PRELIMINARY PREPRINT VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference.

Wasserstein Unsupervised Reinforcement Learning

Shuncheng He, Yuhang Jiang, Hongchang Zhang, Jianzhun Shao, Xiangyang Ji

Tsinghua University
Beijing, China
hesc16@mails.tsinghua.edu.cn

Abstract

Unsupervised reinforcement learning aims to train agents to learn a handful of policies or skills in environments without external reward. These pre-trained policies can accelerate learning when endowed with external reward, and can also be used as primitive options in hierarchical reinforcement learning. Conventional approaches of unsupervised skill discovery feed a latent variable to the agent and shed its empowerment on agent's behavior by *mutual information* (MI) maximization. However, the policies learned by MI-based methods cannot sufficiently explore the state space, despite they can be successfully identified from each other. Therefore we propose a new framework Wasserstein unsupervised reinforcement learning (WURL) where we directly maximize the distance of state distributions induced by different policies. Additionally, we overcome difficulties in simultaneously training $N(N > 2)$ policies, and amortizing the overall reward to each step. Experiments show policies learned by our approach outperform MI-based methods on the metric of Wasserstein distance while keeping high discriminability. Furthermore, the agents trained by WURL can sufficiently explore the state space in mazes and MuJoCo tasks and the pre-trained policies can be applied to downstream tasks by hierarchical learning.

1 Introduction

Autonomous agents can learn to solve challenging tasks by deep reinforcement learning, including locomotive manipulation (Lillicrap et al. 2015); (Haarnoja et al. 2018) and game playing (Mnih et al. 2015); (Silver et al. 2016). The reward signal specified by the task plays an important role of supervision in reinforcement learning. However, recent research reveals the possibilities that agents can acquire diverse *skills* or *policies* in the absence of reward signal (Eysenbach et al. 2019); (Gregor, Rezende, and Wierstra 2016); (Achiam et al. 2018). This setting is called *unsupervised reinforcement learning*.

Practical applications of unsupervised reinforcement learning have been studied. The skills learned without reward can serve as primitive options for hierarchical RL in long horizon tasks (Eysenbach et al. 2019). Also the primitive options may be useful for transferring across different tasks. In model-based RL, the learned skills enables the agent to plan in the skill space (Sharma et al. 2020). Unsupervised learning methods may alleviate the cost of supervision: in certain cases, designing reward function requires human supervision (Christiano et al. 2017). The intrinsic reward derived from unsupervised learning can enhance exploration when combined with task reward (Houthooft et al. 2016); (Gupta et al. 2018).

The key point of unsupervised reinforcement learning is how to learn a set of policies that can sufficiently explore the state space. Previous methods make use of a latent variable and maximize the mutual information (MI) between the latent variable and the behavior (Eysenbach et al. 2019). Consequently the diversity in the latent space is cast into the state space. These methods are able to obtain different skills which are distinguishable from each other. However, limitations of MI-based methods are pointed out as the diversity of learned skills is restricted by the Shannon entropy of the latent variable. In addition, discriminability of skills does not always lead to the goal of sufficient exploration of the environment.

In this paper, we propose a new approach of unsupervised reinforcement learning which is essentially different form MI-based methods. The motivation of our method is to increase the discrepancy of learned policies so that the agents can explore the state space extensively and reach the state as "far" as possible compared to other policies. This idea incentivizes us to employ a *geometry-aware* metric to measure the discrepancy between the state distributions induced by different policies. In recent literature of generative modeling, the optimal transport (OT) cost is a new direction to measure distribution distance (Tolstikhin et al. 2018) since it provides a more geometry-aware topology than f-divergences, including GAN (Nowozin, Cseke, and Tomioka 2016). Therefore, we choose Wasserstein distance, a well-studied distance from optimal transport, to measure the distance between different policies in unsupervised reinforcement learning. By maximizing Wasserstein distance, the agents equipped with different policies may drive themselves to enter different areas of state space and keep as "far" as possible from each other to earn greater diversity.

Our contributions are four-fold. First, we propose a novel framework adopting Wasserstein distance as discrepancy measure for unsupervised reinforcement learning. This framework is well-designed to be compatible with various

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Wasserstein distance estimation algorithms, both in primal form and in dual form. Second, as Wasserstein distance is defined on **two** distributions, we extend our framework to multiple policy learning. Third, to address the problem of sparse reward provoked by Wasserstein distance estimation, we devise an algorithm to amortize Wasserstein distance between two bunches of samples to stepwise intrinsic reward. Four, we empirically demonstrate our approach surpasses the diversity of MI-based methods and can cover the state space by incremental learning.

2 Distribution discrepancy measure

In this Section, we briefly review Wasserstein distance and its estimation methods.

2.1 Wasserstein distance and optimal transport

Measuring discrepancy or distance between two probability distributions can be seen as a transport problem (Villani 2009). Consider two distributions p, q on domains $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^m$. Let $\Gamma[p, q]$ be the set of all distributions on product space $\mathcal{X} \times \mathcal{Y}$, with their marginal distributions on $\mathcal{X}$ and $\mathcal{Y}$ being p, q respectively. Therefore, given a proper cost function $c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ for moving mass from x to y, the Wasserstein distance is defined as

$$W_c(p, q) = \inf_{\gamma \in \Gamma[p,q]} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\gamma. \tag{1}$$

The joint distribution family $\Gamma[p, q]$ essentially forms a family of bijective plans transporting probability mass from p to q. Minimizing the transport cost is an optimal transport problem. The optimization problem suffers from supercubic complexity, since the problem becomes linear programming when $\mathcal{X}$ and $\mathcal{Y}$ are finite discrete sets (Cuturi 2013); (Genevay et al. 2016). To circumvent the difficulties, a regularizer is added to the optimization objective. The smoothed Wasserstein distance is defined as

$$\tilde{W}_c(p, q) = \inf_{\gamma \in \Gamma[p,q]} \left[\int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\gamma + \beta KL(\gamma | pq) \right]. \tag{2}$$

Minimizing the cost together with the KL divergence encourages the joint distribution $\gamma(x, y)$ to move close to $p(x)q(y)$. As $\beta \to 0$, the smoothed distance converges to $W_c(p, q)$ (Pacchiano et al. 2020).

The objective function is convex if $c(x, y)$ is a proper cost function. Therefore the infimum can be calculated either by primal formulation or dual formulation. In Section 2.2 and 2.3, we introduce practical methods estimating Wasserstein distance from distribution samples.

2.2 Primal form estimation

Solving the optimal transport problem from the primal formulation is hard. However, the problem has analytical solutions when the distributions are on one-dimensional Euclidean space and cost function is l_p measure ($p \geq 0$) (Rowland et al. 2019). Inspired by 1-D Wasserstein distance estimation, we may estimate Wasserstein distance in high dimensional Euclidean spaces by projecting distributions to $\mathbb{R}$. Suppose p, q are probability distributions on $\mathbb{R}^d$. For a vector v on the unit sphere S^{d-1} in $\mathbb{R}^d$, the projected distribution $\Pi_v(p)$ is the marginal distribution along the vector by integrating p in the orthogonal space of v. Estimating Wasserstein distance on 1-D space results sliced Wasserstein distance (SWD) (Wu et al. 2019); (Kolouri, Rohde, and Hoffmann 2018):

$$SW(p, q) = \mathbb{E}_{v \sim U(S^{d-1})} \left[W(\Pi_v(p), \Pi_v(q)) \right], \tag{3}$$

where $U(S^{d-1})$ means the uniform distribution on unit sphere S^{d-1}. In practical use, the projected distribution $\Pi_v(\hat{p})$ of empirical distribution $\hat{p} = \frac{1}{N} \sum_{n=1}^{N} \delta_{x_n}$ can be written as $\Pi_v(\hat{p}) = \frac{1}{N} \sum_{n=1}^{N} \delta_{\langle x_n, v \rangle}$, where $\langle \cdot, \cdot \rangle$ denotes inner product and δ is Dirac distribution.

To reduce estimation bias of SWD, Rowland et al. (2019) proposed projected Wasserstein distance (PWD) by disentangling coupling calculation and cost calculation. PWD obtains optimal coupling by projecting samples to $\mathbb{R}$, and calculates costs in original space $\mathbb{R}^d$ rather than the projected space.

2.3 Dual form estimation

Define set $\mathcal{A} = \{(u, v) | \forall (x, y) \in \mathcal{X} \times \mathcal{Y} : u(x) - v(y) \leq c(x, y)\}$. By Fenchel-Rockafellar duality, the dual form of Wasserstein distance is (Villani 2009)

$$W_c(p, q) = \sup_{(\mu, \nu) \in \mathcal{A}} \mathbb{E}_{x \sim p(x), y \sim q(y)} \left[\mu(x) - \nu(y) \right], \tag{4}$$

where $\mu : \mathcal{X} \to \mathbb{R}$ and $\nu : \mathcal{Y} \to \mathbb{R}$ are continuous functions on their domains. The dual formulation provides us a neural approach to estimate Wasserstein distance, circumventing the difficulties to find the optimal transport plan between two probability distributions. The dual form of smoothed Wasserstein distance shows more convenience since there are no constraints on μ, ν:

$$\begin{aligned} \tilde{W}_c(p, q) = \sup_{\mu, \nu} & \mathbb{E}_{x \sim p(x), y \sim q(y)} \left[\mu(x) - \nu(y) \right. \\ & \left. - \beta \exp \left(\frac{\mu(x) - \nu(y) - c(x, y)}{\beta} \right) \right]. \end{aligned} \tag{5}$$

Alternative dual formulation emerges when $\mathcal{X} = \mathcal{Y}$. It is the most common case that two distributions are defined in the same space. Under this assumption, Kantorovich-Rubinstein duality gives another dual form objective in which only one function with Lipschitz constraint is optimized (Villani 2009).

$$W_c(p, q) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim p(x), y \sim q(y)} \left[f(x) - f(y) \right]. \tag{6}$$

The maxima of dual problem theoretically equals to the minima of primal problem. However, sliced Wasserstein distance and projected Wasserstein distance have no such guarantee. Nonetheless, the primal form estimation methods show competitive accuracy empirically.

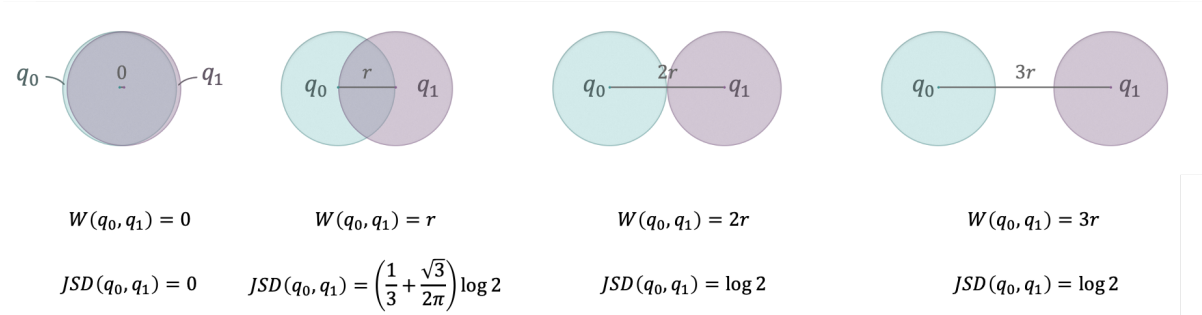

Figure 1: Examples of Jensen-Shannon divergence and Wasserstein distance between q_0 and q_1. q_0 and q_1 are uniform distributions over a round plate with radius r. The distance between the centers of q_0, q_1 varies from 0 to $3r$.

3 Wasserstein unsupervised reinforcement learning

3.1 MI-based unsupervised reinforcement learning

Traditional unsupervised reinforcement learning adopts mutual information to seek diverse skills. Mutual information between two random variables is popularly perceived as the degree of empowerment (Gregor, Rezende, and Wierstra 2016); (Kwon 2021): $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. For instance, DIAYN (Eysenbach et al. 2019) mainly aims to maximize $I(S;Z)$, the mutual information between latent variables and states reached by agent. Conventionally, the prior of latent variable $p(z)$ is fixed to a uniform distribution which has maximal entropy. The maximization process of $I(S;Z)$ broadcasts the diversity in Z to states S through policy $\pi(a|s,z)$. However, estimating mutual information involves intractable posterior distribution $p(z|s)$. With the tool of variational inference, a feasible lower bound is ready by approximating $p(z|s)$ with $q_\phi(z|s)$. We call $q_\phi(z|s)$ a learned discriminator trying to recognize the latent variable behind the policy from behavior. For example, the discriminator is a neural net predicting labels as in classification tasks when $p(z)$ is a categorical distribution.

From the view of optimization, the agent and the discriminator are trained in the cooperative way to maximize the same objective. This learning process comes to the end immediately after the discriminator can successfully infer z behind the policy. However, the learned policy is not naturally diverse enough. We will explain this claim by a simple example. Suppose the latent variable Z is randomly selected from $\{0, 1\}$. The mutual information $I(S;Z)$ equals to the Jensen-Shannon divergence between the two conditional probability distributions $q_0 = p(s|Z = 0)$ and $q_1 = p(s|Z = 1)$. As illustrated in Fig. 1, the JS divergence reaches the maximal value when the supports of q_0, q_1 do not overlap. Actually, the decomposition of mutual information $I(S;Z) = H(Z) - H(Z|S)$ also implies that $I(S;Z)$ is upper bounded by $H(Z)$, which is fixed by a predetermined distribution.

To address this issue, we propose our method that uses Wasserstein distance as intrinsic reward to encourage the agent to explore for farther states. In Fig. 1, Wasserstein distance provides information about how far the two distributions are, while the JS divergence fails. Therefore, our method will drive the agent to reach different areas as far as possible in the unknown space of valid states.

Algorithm 1: Naive WURL (test function)

Initialize two policy π_{θ_1}, π_{θ_2}, and replay buffers for each policy $\mathcal{D}_1 = \{\}$, $\mathcal{D}_2 = \{\}$. Intialize test functions.
while Maximum number of episode is not reached **do**
  Select policy l randomly or in turn. `done` = `False`.
  while Not `done` **do**
    Sample action from π_{θ_l}, execute, and receive s' and `done`.
    if $l = 1$ **then**
      Set reward $r = f(s)$ or $r = \mu(s)$.
    else
      Set reward $r = -f(s)$ or $r = -\nu(s)$.
    end if
    $\mathcal{D}_l$=$\mathcal{D}_l \cup \{(s, a, s', r)\}$.
    Train π_{θ_l} with SAC.
    Train test functions by sampling $\mathcal{D}_1$, $\mathcal{D}_2$.
  end while
end while

3.2 Wasserstein distance as intrinsic reward

The Wasserstein distance can only measure discrepancy between two distributions. Therefore for the most naive approach to Wasserstein unsupervised reinforcement learning (WURL), we train a policy pair parameterized by $\pi_{\theta_1}, \pi_{\theta_2}$, with Wasserstein distance between the state distributions $p_{\pi_{\theta_1}}(s), p_{\pi_{\theta_2}}(s)$ as their intrinsic reward.

As Pacchiano et al. (2020) mentioned, dual form estimation allows us to assign reward at every step using test functions. We adopt two manners of dual formulation, TF1 (Arjovsky, Chintala, and Bottou 2017) and TF2 (Abdullah, Pacchiano, and Draief 2018). TF1 has one test function f with Lipschitz constraint optimizing the objective in Eqn. 6. Meanwhile TF2 has two test functions μ, ν without any constraint. μ, ν are trained according to Eqn. 5. As long as the test functions are optimal dual functions, the test functions give scores of each state. By splitting the maximization objective in Eqn. 6, we can assign $f(x)$ as reward for policy 1,

and assign $-f(y)$ as reward for policy 2 to push $W_c(p, q)$ higher. Similar treatment can be applied to TF2. Combining RL training and test function training, we obtain Alg. 1.

However, primal form estimation can only be executed after policy rollout by collecting states in one episode and compute the distance with states sampled from the replay buffer from another policy. We refer this pattern of reward granting to Alg. 2. The challenge of sparse reward emerges in this training manner since the agent receives no reward until the episode ends. We will address this issue in Section 3.4.

The backend training algorithm of RL can vary. Off-policy algorithms like Soft-Actor-Critic (SAC) (Haarnoja et al. 2018) and on-policy algorithms like TRPO (Schulman et al. 2015), PPO (Schulman et al. 2017) can all be deployed on WURL. Since SAC enjoys higher sample efficiency and suits environments with continuous action space, we choose SAC as our backend RL algorithm in our experiments.

Algorithm 2: Naive WURL (final reward)

Initialize two policy π_{θ_1}, π_{θ_2}, and replay buffers for each policy $\mathcal{D}_1 = \{\}$, $\mathcal{D}_2 = \{\}$.
while Maximum number of episode is not reached **do**
  Select policy l randomly or in turn. Set trajectory $\mathcal{S} = \{\}$. `done` = `False`.
  while Not `done` **do**
    Sample action from π_{θ_l}, execute, and receive s' and `done`.
    $\mathcal{S} = \mathcal{S} \cup \{s'\}$. Set reward $r = 0$.
    if `done` **then**
      Sample target batch of states $\mathcal{T}$ from $\mathcal{D}_{3-l}$.
      Set reward $r = W(\mathcal{S}, \mathcal{T})$ by any Wasserstein distance estimation method.
    end if
    $\mathcal{D}_l$=$\mathcal{D}_l \cup \{(s, a, s', r)\}$.
    Train π_{θ_l} with SAC.
    Train test functions if WDE algorithm is one of the dual form methods.
  end while
end while

3.3 Learning with multiple policies

Learning $N (N > 2)$ policies at the same time requires the arbitrary policy i to keep distance with all other policies. We expect the policy to maximize the average Wasserstein distance between state distribution p_i induced by policy i, and other state distributions $\frac{1}{N-1}\sum_{j=1,j\neq i}^{N} W(p_i, p_j)$. Also we can use the Wasserstein distance between p_i and the average distribution of all others'. However this incurs underestimation due to the inequality

$$W\left(p_i, \frac{1}{N-1}\sum_{j=1,j\neq i}^{N} p_j\right) \leq \frac{1}{N-1}\sum_{j=1,j\neq i}^{N} W(p_i, p_j). \tag{7}$$

Practically we use $\min_{j=1,j\neq i}^{N} W(p_i, p_j)$ as reward to keep the current policy away from the nearest policy. As the number of policies growing, the number of times of distance computing grows as $\mathcal{O}(N^2)$. The dual form requires $\mathcal{O}(N^2)$ test functions which provokes memory and time consumption concern since every test function is a neural network and needs training. The primal form reward computation complexity also rises to $\mathcal{O}(N^2)$. Fortunately, sliced Wasserstein distance or projected Wasserstein distance gives us a faster and more lightweight solution without training and inferring through neural networks.

3.4 Amortized reward

In contrast to test functions, the primal form estimation of Wasserstein distance produces one final reward at the end of an episode since we cannot estimate distribution distance from one sample. This nature of primal form estimation incurs sparse reward which imposes challenges on reinforcement learning and may impair the performance of value-based RL algorithms (Andrychowicz et al. 2017).

Noting that the primal form estimation automatically yields an optimal matching plan, we could decompose the overall Wasserstein distance into every sample. Formally speaking, suppose batch $\mathcal{S} = \{x_n\}_{n=1}^{N}$ is the set of states in one episode, and batch $\mathcal{T} = \{y_m\}_{m=1}^{M}$ is the state sample set of target distribution. We further suppose $P_{N\times M}$ is the optimal matching matrix given cost matrix $C_{N\times M}$. Denote P_i as the ith row of matrix P, then the sample x_i has its own credit by computing $P_i C^\top \mathbf{1}$ where $\mathbf{1}$ is $N \times 1$ vector filled with ones. Combining with projected Wasserstein distance, we have the following algorithm for crediting amortized reward. The matching matrix computation algorithm is stated in Appendix.

Algorithm 3: Amortized reward crediting

Given source batch $\mathcal{S} = \{x_n\}_{n=1}^{N}$ and target batch $\mathcal{T} = \{y_m\}_{m=1}^{M}$.
Compute cost matrix $C_{N\times M}$
for k from 1 to K **do**
  Sample v_k from $U(S^{d-1})$
  Compute projected samples $\hat{x}_n^{(k)} = \langle x_n, v_k\rangle$, $\hat{y}_m^{(k)} = \langle y_m, v_k\rangle$
  Compute matching matrix $P_{N\times M}^{(k)}$ from projected samples
  Compute reward vector $r^{(k)} = P^{(k)} C^\top \mathbf{1}$
end for
Return mean reward vector $r = \frac{1}{K}\sum_{k=1}^{K} r^{(k)}$

3.5 Training schedule

Our proposed method can either train N policies at the same time or train incrementally by introducing new policies. Provided N diverse policies, a new policy is trained to maximize the average Wasserstein distance from other policies by collecting state samples of policy $1, 2, \ldots, N$ at the beginning. The incremental training schedule provides flexibility of extending the number of diverse policies, especially when we are agnostic about how many policies are suitable

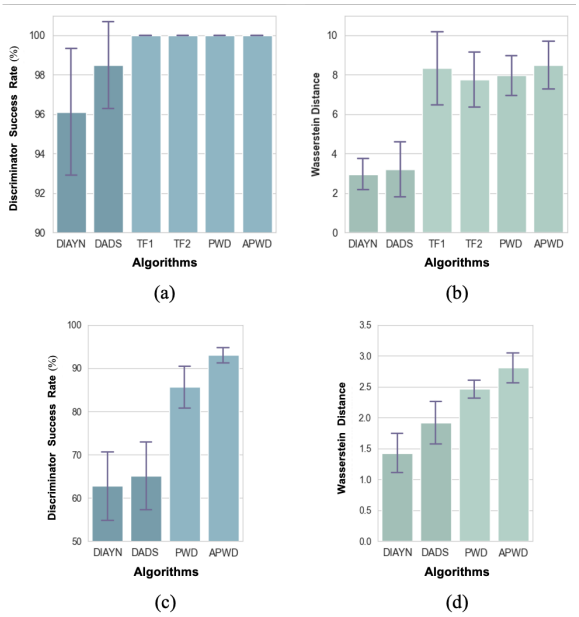

Figure 2: In the FreeRun environment, (a) and (b) show the results of learning 2 policies. (c) and (d) show the results of learning 10 policies at the same time. (a) and (c) show the success rate of the discriminator distinguishing policies. (b) and (d) show the mean Wasserstein distance between every two policies. Error bars represent standard deviations across 5 random runs.

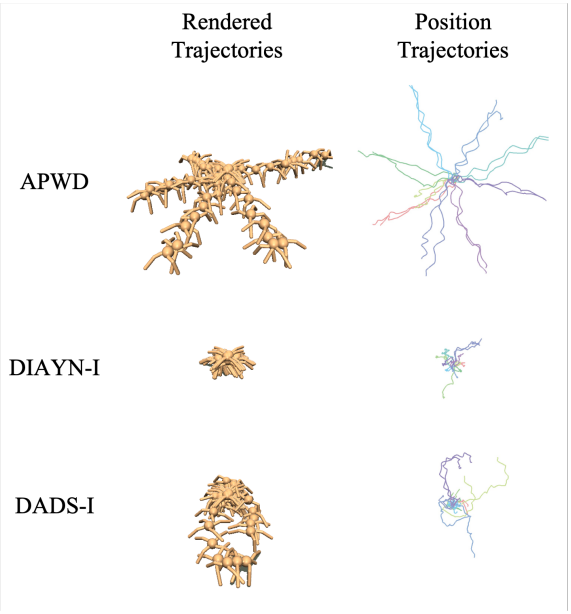

Figure 3: Visualized policies in MuJoCo Ant environment. The left column shows the rendered trajectories of 5 out of 10 total policies. The right column shows the X-Y position trajectories of 10 policies with different colors.

Algorithm	Metric	HalfCheetah	Ant	Humanoid
APWD(Ours)	DSR	0.981 ± 0.010	0.989 ± 0.002	0.998 ± 0.002
	WD	$\mathbf{39.21 \pm 0.44}$	$\mathbf{8.57 \pm 0.39}$	$\mathbf{556.96 \pm 27.78}$
DIAYN-I	DSR	0.978 ± 0.006	0.985 ± 0.001	0.999 ± 0.001
	WD	13.01 ± 2.89	4.50 ± 0.27	279.94 ± 27.98
DADS-I	DSR	0.994 ± 0.003	0.793 ± 0.035	0.999 ± 0.001
	WD	11.75 ± 1.90	7.53 ± 0.27	420.67 ± 56.83

Table 1: Policy diversity on three MuJoCo locomotion environments. Metric DSR represents the discriminator success rate and WD denotes the mean Wasserstein distance between two policies.

for a certain environment beforehand. On the contrary, mutual information based unsupervised reinforcement learning is limited by fixed number of policies that cannot be easily extended due to the fixed neural network structure.

Similar idea appears in Achiam et al. (2018), in which skills are trained by a curriculum approach. The objective of curriculum training is to ease the difficulty of classifying large number of skills. However, the maximum number of skills is still fixed in advance therefore one cannot flexibly add new policies in.

4 Experiments

4.1 Policy diversity

We first examine our methods on a FreeRun environment where the particle agent spawns at the center. The agent can only control the acceleration on the X-Y plane and take the current position and velocity as observation. The particle runs freely on the plane with a velocity limit for a fixed number of steps. We compare the performance of the two groups of algorithms:

- Mutual information as intrinsic reward: DIAYN, DADS;
- Wasserstein distance as intrinsic reward: TF1, TF2, PWD, APWD.

DADS (Sharma et al. 2020) is another mutual information based algorithm that maximizes $I(S_{t+1}; Z|S_t)$ utilizing model-based framework. It claims lower entropy of learned skills than DIAYN. TF1 and TF2 are two methods which adopt test function optimizing Eqn. 6 and Eqn. 5 respectively. PWD (Projected Wasserstein Distance) uses a primal form Wasserstein distance estimation method described in Section 2.2, and APWD is the amortized version of PWD to avoid sparse reward stated in Section 3.4. In the setting of learning 10 policies, TF1 and TF2 are unavailable since they are not compatible with multiple policies as mentioned in Section 3.2. We examine these methods from two aspects. First, we train a neural network (discriminator) to distinguish each policy from rollout state samples, and compare the success rate of the discriminator. Second, we estimate the mean Wasserstein distance between every two state distributions of two different policies. Fig. 2 shows policies learned by Wasserstein distance based algorithms generally has greater discriminability, and larger distance between the state distributions of every two policies. Fig. 2 also demonstrates amortized reward improves performance.

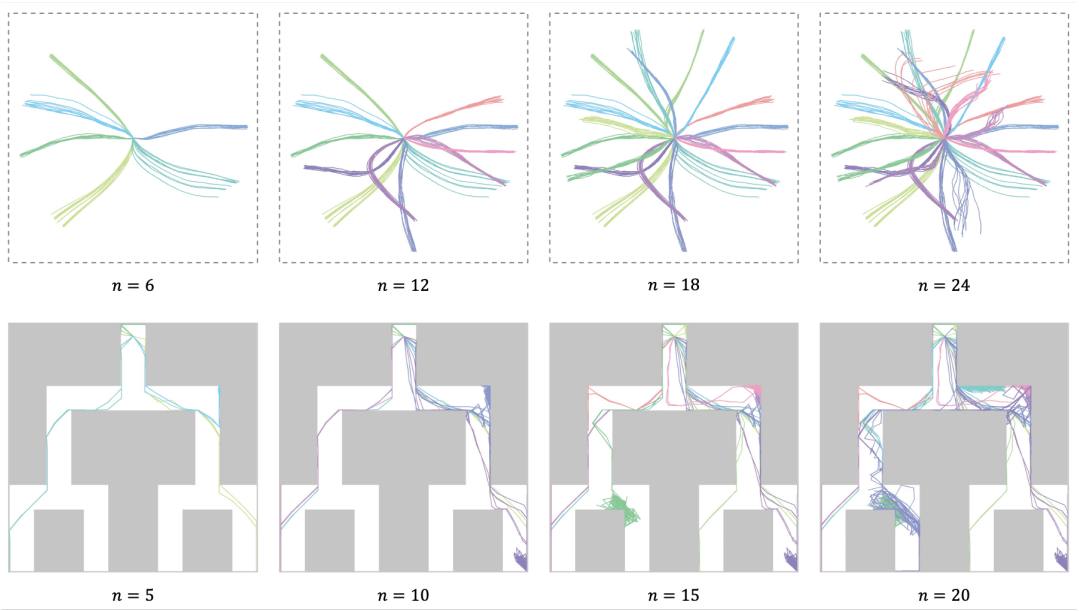

Figure 4: Results of training a bunch of policies incrementally. From left to right, new policies tend to reach new territory. As the number of policy n grows, the policies gradually fill the state space.

We also examine our algorithms on several MuJoCo tasks, including three classical locomotion environments, and two customized point mass environments where a simplified ball agent wanders in different maps with various landscapes and movable objects (see Appendix for demonstrations). Considering the significantly larger action space and state space in MuJoCo environments, we replace the shared actor network $\pi(a|s, z), z = \texttt{OneHot}(i)$ in DIAYN and DADS with N individual networks $\pi_i(a|s), i = 1, \ldots, N$, while keep the discriminator network unchanged. DIAYN and DADS with individual actors (DIAYN-I and DADS-I for short) enjoy greater parameter space and they are capable to learn a more diverse set of policies in MuJoCo tasks. Table 1 shows in the most cases, all three unsupervised RL approaches yield highly distinguishable policies. However, APWD achieves better performance on Wasserstein distance, which means the policies learned by APWD are more distantly distributed in the state space. Fig. 3 visualizes the differences of three policy sets in MuJoCo Ant environment and clearly shows that APWD encourages the policies to keep far from each other. The results verify our hypothesis in Section 3.1, that mutual information based intrinsic reward is unable to drive the policies far from each other when JS divergence saturates.

4.2 Incremental learning

Our method provides flexibility to enlarge the policy set. New policies can be trained one by one, or trained based on policies in hand. We show the process of incremental learning to illustrate how the newly learned policies gradually fill the state space in Fig. 4. As new policies added in, the particle agent in FreeRun and TreeMaze runs to new directions and new areas, and behaves differently (dithering around, turning, etc.).

Algorithm	FreeRun	Ant
APWD(Ours)	**125.56** $\pm$ **12.63**	**100.00** $\pm$ **18.03**
DIAYN-I	15.06 $\pm$ 26.97	35.00 $\pm$ 7.07
DADS-I	109.78 $\pm$ 27.08	46.00 $\pm$ 10.84

Table 2: Rewards of meta-policies on two hierarchical RL scenarios. Each meta-policy is trained with a sub-policy set which contains 10 policies pre-trained with a specific unsupervised RL algorithm listed above.

4.3 Downstream tasks

In previous unsupervised RL literature, the policies (or skills) learned without any task reward can be utilized either in hierarchical reinforcement learning or in planning (Eysenbach et al. 2019); (Sharma et al. 2020). Likewise, we examined our methods on downstream tasks including two navigation tasks based on the particle environment FreeRun and MuJoCo Ant. Both tasks require the agent to reach specific goals in a given order. The agent receives +50 reward for each goal reached. In FreeRun navigation task, we penalize each step with small negative reward to encourage the agent to finish the task as quickly as possible.

To tackle these navigation tasks with pre-trained policies, we employ a meta-policy to choose one sub-policy to execute during H steps (H is fixed in advance in our tasks). The meta-policy observes agent state every H steps , chooses an action corresponded to a sub-policy, and then receives the reward that the sub-policy collected during the successive H steps. Therefore, we can train the meta-policy with any

compatible reinforcement learning algorithms. In our experiments, we adopt PPO as meta-policy trainer (Schulman et al. 2017).

Table 2 presents the results on FreeRun and MuJoCo Ant navigation tasks. The pre-trained 10 policies with DIAYN-I, DADS-I and our proposed method APWD serve as the sub-policies in the hierarchical framework. Since our method yields more diverse and distant sub-policies, APWD based hierarchical policy achieves higher reward in both navigation tasks without doubt.

5 Related work

Learning in a reward-free environment has been attracting reinforcement learning researchers for long. Early research takes mutual information as maximization objective. They explored the topics on which variable of the policy should be controlled by the latent variable, and how to generate the distribution of the latent variable. VIC (Gregor, Rezende, and Wierstra 2016) maximizes $I(Z; S_f)$ to let the final state of a trajectory be controlled by latent code Z, while allowing to learn the prior distribution $p(z)$. VALOR (Achiam et al. 2018) takes similar approach maximizing $I(Z; \tau)$ where τ denotes the whole trajectory, but keeps $p(z)$ fixed by Gaussian distribution. (Hausman et al. 2018) uses a network to embed various tasks to the latent space. DIAYN (Eysenbach et al. 2019) improves the performance by maximizing $I(Z; S)$, fixing prior distribution, while minimizes $I(Z; A|S)$. Recent papers show their interests on better adapting unsupervised skill discovery algorithm with MDP, on the aspect of transition model and initial state. DADS (Sharma et al. 2020) gives a model based approach by maximizing $I(S'; Z|S)$ so that the learned skills can be employed in planning. Baumli et al. (2020) alternates the objective to $I(S_f; Z|S_0)$ in order to avoid state partitioning skills in case of various start states.

However, the nature of these methods restrict the diversity of learned skills since the mutual information is upper bounded by $H(Z)$. Recent research on mutual information based methods tries to enlarge the entropy of the prior distribution through fitting the uniform distribution on valid state space $U(S)$. EDL (Campos et al. 2020) takes three seperate steps by exploring the state space, encoding skills and learning skills. EDL first uses state marginal matching algorithm to yield a suffiently diverse distribution of states $p(s)$, then a VQ-VAE is deployed to encode the state space to the latent space, which creates $p(z|s)$ as the discriminator to train the agent. Skew-fit (Pong et al. 2019) adopts goal conditioned policies where the goal is sampled through importance sampling in the skewed distribution of $p(s)$ acquired by current policy. The agent can gradually extend their knowledge of state space by fitting the skewed distribution. Both methods claim they have state-covering skills.

Wasserstein distance as an alternative distribution discrepancy measure is attracting machine learning researchers recently (Ozair et al. 2019). Especially in the literature of generative models (Arjovsky, Chintala, and Bottou 2017); (Ambrogioni et al. 2018); (Patrini et al. 2020); (Tolstikhin et al. 2018), Wasserstein distance behaves well in the situations where distributions are degenerate on a sub-manifold in pixel space. In reinforcement learning, Wasserstein distance is used to characterize the differences between policies instead of commonly used f-divergences, e.g., KL divergence (Zhang et al. 2018). Pacchiano et al. (2020) reports improvements in trust region policy optimization and evolution strategies, and Dadashi et al. (2021) shows its efficacy in imitation learning by minimizing Wasserstein distance between behavioral policy and expert policy. Our proposed method inherits the motivations of using Wasserstein distance as a new distribution discrepancy measure in generative models and policy optimization. To increase the diversity of policies in unsupervised reinforcement learning, Wasserstein distance appears to be a more appropriate measure than f-divergences derived from mutual information based methods.

6 Discussion

6.1 Limitations

Our work uses Wasserstein distance as a new metric measuring discrepancy between probability distributions. Although this new metric provides better performance on unsupervised reinforcement learning in certain environments stated in Section 4, there are limitations or difficulties for further application. First, Wasserstein distance depends on cost function $c(x, y)$. In our experiments, we choose l_2-norm since the agent is conducting navigation tasks. However, choosing a proper cost function may be difficult for other reinforcement learning environments. Second, different dimensions of state space may have different semantic intention such as position, force, angle. Therefore, implementing WURL in the full dimension of state space may not properly balance between different dimensions. Third, image-based observations could not be used for calculating Wasserstein distance directly. Nevertheless, these limitations or difficulties in deploying WURL in a larger range of applications may imply future research directions.

6.2 Conclusion

We build a framework of Wasserstein unsupervised reinforcement learning (WURL) of training a set of diverse policies. In contrast to conventional methods of unsupervised skill discovery, WURL employs Wasserstein distance based intrinsic reward to enhance the distance between different policies, which has theoretical advantages to mutual information based methods (Arjovsky, Chintala, and Bottou 2017). We overcome the difficulties of extending the WURL framework for multiple policy learning. In addition, we devise a novel algorithm combining Wasserstein distance estimation and reinforcement learning, addressing reward crediting issue. Our experiments demonstrate WURL generates more diverse policies than mutual information based methods such as DIAYN and DADS, on the metric of discriminability (MI-based metric) and Wasserstein distance. Furthermore, WURL excites autonomous agents to form a set of policies to cover the state space spontaneously and provides a good sub-policy set for sequential navigation tasks.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2018AAA0102800, National Natural Science Foundation of China under Grant 61620106005.

References

Abdullah, M. A.; Pacchiano, A.; and Draief, M. 2018. Reinforcement Learning with Wasserstein Distance Regularisation, with Applications to Multipolicy Learning. *arXiv preprint arXiv:1802.03976*.

Achiam, J.; Edwards, H.; Amodei, D.; and Abbeel, P. 2018. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*.

Ambrogioni, L.; Güçlü, U.; Güçlütürk, Y.; Hinne, M.; van Gerven, M. A. J.; and Maris, E. 2018. Wasserstein Variational Inference. In *Advances in Neural Information Processing Systems*, volume 31.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Pieter Abbeel, O.; and Zaremba, W. 2017. Hindsight Experience Replay. In *Advances in Neural Information Processing Systems*, volume 30.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 214–223. PMLR.

Baumli, K.; Warde-Farley, D.; Hansen, S.; and Mnih, V. 2020. Relative Variational Intrinsic Control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35.

Campos, V.; Trott, A.; Xiong, C.; Socher, R.; Giro-i Nieto, X.; and Torres, J. 2020. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, 1317–1327. PMLR.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, volume 30.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26: 2292–2300.

Dadashi, R.; Hussenot, L.; Geist, M.; and Pietquin, O. 2021. Primal Wasserstein Imitation Learning. In *International Conference on Learning Representations*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is All You Need: Learning Skills without a Reward Function. In *International Conference on Learning Representations*.

Genevay, A.; Cuturi, M.; Peyré, G.; and Bach, F. 2016. Stochastic Optimization for Large-scale Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 29.

Gregor, K.; Rezende, D. J.; and Wierstra, D. 2016. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

Gupta, A.; Mendonca, R.; Liu, Y.; Abbeel, P.; and Levine, S. 2018. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, volume 31.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 1861–1870. PMLR.

Hausman, K.; Springenberg, J. T.; Wang, Z.; Heess, N.; and Riedmiller, M. 2018. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*.

Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2016. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, 1109–1117.

Kolouri, S.; Rohde, G. K.; and Hoffmann, H. 2018. Sliced wasserstein distance for learning gaussian mixture models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3427–3436.

Kwon, T. 2021. Variational Intrinsic Control Revisited. In *International Conference on Learning Representations*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In *Advances in Neural Information Processing Systems*, volume 29.

Ozair, S.; Lynch, C.; Bengio, Y.; Oord, A. v. d.; Levine, S.; and Sermanet, P. 2019. Wasserstein dependency measure for representation learning. In *Advances in Neural Information Processing Systems*, volume 32.

Pacchiano, A.; Parker-Holder, J.; Tang, Y.; Choromanski, K.; Choromanska, A.; and Jordan, M. 2020. Learning to score behaviors for guided policy optimization. In *International Conference on Machine Learning*, 7445–7454. PMLR.

Patrini, G.; van den Berg, R.; Forre, P.; Carioni, M.; Bhargav, S.; Welling, M.; Genewein, T.; and Nielsen, F. 2020. Sinkhorn autoencoders. In *Uncertainty in Artificial Intelligence*, 733–743. PMLR.

Pong, V. H.; Dalal, M.; Lin, S.; Nair, A.; Bahl, S.; and Levine, S. 2019. Skew-fit: State-covering self-supervised reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Rowland, M.; Hron, J.; Tang, Y.; Choromanski, K.; Sarlos, T.; and Weller, A. 2019. Orthogonal estimation of wasserstein distances. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 186–195. PMLR.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2020. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations (ICLR)*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Tolstikhin, I.; Bousquet, O.; Gelly, S.; and Schoelkopf, B. 2018. Wasserstein Auto-Encoders. In *International Conference on Learning Representations*.

Villani, C. 2009. *Optimal transport: old and new*, volume 338. Springer.

Wu, J.; Huang, Z.; Acharya, D.; Li, W.; Thoma, J.; Paudel, D. P.; and Gool, L. V. 2019. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3713–3722.

Zhang, R.; Chen, C.; Li, C.; and Carin, L. 2018. Policy Optimization as Wasserstein Gradient Flows. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5737–5746. PMLR.