrates that even if the automatic transformation can enable agents to learn features in a more appropriate way, such a process is often uncontrollable. Introducing simple distribution constraints and similarity learning can greatly alleviate this problem, and will not do damage to the sample-efficiency or asymptotic performance.

F.6 OTHER HYPER PARAMETERS

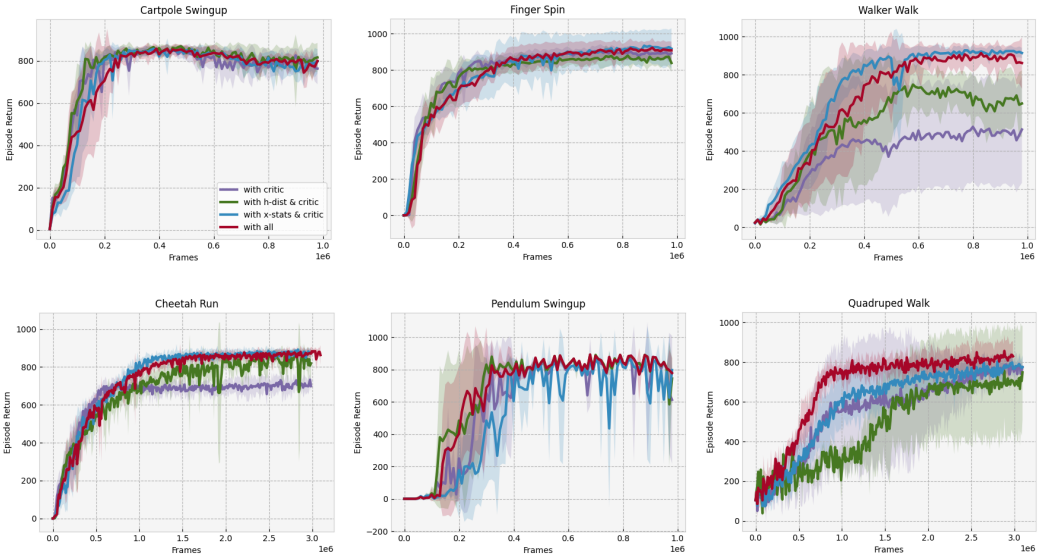

Figure 3: Ablation studies in 5 tasks for shift loss analysis. The version of **with critic** has the ability to solve most of the tasks and improve the data-efficiency, it may sink into suboptimal cases. Both of the trems in the shift loss have the ability to alleviate this problem, and if we use the overall constraints (i.e. **with all**), the performance can lead ahead of all tasks with a better stability.

Table 3: An overview of used hyper-parameters in the DeepMind control suite experiments.

Hyperparameter	**Setting**
Image size	$(84, 84)$
Replay buffer capacity	10^6
Action repeat	2
	Hopper Hop: 4
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	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	$\text{linear}(1.0, 0.1, 100000)$ for 1M frames
	$\text{linear}(1.0, 0.1, 500000)$ for 3M frames