Isometric Regularization for high-level actions on Dynamic-Aware Embeddings

Dohyeok Lee[1], **Taehyun Cho**[1]
[1] Seoul National University
{dohyeoklee, talium}@cml.snu.ac.kr

Abstract

Self-supervised representation learning in reinforcement learning embeds the states or action sequences and improves the sample efficiency. Having a good representation helps the agent generalize from a small number of samples as the model could estimate the value of its neighborhood well. However, those embedding schemes do not consider the distortion between the manifold of raw data and the underlying structure of state and action space. In this paper, we propose *isometric regularization for high-level actions* which learns the latent space that preserves the geometry of action space. In particular, we leverage Riemannian geometry by applying several isometric regularization to the decoder or encoder model. Our method shows a significant improvement upon recent embedding models on Mujoco continuous control tasks from pixel inputs.

1 Introduction

Reinforcement Learning (RL) is to learn optimal control from interacting with an unknown environment that has pre-defined rewards. Due to the curse-of-dimensionality arising from complex high-dimensional state-action space, Deep Reinforcement Learning (DRL) has been developed with the help of neural networks as a function approximator, which has a large representational capacity. However, due to the inherent design of neural networks, large amounts of data are required and DRL has lower sample efficiency as the agent additionally learns dynamics through interaction with the environment. Therefore, in order to increase the sample efficiency, self-supervised representation learning has been actively studied to learn the better representation of state-action space.

Instead of using pixel observations that are high-dimensional, the underlying structure of inputs or dynamics can often be described as low-dimensional latent spaces, and using such good representations can increase sample efficiency, generalization, and robustness. While recent papers focused on the model representation which embeds states and actions such that nearby embeddings have similar distributions of the next states, we will investigate whether isometric embedding improves sample efficiency on complex domains.

2 Preliminaries

2.1 Notation

We consider the Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$, where $\mathcal{S}$ states the state space, $\mathcal{A}$ the action space, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ the transition kernel, and γ the discount factor. We denote the transition of $k-$step action sequences as $s_{t+k} \sim P(\cdot|s_t, a_{1:k})$.

Preprint. Under review.

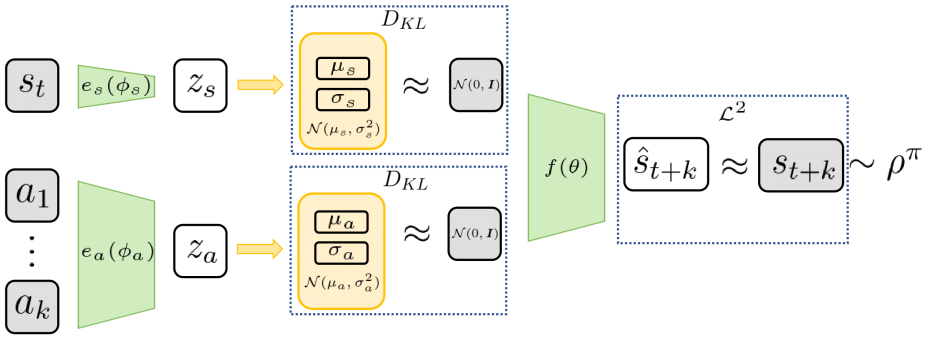

Figure 1: Illustration of DynE architecture.

For a given smooth mapping $f : \mathcal{M} \to \mathcal{N}, x \in \mathbb{R}^m \mapsto y \in \mathbb{R}^n$, a Jacobian at point x is denoted by the matrix $J(x) = \left(\frac{\partial f^i}{\partial x^j}(x)\right) \in \mathbb{R}^{n \times m}$.

2.2 Dynamics-Aware Embeddings

We refer to the expression of [4], where encoders e_s and e_a are distributions embedding a state and action sequence into latent spaces $z_s \in \mathcal{Z}_s$ and $z_a \in \mathcal{Z}_a$ respectively. We parametrize each encoder by ϕ_s and ϕ_a, and ρ^π is the marginal distribution under a behavior policy $\pi(s'|s, a_{1:k})$.

Then DynE objective is followed as:

$$\mathcal{L}(\phi_s, \phi_a, \theta) = \mathbb{E}_{s, a_{1:k}, s' \sim \rho^\pi}\Big[- \log P(s'|z_s, z_a; \theta) \tag{1}$$

$$+ \alpha D_{KL}\big(e_s(s; \phi_s) \,||\, \mathcal{N}(0, \boldsymbol{I})\big) + \beta D_{KL}\big(e_a(a_{1:k}; \phi_a) \,||\, \mathcal{N}(0, \boldsymbol{I})\big)\Big] \tag{2}$$

Inspired by the variational autoencoder structure, the first term is to predict s' through the abstracted state and high-level action and the second and third terms compress the raw state and action sequence which can be interpreted in the informational bottleneck. The low dimensionality of the latent representations prevents trivial identity mapping.

In practice, the dynamics $P(s'|z_s, z_a; \theta)$ was designed by isotropic Normal distribution with mean $f(z_s, z_a; \theta)$ where the first term is derived into $||f(z_s, z_a; \theta) - s'||_2^2$. Each encoder e_s and e_a was replaced with diagonal Normal distribution $\mathcal{N}(\mu_s, \sigma_s^2)$ and $\mathcal{N}(\mu_a, \sigma_a^2)$. The behavior policy $\pi(s'|s, a_{1:k})$ is set as uniform distribution on raw action space for every timestep.

With a fully trained encoder denoting the freeze parameter as $\overline{\phi}_a$, the decoder $d_a(\psi_a)$ is learned by

$$\mathcal{L}(\psi_a) = \mathbb{E}_{z_a \sim \mathcal{N}(0, \boldsymbol{I})}\Big[||e_a(d_a(z_a; \psi_a); \overline{\phi}_a) - z_a||_2^2 + \eta ||d_a(z_a; \psi_a)||_2^2\Big] \tag{3}$$

In contrast to the conventional autoencoder structure, the reconstruction loss is formulated by high-level action z_a. To avoid multiple outcomes, DynE gives the minimum-norm regularization which leads trajectories being smooth and energy efficient. While η was reported as 10^{-2} in the original paper, we found that actual implementation was done by 10^{-4}. After the decoder is also fully trained, the agent begins to learn the high-level policy based on the selection of learned high-level actions. DynE extended TD3 algorithms[1] and DPG[2] to work against the high-level actions. Furthermore, the additional input i is augmented to represent the length of the embedded action sequence. To train the high-level policy μ^{DynE}, the reformulated Bellman equation of augmented critic function Q^{DynE} is written as:

$$Q^{\text{DynE}}(e_s(s_t), z_{a,t}, i) = \sum_{j=0}^{k-i-1} \gamma^j r_{t+j} + \gamma^{k-i} Q^{\text{DynE}}\Big(e_s(s_{t+k-i}), \mu^{\text{DynE}}(e_s(s_{t+k-i})), i = 0\Big) \tag{4}$$

Then, for a data collecting policy π, the gradient of return J_π with respect to the deterministic policy μ^{DynE} can be estimated by

$$\nabla_\omega J_\pi(\mu_\omega^{\text{DynE}}) \approx \mathbb{E}_{s\sim\rho^\pi}\Big[\nabla_\omega \mu_\omega^{\text{DynE}}(e_s(s)), \nabla_{z_a} Q^{\text{DynE}}(e_s(s), z, 0)\Big|_{z=\mu_\omega^{\text{DynE}}(e_s(s))}\Big] \tag{5}$$

3 Isometric Regularization for high-level actions

In our work, we investigate the effectiveness of isometric regularization on action embeddings. Concretely, the objective of the encoder or decoder was regularized by $\mathcal{L}_{isometry}$ which forces the map between raw action space and high-level action space to be isometry (i.e., preserving the distance).

First, an action encoder $e_a(\phi_a)$ is regularized such that compressing the action sequences to a high-level action becomes isometry.

$$\mathcal{L}(\phi_s, \phi_a, \theta) = \mathbb{E}_{s,a_{1:k},s'\sim\rho^\pi}\Big[-\log P(s'|z_s, z_a; \theta) \tag{6}$$

$$+ \alpha D_{KL}\big(e_s(s;\phi_s) \,||\, \mathcal{N}(0, \boldsymbol{I})\big) + \beta D_{KL}\big(e_a(a_{1:k};\phi_a) \,||\, \mathcal{N}(0, \boldsymbol{I})\big) \tag{7}$$

$$+\nu_e \mathcal{L}_{isometry}(e_a(a_{1:k};\phi_a), H, G)\Big] \tag{8}$$

Also, an action decoder $d_a(\psi_a)$ reconstructs raw action sequences from high-level action while keeping the mapping to be isometric. For a pretrained encoder $e_a(\phi_a)$, our learning objective is to minimize

$$\mathcal{L}(\psi_a) = \mathbb{E}_{z_a\sim\mathcal{N}(0,\boldsymbol{I})}\Big[||e_a(d_a(z_a;\psi_a);\overline{\phi}_a) - z_a||_2^2 + \eta||d_a(z_a;\psi_a)||_2^2 \tag{9}$$

$$+\nu_d \mathcal{L}_{isometry}(d_a(z_a;\psi_a), H, G)\Big] \tag{10}$$

In order to minimize the distortion of the manifold, we choose $\mathcal{L}_{isometry}$ among 3 different candidates for isometric regularization, *iso, iso-log,* and *iso-harmonic*

$$\mathcal{L}_{iso}(f(x), H, G) = \sum_i (\lambda^i_{J^T H(f(x))JG^{-1}} - 1)^2 \tag{11}$$

$$\mathcal{L}_{iso-log}(f(x), H, G) = \sum_i \log^2(\lambda^i_{J^T H(f(x))JG^{-1}}) \tag{12}$$

$$\mathcal{L}_{iso-harmonic}(f(x), H, G) = \text{Tr}(J^T H(f(x))JG^{-1}) \tag{13}$$

where $\lambda^i_{J^T H(f(x))JG^{-1}}$ is the $i-$th eigenvalues of $J^T H(f(x))JG^{-1}$. Each term induces $J^T H(f(x))JG^{-1}$ to be an identity matrix by forcing its eigenvalues to be all one. In our experiment, we set both Riemannian metrics H and G as identity matrix $\boldsymbol{I}$.

hyperparameter	Value
discount factor γ	0.99
batch size	100
learning rate	1e-4
KL loss gain α, β	1e-4
norm loss gain η	1e-4
iso-action encoder loss gain ν_e	1e-3
iso-action decoder loss gain ν_d	1e-4
trajectory length	4
encoding training step	50
decoding training step	10
optimizer	Adam

Table 1: Hyperparameter used in the experiments

4 Experiments

We conduct experiments on ReacherVertical-v2, ReacherTurn-v2 environment referred to [3]

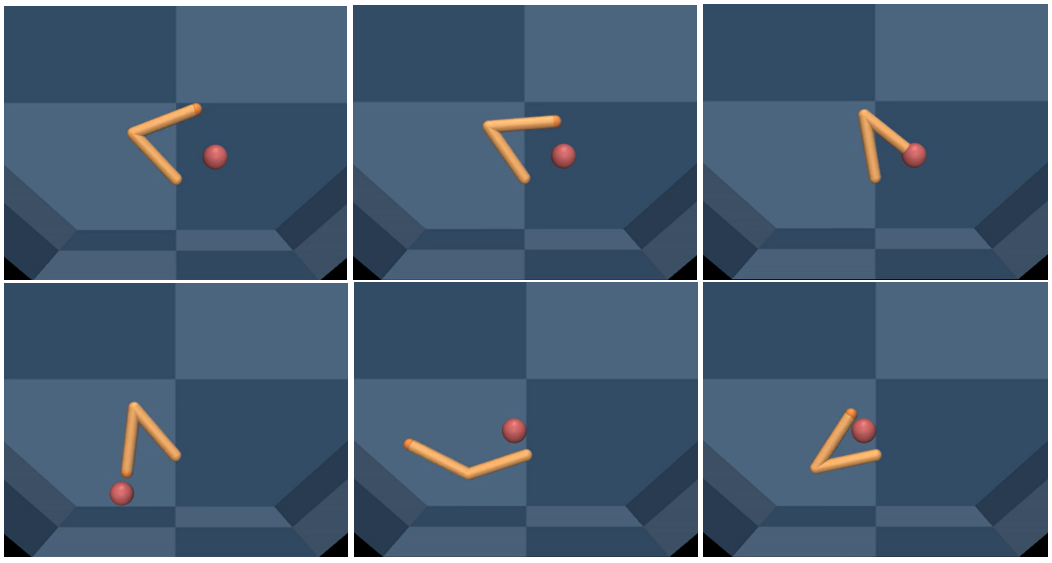

Figure 2: ReacherVertical-v2

Figure 2 shows how the ReacherVertical-v2 environment works. The goal is to make the orange-colored end effector located at the end of the arm manipulator reach the red ball. Due to the environmental simplicity, training time was 2-3 hours.

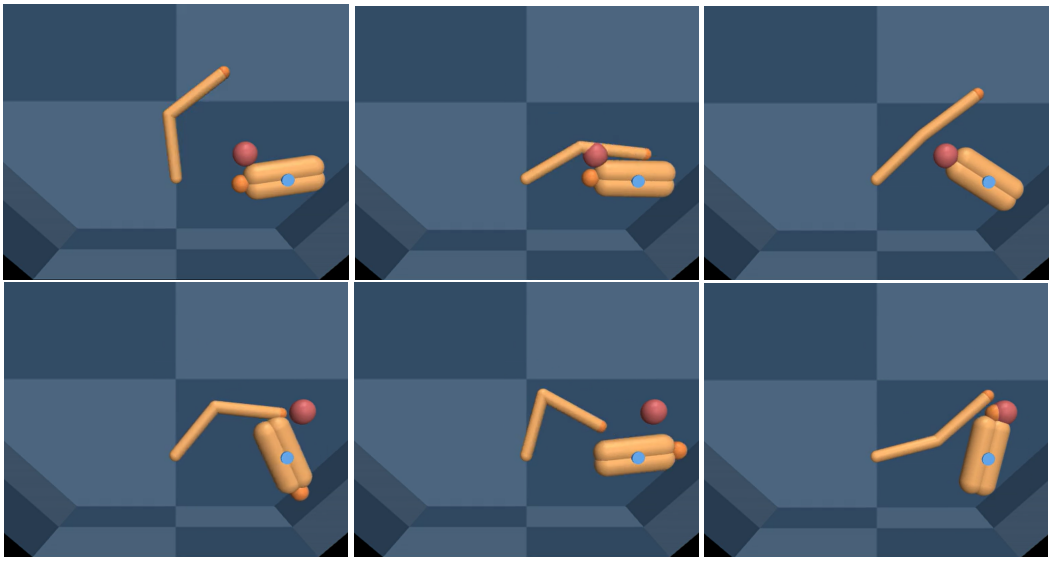

Figure 3: ReacherTurn-v2

Figure 3 shows how the ReacherTurn-v2 environment works. The goal of the task is to make the orange ball of a rigid body which is fixed by the blue axis (located at the lower right of the image) reach the red ball by touching or pushing by arm manipulator. While the dynamic is more complicated than the previous task, the training time was at least 10 hours.

We experiment on the baseline algorithm TD3 and use the three regularizers mentioned above. For each variant, we named *iso-TD3, iso-log-TD3, iso-harmonic-TD3. iso-TD3*. All of the hyperparameters we used are reported in Table 1. The performance of each line is average in 3 seeds.

4.1 Results

To show the effect of isometric regularizer for high-level actions clearly, we only use action encoder e_a and action decoder d_a and remove state encoder e_s

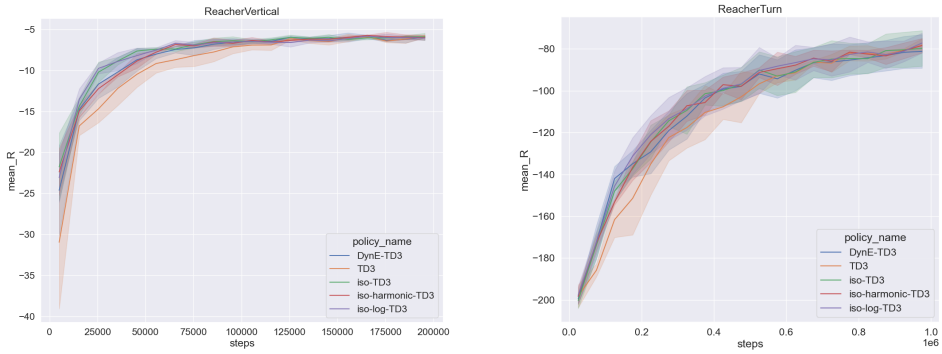

Figure 4: Performance of three different isometric variants (*iso, iso-harmonic, iso-log*) of DynE-TD3 by adding regularization on decoder. Each line is the average of 3 random seeds.

In Figure 4, *iso* and *iso-log-TD3* shows faster learning behavior than DynE-TD3 and TD3.

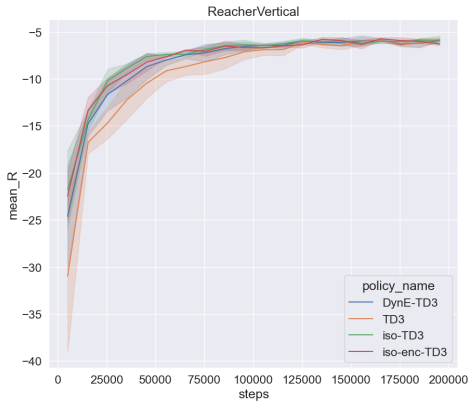

Figure 5: Performance of two different isometric variants (*iso, iso-enc*) of DynE-TD3. *iso-enc* additionally regularizes the encoder with the same term. Each line is the average of 3 random seeds.

In Figure 5, *iso* and *iso-enc* shows faster learning than DynE-TD3 and TD3. But *iso* and *iso-enc* seem no different in this experiment.

5 Conclusion

In this work, we present the isometric regularization for high-level actions while embedding the dynamic. The effectiveness of isometry regularization in action decoder shows clearly better performance in ReacherVertical, which is a less complex environment than ReacherTurn. For future work, we need to further tune the hyperparameter ν_e, ν_d for a fair evaluation of our work and investigate the effect of embedding from the more complex environment, such as hopper, halfcheetah of MuJoCo, and real robot manipulator.

References

[1] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

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

[3] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, nov 2020. doi: 10.1016/j.simpa.2020.100022. URL `https://doi.org/10.1016%2Fj.simpa.2020.100022`.

[4] William Whitney, Rajat Agarwal, Kyunghyun Cho, and Abhinav Gupta. Dynamics-aware embeddings. *arXiv preprint arXiv:1908.09357*, 2019.