Accepted by Deep Reinforcement Learning Workshop, NeurIPS 2022

TOWARDS A UNIFIED POLICY ABSTRACTION THEORY AND REPRESENTATION LEARNING APPROACH IN MARKOV DECISION PROCESSES

Min Zhang*, **Hongyao Tang***, **Jianye Hao**† and **Yan Zheng**
College of Intelligence and Computing, Tianjin University
{min_zhang,bluecontra,jianye.hao,yanzheng}@tju.edu.cn

ABSTRACT

Lying on the heart of intelligent decision-making systems, how policy is represented and optimized is a fundamental problem. The root challenge in this problem is the large scale and the high complexity of policy space, which exacerbates the difficulty of policy learning especially in real-world scenarios. Towards a desirable surrogate policy space, recently policy representation in a low-dimensional latent space has shown its potential in improving both the evaluation and optimization of policy. The key question involved in these studies is *by what criterion we should abstract the policy space* for desired compression and generalization. However, both the theory on policy abstraction and the methodology on policy representation learning are less studied in the literature. In this work, we make very first efforts to fill up the vacancy. First, we propose a unified policy abstraction theory, containing three types of policy abstraction associated to policy features at different levels. Then, we generalize them to three policy metrics that quantify the distance (i.e., similarity) of policies, for more convenient use in learning policy representation. Further, we propose a policy representation learning approach based on deep metric learning. For the empirical study, we investigate the efficacy of the proposed policy metrics and representations, in characterizing policy difference and conveying policy generalization respectively. Our experiments are conducted in both policy optimization and evaluation problems, containing trust-region policy optimization (TRPO), diversity-guided evolution strategy (DGES) and off-policy evaluation (OPE). Somewhat naturally, the experimental results indicate that there is no a universally optimal abstraction for all downstream learning problems; while the *influence-irrelevance policy abstraction* can be a generally preferred choice.

1 INTRODUCTION

How to obtain the optimal policy is the ultimate problem in many decision-making systems, such as Game Playing (Mnih et al., 2015), Robotics Manipulation (Smith et al., 2019), Medicine Discovery (Schreck et al., 2019). Policy, the central notion in the aforementioned problem, defines the agent's behavior under specific circumstances. Towards solving the problem, a lot of works carry out studies on policy with different focal points, e.g., how policy can be well represented (Ma et al., 2020; Urain et al., 2020), how to optimize policy (Schulman et al., 2017a; Ho & Ermon, 2016) and how to analyze and understand agents' behaviors (Zheng et al., 2018; Hansen & Ostermeier, 2001).

The root challenge to the studies on policy is the large scale and the high complexity of policy space, especially in real-world scenarios. As a consequence, the difficulty of policy learning is escalated severely. Intuitively and naturally, such issues can be significantly alleviated if we have an ideal surrogate policy space, which are compact in scale while keep the key features of policy space. Related to this direction, low-dimensional latent representation of policy plays an important role in Reinforcement Learning (RL) (Tang et al., 2020), Opponent Modeling (Grover et al., 2018), Fast Adaptation (Raileanu et al., 2020; Sang et al., 2022), Behavioral Characterization (Kanervisto et al., 2020) and etc. In these domains, a few preliminary attempts have been made in devising different policy representations. Most policy representations introduced in prior works resort to encapsulating the information of policy distribution under interest states (Harb et al., 2020; Pacchiano

*Equal contribution.
†Corresponding author: Jianye Hao (jianye.hao@tju.edu.cn).

et al., 2020), e.g., learning policy embedding by encoding policy's state-action pairs (or trajectories) and optimizing a *policy recovery* objective (Grover et al., 2018; Raileanu et al., 2020). Rather than policy distribution, some other works resort to the information of policy's influence on the environment, e.g., state(-action) visitation distribution induced by the policy (Kanervisto et al., 2020; Mutti et al., 2021). Recently, Tang et al. (2020) offers several methods to learn policy representation through policy contrast or recovery from both policy network parameters and interaction experiences. Put shortly, the key question of policy representation learning is *by what criterion we should abstract the policy space* for desired compression and generalization. Unfortunately, both a unified theory on policy abstraction and a systematic methodology on policy representation are currently missing.

In this paper, we make first efforts to fill up the plank in both the theory and methodology. Inspired by the state abstraction theory (Li et al., 2006), first we introduce a unified theory of policy abstraction. We start from proposing three types of policy abstraction: *distribution-irrelevance*, *influence-irrelevance*, and *value-irrelevance*. They follow different abstraction criteria, each of which concerns distinct features of policy. Concretely, we utilize the exact equivalence relations between policies and derive the corresponding policy abstractions. Further, we generalize the exact equivalence relations to policy metrics, allowing quantitatively measure the distance (i.e., similarity) between policies. Such policy metrics are more informative than the binary outcomes of policy equivalence and thus provide more usefulness in policy representation learning. Moreover, towards applying practical policy representation in downstream learning problems, we introduce a policy representation learning approach based on deep metric learning (Kaya & Bilge, 2019). We propose an *alignment loss* for a unified objective function of learning with different policy metrics. The policy representation is learned to render the abstraction criterion through minimizing the difference between the distance of policy embeddings and the quantity measure by the policy metrics. In particular, we use Maximum Mean Discrepancy (Gretton et al., 2012; Nguyen-Tang et al., 2021) for efficient empirical estimation of the policy metrics; and we adopt Layer-wise Permutation-invariant Encoder (Tang et al., 2020) for structure-aware encoding of the parameters of policy network.

In addition to the theoretical understanding of policy abstraction, we further investigate the empirical efficacy of different policy metrics and representations in characterizing policy difference and conveying policy generalization respectively. We conduct experiments in both policy optimization and policy evaluation problems. For policy optimization, we adopt Trust-Region Policy Optimization (TRPO) and Diversity-Guided Evolution Strategy (DGES) as the problem settings from (Kanervisto et al., 2020), covering both gradient-based and gradient-free policy optimization. For policy evaluation, we consider Off-policy Evaluation (OPE). In particular, we establish a series of OPE settings with different configurations of training data and generalization tasks. These settings reflect the circumstances often encountered in RL. Our experimental results indicate that, somewhat naturally, there is no a universally optimal abstraction for all downstream learning problems. Additionally, it turns out that the influence-irrelevance abstraction can be a preferred choice in general cases.

Our main contributions are summarized as follows: 1) We focus on the general policy abstraction problem and to our knowledge, we propose a unified theory of policy abstraction along with several policy metrics for the first time. 2) We propose a unified policy representation learning approach based on deep metric learning. 3) We empirically evaluate the efficacy of our proposed policy representations in multiple fundamental problems (i.e., TRPO, DGES and OPE).

2 Background

Reinforcement Learning We consider a Markov Decision Process (MDP) (Puterman, 2014) typically defined by a five-tuple $\langle S, A, P, R, \gamma \rangle$, with the state space S, the action space A, the transition probability $P : S \times A \to \Delta(S)$, the reward function $R : S \times A \to \mathbb{R}$ and the discount factor $\gamma \in [0, 1)$. $\Delta(X)$ denotes the probability distribution over X. A stationary policy $\pi : S \to \Delta(A)$ is a mapping from states to action distributions, which defines how to behave under specific states. An agent interacts with the MDP at discrete timesteps by its policy π, generating trajectories with $s_0 \sim \rho_0(\cdot)$, $a_t \sim \pi(\cdot|s_t)$, $s_{t+1} \sim P(\cdot \mid s_t, a_t)$ and $r_t = R(s_t, a_t)$, where ρ_0 is the initial state distribution. We use $P^\pi(s'|s) = \mathbb{E}_{a\sim\pi(\cdot|s)} P(s'|s, a)$ to denote the distribution of next state s' when performing policy π at state s. For a policy π, the return $G_t = \sum_{t=0}^{\infty} \gamma^t r_t$ is the random variable for the sum of discounted rewards while following π, whose distribution is denoted by Z^π. The value function of policy π defines the expected return for state s, i.e., $V^\pi(s) = \mathbb{E}_\pi[G_t \mid s_0 = s]$. The goal of an RL agent is to learn an optimal policy π^* that maximizes $J(\pi) = \mathbb{E}_{s_0\sim\rho_0(\cdot)}[V^\pi(s_0)]$.

Metric Learning Here we recall the standard definition of metrics which is central to our work.

Definition 1 (Metrics (Royden, 1968)). *Let X be a non-empty set of data elements and a **metric** is a real-valued function d: $X \times X \to [0, \infty)$ such that for all $x, y, z \in X$: 1) $d(x, y) = 0 \iff x = y$; 2) $d(x, y) = d(y, x)$; 3) $d(x, y) \leq d(x, z) + d(z, y)$. A **pseudo-metric** d is a metric with the first condition replaced by $x = y \implies d(x, y) = 0$. The combination $\langle X, d \rangle$ is called a metric space.*

A metric d is often used to quantify the *distance* between two data elements in a general sense. In this paper, we will also use *metric* to stand for *pseudo-metric* for brevity. Typically metric learning aims to reduce the distance between similar data and increase the distance between dissimilar data. With nonlinear transformation offered by deep neural networks, Deep Metric Learning allows us to find such optimal metrics by optimizing a latent representation space of raw data.

3 Policy Abstraction Theory

Inspired by the state abstraction theory (Li et al., 2006), in this section, we make the first effort in proposing a unified policy abstraction theory. First, we propose the formal definition of three types for policy abstraction; then, we generalize the abstractions to three types of policy metrics. Finally, we analyze the properties of policy abstraction and compare them in several Gridworld MDPs.

3.1 Policy Abstraction

First of all, following the classic definition of an abstraction (Giunchiglia & Walsh, 1992), we propose a general definition of policy abstraction as follows:

Definition 2 (Policy Abstraction). *A policy abstraction $f : \Pi \to \mathcal{X}$, is a mapping from ground policy space Π to an abstract space $\mathcal{X}$. $f(\pi) \in \mathcal{X}$ is the abstract policy representation corresponding to ground policy $\pi \in \Pi$, and the inverse image $f^{-1}(\chi)$ with $\chi \in \mathcal{X}$, is the set of ground policies that correspond to χ under abstraction function f.*

It is apparent that there are many such abstractions since we may have many possible ways to partition the policy space. However, we are only interested in some useful ones among them that follow specific abstraction criteria to preserve the important features related to decision making. In this paper, we present three types of policy abstraction which are defined below:

Definition 3. *Given an MDP and a ground policy space Π, for any two policies $\pi_i, \pi_j \in \Pi$, we define three types of policy abstraction as follows:*

1. *A distribution-irrelevance abstraction (f_π) is such that for all $s \in S$, $a \in A$, $f_\pi(\pi_i) = f_\pi(\pi_j)$ implies that $\pi_i(a \mid s) = \pi_j(a \mid s)$.*
2. *An influence-irrelevance abstraction (f_{P^π}) is such that for all $s, s' \in S$, $f_{P^\pi}(\pi_i) = f_{P^\pi}(\pi_j)$ implies that $P^{\pi_i}(s'|s) = P^{\pi_j}(s'|s)$.*
3. *A value-irrelevance abstraction (f_{V^π}) is such that for all $s \in S$, $f_{V^\pi}(\pi_i) = f_{V^\pi}(\pi_j)$ implies that $V^{\pi_i}(s) = V^{\pi_j}(s)$.*

These abstractions aggregate policies based on the corresponding equivalence relations with respective concerns on different features of policy. Intuitively, the distribution-irrelevance abstraction (f_π) preserves the action distribution of the policy; the influence-irrelevance abstraction (f_{P^π}) preserves the state transition distribution induced by the policy, i.e., the influence caused by the policy on the environment; and value-irrelevance abstraction (f_{V^π}) preserves the value function of the policy. In addition to the policy abstractions introduced in Definition 3, we provide some other ones in Appendix A.2. Moreover, we revisit the policy abstractions adopted in prior related works and summarize them from the angle of our policy abstraction theory in Table 4 of Appendix B.

3.2 Policy Metrics

The policy abstractions allow us to aggregate policies according to equivalence relation. However, exact equivalence is rarely encountered in continuous policy space (e.g., the usual case with neural policies), thus useful abstraction can be seldom obtained. Moreover, the equivalence relation offers only qualitative (i.e., binary) outcomes and is incapable of measuring the similarity between policies, which is significant to policy representation learning. To this end, we generalize the policy abstractions to policy metrics which quantitatively measures the distance between two policies.

Corresponding to the three types of policy abstraction, we define the following three policy metrics:

Definition 4. *Given an MDP, a ground policy space Π, a state distribution $p(s)$ and a distribution (pseudo-)metric $D(\cdot, \cdot)$, for any two policies $\pi_i, \pi_j \in \Pi$, we define three policy metrics as follows:*

Table 1: Properties of different policy abstraction.

Abstraction	Abstraction Criterion (for $\pi_1, \pi_2, \forall s, s', a \in S^2 \times A$)	Fineness	Task Relevance
f_Θ	Policy Parameter Equivalence ($\theta_1 = \theta_2$)	Highest	None
f_π	Action Distribution Equivalence ($\pi_i(a \mid s) = \pi_j(a \mid s)$)	High	Low
f_{P^π}	Dynamics Influence Equivalence ($P^{\pi_i}(s'\|s) = P^{\pi_j}(s'\|s)$)	Middle	Middle
f_{V^π}	Value Function Equivalence ($V^{\pi_i}(s) = V^{\pi_j}(s)$)	Low	High
f_0	Triviality (taking all policies as the same)	Lowest	None

1. *A distribution-irrelevance metric:* $d_\pi(\pi_i, \pi_j) = \mathbb{E}_{s \sim p(s)} [D(\pi_i(a \mid s), \pi_j(a \mid s))]$.
2. *An influence-irrelevance metric:* $d_{P^\pi}(\pi_i, \pi_j) = \mathbb{E}_{s \sim p(s)} [D(P^{\pi_i}(s' \mid s), P^{\pi_j}(s' \mid s))]$.
3. *A value-irrelevance metric:* $d_{V^\pi}(\pi_i, \pi_j) = \mathbb{E}_{s \sim p(s)} [D(Z^{\pi_i}(s), Z^{\pi_j}(s))]$.

These metrics follow the same abstraction criteria as in Definition 3, i.e., the irrelevance regarding action distribution, influence and value, measuring the similarity of policies by the distance at respective levels. Compared to the binary outcomes offered by the equivalence relations, the metrics defined here are continuous, thus are more informative in comparing and representing policies in finer views. Specially, one may see that the equivalence relations used in Definition 3 induce corresponding discrete pseudo-metrics, e.g., $d_\pi^{\text{Eq}}(\pi_i, \pi_j) = 0$ if $f_\pi(\pi_i) = f_\pi(\pi_j)$, and 1 otherwise. Notice the metrics proposed above depends on the distribution metric D and state distribution $p(s)$. For D, typical choices can be Jeffreys Divergence (Jeffreys, 1946) and Maximum Mean Discrepancy (MMD) (Nguyen-Tang et al., 2021). For $p(s)$, intuitively, it should be the distribution of states we are interested in when comparing two policies. We defer the concrete choices for practical implementation of these metrics in Section 4.

3.3 Properties of the Abstractions

Superficially, the three abstractions proposed preserve features that are progressively more relevant to decision making in the learning task, but essentially, what is the relationship between the three abstractions? To investigate the problem, we define the *fineness* of policy abstractions similar to the one for state abstractions used in (Li et al., 2006), to prove how the three abstractions are related.

Definition 5 (Abstraction Fineness)**.** *Let F_Π denotes the set of abstractions on ground policy space Π. Suppose $f_1, f_2 \in F_\Pi$. We say f_1 is finer than f_2, denoted $f_1 \succeq f_2$, iff $\forall$ π_1, $\pi_2 \in \Pi$, $f_1(\pi_1) = f_1(\pi_2)$ implies $f_2(\pi_1) = f_2(\pi_2)$. If, $f_1 \neq f_2$, then f_1 is strictly finer than f_2, denoted $f_1 \succ f_2$. In contrast, we may also say f_2 is (strictly) coarser than f_1, denoted $f_2 \preceq f_1$ ($f_2 \prec f_1$).*

It is easy to see the relation $\succeq$ satisfies self-reflexivity, antisymmetry and transitivity, thus it is a partial ordering. Consider the set of possible policy abstractions, while the coarsest abstraction (f_0) is the trivial representation where all policies are treated as the same; while the finest abstraction is the identity representation, e.g., $f_\Theta(\pi_\theta) = \theta$ for a policy neural network parameterized with $\theta \in \Theta$. With the partial ordering $\succeq$, we further derive the following theory.

Theorem 3.1. *Under the Definition 3 and 5, we have* $(f_\Theta \succeq) f_\pi \succeq f_{P^\pi} \succeq f_{V^\pi} (\succeq f_0)$.

The proof is provided in Appendix A.1. The theorem declares how the three policy abstractions are related to each other in the sense of abstraction fineness with the two extreme cases (f_Θ, f_0) for reference. The coarser the abstraction is, the more the original policy space is abstracted.

In Table 1, we summarize the properties of different policy abstractions, regarding abstraction criteria, abstraction fineness and task relevance. The major conclusion is that there is an inverse relation between abstraction fineness and task relevance. Except for the two extreme cases (f_Θ, f_0) that are totally task-independent, the policy abstraction becomes more task-relevant as the abstraction criterion concerns more policy features related to the learning task. The distribution-irrelevance abstraction f_π concerns the policy behavior in the learning task, defined by action distribution at interested states; meanwhile f_π is coarser than f_Θ since the same policy behavior can be realized by non-unique policy parameter. Taking one step closer to the task, the influence-irrelevance abstraction f_{P^π} cares about the state transition dynamics induced by policy behavior. Obviously, f_{P^π} is coarser than f_π as different behaviors may induce the same transition distribution. The value-irrelevance abstraction f_{V^π} further involves the rewards of long-term dynamics, thus is the most task-relevant and coarsest among the three types of policy abstraction.

Empirical Comparison of Policy Metrics in Gridworld MDPs To compare these policy abstractions in a quantitative view, we demonstrate how the distances of two policies measured by the

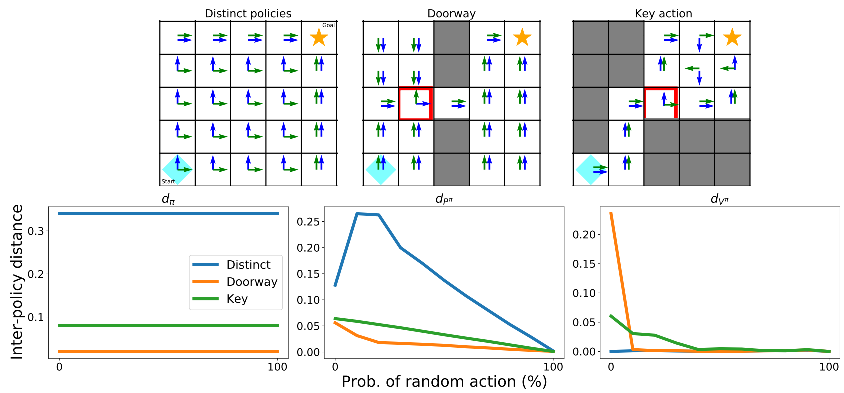


Figure 1: Policy comparison with different policy metrics in Gridworld. *Top Panel:* The illustration of three Gridworld MDPs and two deterministic policies (blue and green). *Bottom Panel:* The distance curves of the two policies measured by $d_\pi, d_{P^\pi}, d_{V^\pi}$ (y-axi), against the stochasticity of the environment (x-axi). d_{P^π} is able to distinguish the two policies across all the settings.

corresponding policy metrics differ in several Gridworld MDPs. We use *Distinct Policies*, *Doorway* from (Kanervisto et al., 2020) and design a new task, *Key Action* for simple prototypes of tasks with different features; moreover, we increase the stochasticity of the environment for a better evaluation. In particular, $\mathbb{E}_{s\sim p(s)} D(\cdot,\cdot)$ is calculated by average the absolute differences over all states. The illustrations and results are shown in Fig. 1 while more results can be found in Appendix C.

We observe that the distribution-irrelevance metric d_π may fail to show the difference in dynamics and outcome between the two policies, e.g., in Doorway. This is because d_π measures the difference in the action distribution itself, i.e., independent of the dynamics as well as the increasing stochasticity of the environment. This issue may be resolved by using a designated distribution that concentrates over key states. Conversely, the value-irrelevance d_{V^π} measures the difference in the outcomes of the two policies regardless their differences in action distribution and dynamics, e.g., in Distinct Policies. Another discovery is that d_{V^π} quickly degenerates and turns to be not informative as the increase of stochasticity, showing its poor robustness. By contrast, the influence-irrelevance metric d_{P^π} is a *sweet* intermediate point, consistently keeping the ability of distinguishing the two policies across all the environments and stochasticity configurations. In a summary, different policy abstractions and metrics may yield different outcomes for the same two policies and the optimality depends on the specific downstream learning problem concerned. Later, we evaluate these options in representative downstream RL problems in Section 5 and 6 for useful insights.

4 Policy Representation Learning Approach

The next question concerned in practice is: *how can we learn the representation of RL policies (usually modeled by NNs) in a general way?* Based on the policy metrics introduced above, we propose a policy representation learning approach by following the principle of Deep Metric Learning.

4.1 Learning Policy Representation by Embedding Alignment

The policy metrics proposed in previous section measure the quantitative relationship between policies from different perspectives of the policy abstraction criteria. For a unified objective function of learning from different policy metrics, we use the *alignment loss*, with which the difference between the distances of two policies in the representation space and in the policy metric space is minimized. Concretely, consider a policy representation function f_ψ, and the alignment loss can be formalized as,

$$\mathcal{L}_{\text{AL}}(\psi) = \mathbb{E}_{\pi,\pi'\in\Pi}\left[\left(\|f_\psi(\pi) - f_\psi(\pi')\|_2 - \eta d_*(\pi,\pi')\right)^2\right], \tag{1}$$

where we consider $d_* \in \{d_\pi, d_{P^\pi}, d_{V^\pi}\}$ and η is the weight for scaling. As we can see, $\mathcal{L}_{\text{AL}}(\psi)$ consists of two metrics, i.e., the L_2 distance function ($\|\cdot\|_2$) of two inputs and the policy metric ($d_*(\cdot,\cdot)$). Intuitively, minimizing the alignment loss is to align the two metrics by optimizing the policy representation function f_ψ. By this means, we are able to learn different policy representation functions, which maps the ground policy $\pi\in\Pi$ to the latent embedding $\chi_\pi = f_\psi(\pi) \in \mathcal{X}$. The embedding preserves the policy features corresponding to the abstraction criteria reflected by the specific policy metric considered. For a practical implementation, the following problems are the estimation of policy metrics d_* and the realization of the training for policy representation function f_ψ, which are detailed in the next two subsections respectively.

In the literature of learning policy representation, representative methods follow the principle of behavior recovering (Grover et al., 2018) and policy contrast (Tang et al., 2020). To our knowledge, none of prior works take a systematic view of policy abstraction. In Table 4, we show that prior methods are specific instances of one of our proposed policy abstractions that differ in realization.

4.2 Estimating Policy Metrics via Maximum Mean Discrepancy

Given a tractable metric D and a state distribution p, the policy metrics (i.e., d_π, d_{P^π}, d_{V^π}) can be calculated exactly if the probability distributions (i.e., π, P^π, Z^π) are available. However, this is usually infeasible in practice; instead, in more regular cases, only finite samples of policy interaction are available. Although the empirical distributions can be estimated in simple MDPs where the state-action space is finite (as in Appendix D), unfortunately, approximating the distributions and computing the metrics are non-trivial, especially with high-dimensional continuous state-action space.

Therefore, we estimate the policy metrics directly from the samples, bypassing estimating the empirical distributions (i.e., $\widetilde{\pi}$, $\widetilde{P}^\pi$, $\widetilde{Z}^\pi$). In particular, we adopt MMD (Nguyen-Tang et al., 2021; Sejdinovic et al., 2012) as the distribution metric, i.e., let D be D_{MMD}. MMD measures the maximum value of the mean discrepancy of two distributions regarding all possible functions in a predefined family. Conventionally, let the class of functions $h : X \to \mathbb{R}$ be a unit ball in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ associated with a continuous kernel $k(\cdot,\cdot)$ on X, p, q be two distribution defined on X, x, x' and y, y' be i.i.d. samples from p and q respectively, the MMD is defined as:

$$\begin{aligned} D_{\text{MMD}}(p, q; \mathcal{H}) &= \sup_{h\in\mathcal{H}:\|h\|_{\mathcal{H}}\le 1} \left(\mathbb{E}_{x\sim p}[h(x)] - \mathbb{E}_{y\sim q}[h(y)]\right) = \|\mu_p - \mu_q\|_{\mathcal{H}} \\ &= \left(\mathbb{E}_{x,x'}[k(x,x')] + \mathbb{E}_{y,y'}[k(y,y')] - 2\mathbb{E}_{x,y}[k(x,y)]\right)^{\frac{1}{2}}, \end{aligned} \tag{2}$$

where $\mu_p = \int_X k(x,\cdot)p(\mathrm{d}x)$ is the mean embedding of p into $\mathcal{H}$(Smola et al., 2007). Thus, MMD can be empirically estimated with samples $\{x_i\}_{i=1}^N \sim p$ and $\{y_i\}_{i=1}^M \sim q$:

$$\widetilde{D}^2_{\text{MMD}}(\{x_i\}, \{y_i\}; k) = \frac{1}{N^2}\sum_{i,j} k(x_i, x_j) + \frac{1}{M^2}\sum_{i,j} k(y_i, y_j) - \frac{2}{NM}\sum_{i,j} k(x_i, y_j). \tag{3}$$

According to Eq. 3, we can estimate the policy metrics $d_\pi, d_{P^\pi}, d_{V^\pi}$ empirically from the samples $\{a_i\}, \{s'_i\}, \{G_i\}$ of different policies respectively, under the sampled states $\{s_i\}$ for the expectation $\mathbb{E}_{p(s)}$. However, it is often impractical to obtain multiple samples under the same state. Thus, we resort to estimating the surrogates, e.g., $\hat{d}_{P^\pi}(\pi_i, \pi_j) = D\left(P^{\pi_i}(s, s'), P^{\pi_j}(s, s')\right)$ for d_{P^π}, where the joint distributions rather than the state-conditioned distributions are measured. We use Gaussian kernel by default, i.e., $k(x, x') = \exp\left(-\frac{\|x-x'\|_2^2}{2\sigma^2}\right)$. Consequently, the empirical estimates of the policy metrics serve as the self-supervision in Eq. 1.

4.3 Realizing the Training of Policy Representation Function

With the empirical policy metrics provided in previous section, the training of policy representation is straightforward with a differentiable function f_ψ by optimizing the alignment loss (Eq. 1). The realization of policy representation function concerns two aspects: 1) the choice of policy data (or original representation) and 2) the construction of f_ψ (i.e., how policy data is encoded).

For the first aspect, we focus on parameterized policy π_θ (typically by a neural network) and use policy parameter θ as the policy data. One may recall that θ itself can be viewed as the finest representation obtained by policy abstraction f_Θ in Table 1. Such an original representation (i.e., θ) is high-dimensional and highly nonlinear, offering no help in the compression and generalization of policy space. In addition, we are aware that in some cases the policy parameters may be not available, and thus the interaction experiences generated by the policy can be alternative policy data, as used in (Grover et al., 2018; Tang et al., 2020). Our policy representation learning approach is compatible with such alternatives with the need of possible slight modifications. For the second aspect, we adopt Layer-wise Permutation-invariant Encoder (LPE) (Tang et al., 2020) as the implementation choice of f_ψ, which has demonstrated the effectiveness in encoding conventional policy networks. To be specific, for the parameter $\theta = \{W_i, b_i\}_{i=0}^k$ of policy π, i.e., the weights and biases of k-layer MLP,[1]

[1]The activation function is not considered since the structure is fixed for policies in convention RL setting. In principle, LPE can be generalized to tailor other advanced network structure.

the weight $W_i \in \mathbb{R}^{l_i \times l_{i+1}}$ and bias $b_i \in \mathbb{R}^{1 \times l_{i+1}}$ (l_i is the unit number of the i-layer; l_0 and l_k are for the input and output layers) are concatenated ($\oplus$) and transposed, followed by a MLP ($f_{\psi,i}$) and a mean-reduce operation (MR), resulting in a layer embedding z_i; Thereafter, the policy embedding is obtained by concatenating the embedding of each layer. Formally,

$$z_i = \texttt{MR}\left(f_{\psi,i}([W_i \oplus b_i]^\top)\right) = \frac{1}{l_i+1}\sum_{j=1}^{l_i+1} f_{\psi,i}\left(([W_i \oplus b_i]^\top)_{j,\cdot}\right), \quad \chi_{\pi_\theta} = f_\psi(\theta) = \bigoplus_{i=0}^{k} z_i \tag{4}$$

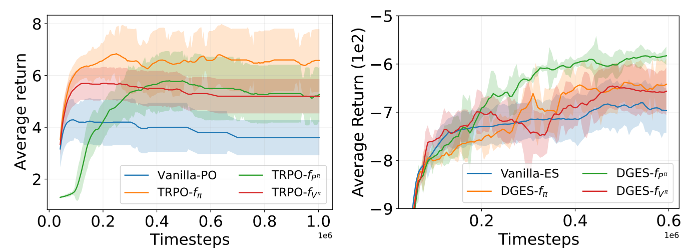
Each row of $[W_i \oplus b_i]^\top$, indexing by the subscript $j, \cdot$, describes a transformation of the i-layer into the next layer. All the rows are fed into $f_{\psi,i}$ separately and are then averaged into z_i. In a consequence, the policy embedding serves as the compact representation of the policy network by summarizing the transformations made by the each layer of it. The significant difference between LPE and a straightforward MLP encoder is that, LPE provides structure-aware representation, i.e., both the intra-layer and inter-layer structures are explicitly considered. Intuitively, this alleviates the difficulty of learning representation from the policy network parameters. Other advanced encoder structures are beyond the scope of this work and we leave them as future work.

Till now, we can update the parameters of LPE $\psi = \{\psi_i\}_{i=0}^k$ by optimizing Eq. 1 with the policy samples from some given dataset of policies and the empirical policy metrics estimated accordingly. Depending on the specific choice of policy metric, the policy representation is learned to render the policy abstraction in Table 1, starting from f_Θ and going downwards to the corresponding level.

5 Applying Policy Abstraction to Policy Optimization

Despite the theoretical understanding of the policy abstraction, we have no idea about how the derived policy metrics behave in different downstream learning problems. To shed some light on this, we evaluate the efficacy of the policy metrics proposed in Sec. 3.2 in policy optimization, including Trust-Region Policy Optimization (TRPO) and Diversity-Guided Evolutionary Strategy (DGES).

Trust-Region Policy Optimization We adopt TRPO problem as the first test stone for our policy abstractions. Specifically, the objective of TRPO problem is to maximize the policy return while constraining the difference between old and new policies: $J_{\text{TRPO}}(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}}[\mathcal{R}(\tau)]$, $s.t.$, $d_*(\pi_\theta, \pi_{\theta_{old}}) \leq \sigma$, where σ is a threshold. For our experiments, we consider the policy metrics $d_* \in \{d_\pi, d_{P^\pi}, d_{V^\pi}\}$. In another word, the learning agent checks if the difference measured by the policy metrics are larger than σ for each policy update. In this experiment, the original TRPO (Schulman et al., 2015) is generalized to incorporate different alternative metrics for the trust-region constraint. Thus, we can evaluate the efficacy of the different trust regions provided by our proposed policy metrics, shedding some light on what policy features we care the most in TRPO.



(a) TRPO (b) DGES

Figure 2: Performance of different policy abstractions in: *(a)* Trust-Region Policy Optimization (TRPO); and *(b)* Diversity-Guided Evolution Strategy (DGES). Results are the mean and half a std (shaded) over 10 and 5 trials for TRPO and DGES respectively.

We adopt a Gridworld environment where the agent can move to one of N directions at each grid and only one direction yields high reward (Kanervisto et al., 2020). The results are shown in Fig. 2(a). We observe that all our TRPO variants (i.e., TRPO-f_*) outperform Vanilla-PO (i.e., no trust-region constraint used), demonstrating the effectiveness of our policy abstractions. Moreover, TRPO-f_π outperforms the others. This is because f_π follows the abstraction criterion regarding action distribution, thus pertains to the essence of TRPO. By contrast, f_{P^π} and f_{V^π} utilize coarser abstraction which does not hold the features of action distribution. In addition, we demonstrate the superiority of our policy abstractions when compared with existing related methods in Appendix E.1.

Diversity-Guided Evolution Strategy Next, we adopt DGES problem as the second test stone for our policy abstractions. Formally, the objective of DGES problem is to maximize the policy return of the current policy π_θ and maximize its policy difference to the ancestor policy $\bar{\pi}$: $J_{\text{DGES}}(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_{\pi_\theta}}[\mathcal{R}(\tau)] + \beta \sum_{p=1}^{N} d_*(\pi_\theta, \bar{\pi})$, where $\beta \geq 0$ is the weight. Similarly, we consider the policy metrics $d_* \in \{d_\pi, d_{P^\pi}, d_{V^\pi}\}$. Here, the choices of policy metrics realize the population diversity in different ways. We aim at exploring the diversity concerning which policy feature is the most effective in DGES.

To explore this, we leverage the Point environment with deceptive rewards (Pacchiano et al., 2020). The results are reported in Fig. 2(b). In comparison to Vanilla-ES (i.e., $\beta = 0$), optimizing policy diversity (i.e., $\beta > 0$) based on our policy metrics (i.e., DGES-f_*) does help exploration and thus leads to better performance. In particular, DGES-f_{P^π} performs the best. Since ES optimizes policy in a gradient-free fashion, the evolution process concerns only policy return. Therefore, the distribution-irrelevance abstraction f_π (i.e., the winner in the TRPO experiment) can be redundant since multiple action distributions may have the same outcome (i.e., influence and value). For the value-irrelevance abstraction f_{V^π}, it turns to be too fine to contain the features of policy behavior (i.e., action distribution and influence). Therefore, the influence-irrelevance abstraction f_{P^π} serves as a sweet point. Furthermore, we provide additional comparative evaluation in Appendix E.2.

6 Applying Policy Abstraction to Off-policy Evaluation

After the investigation in policy optimization, now we move to policy evaluation. Typical OPE (Fu et al., 2021; Harb et al., 2020) focuses on using offline data to evaluate unseen policies. Likewise, we are interested in studying the value generalization performance on unseen policies of the representations learned regarding different policy abstractions. The appealing characteristic of policy representation in value generalization has been studied in (Tang et al., 2020), where a Policy-extended Value Function Approximator (PeVFA,$\mathbb{V}(\chi_\pi)$) takes as input the policy representation χ_π approximates the values of multiple policies and offers implicit value generalization among the policy representation space.

For policy data collection, we run PPO (Schulman et al., 2017b) in OpenAI Gym continuous control tasks: InvertedDoublePendulum-v2 (IDP-v2) and LunarLanderContinuous-v2 (LLC-v2) (Brockman et al., 2016). By collecting the policies at intervals during the learning process, we build an offline policy set, based on which we train our policy representations and a PeVFA $\mathbb{V}(\chi_\pi)$. For concrete problem settings, we establish both weak and strong generalization OPE scenarios which differs at the difficulty of evaluating the unseen policies. For the weak generalization scenario (*easy*), we sample training data uniformly from the whole band of the policy set. For the strong generalization scenario (*hard*), we separate the policy set and use the low-performance policies for the training data, with the rest taken as the unseen policies to evaluate. For both the settings, the ratio of sampling and separation is set to be 20%, 40%, or 80%. We report the results of the ratio 20% in Table 2 and leave the results of other ratios in Appendix G. For evaluation protocols, we report the evaluation (testing) error of unseen policies (**T-error**) and the generalization gap (**G-gap**), i.e., the difference between training and testing error. We denote different policy representations by their underlying policy abstraction (e.g., f_π) correspondingly. Complete details can be found in Appendix F.

Weak Generalization Scenario in OPE First, we study the empirical comparison in the weak generalization scenario. Table 2 reports the results of value generalization for the policy representations learned based on corresponding policy abstractions. To be specific, the f_Θ denotes directly using policy parameters θ as policy representations (i.e., no representation training). For our proposed policy abstractions $f_\pi, f_{P^\pi}, f_{V^\pi}$, we learn the representations for them according to Eq. 1 based on the LPE and MMD estimation (Sec. 4.3). To further complete the comparison, we include two additional representations f_{RE} and f_{EL}: f_{RE} uses a randomly initialized LPE with no further training while f_{EL} uses the LPE trained by the end-to-end OPE loss (see Appendix F.2) respectively. Note that f_{EL} can be viewed as a variant of f_{V^π} since it also learns from values but does not optimize the alignment loss. Besides, we also include the representation (f_{CL}) learned by unsupervised contrastive learning based on InfoNCE loss (van den Oord et al., 2018), as proposed in (Tang et al., 2020).

From the Table 2 (Weak Generalization), we can observe that $f_\pi, f_{P^\pi}, f_{V^\pi}$ outperforms f_Θ, f_{RE}, and f_{EL} in both IDP-v2 and LLC-v2. This demonstrates the effectiveness and superiority of our proposed representations in value function approximation and generalization. f_{EL} is significantly better than the f_Θ and f_{RE}, indicating the advantages of LPE structure and training. We can observe that contrastive policy representation f_{CL} performs poorly. We postulate that with less training data available at the 20% sampling ratio, the f_{CL} with emphasis on policy instance-level comparison suffers from higher evaluation error and generalization gap. The superiority of f_{V^π} compared to f_{EL} from the Table 2 demonstrates the effectiveness of alignment loss. This is because although both f_{V^π}, f_{EL} learn policy representation from the information of policy value, naive end-to-end training is less effective than alignment optimization which establishes the representation space based on the

Table 2: Performance of different policy abstractions in Off-policy Evaluation (OPE). The minimum value for each task is highlighted. Results are the mean ± a std over 10 and 5 trials (for *weak* and *strong* respectively). The f_{V^π} has lower T-error and G-gap on both the generalization tasks.

Env	Abstraction	Weak Generalization		Strong Generalization	
		T-error	G-gap	T-error	G-gap
IDP-v2	f_Θ	0.0059 ± 0.0008	0.0039 ± 0.0006	0.1592 ± 0.0107	**0.0778 ± 0.0437**
	f_{RE}	0.0056 ± 0.0009	0.0038 ± 0.0010	0.1676 ± 0.0086	0.1674 ± 0.0087
	f_{EL}	0.0048 ± 0.0003	0.0027 ± 0.0008	0.1783 ± 0.0060	0.1712 ± 0.0145
	f_{CL}	0.0067 ± 0.0010	0.0046 ± 0.0008	0.1567 ± 0.0081	0.1491 ± 0.0107
	f_π	0.0044 ± 0.0003	0.0025 ± 0.0006	0.1812 ± 0.0013	0.1803 ± 0.0011
	f_{P^π}	**0.0044 ± 0.0003**	0.0024 ± 0.0006	0.1789 ± 0.0045	0.1778 ± 0.0049
	f_{V^π}	0.0046 ± 0.0003	**0.0022 ± 0.0005**	**0.1320 ± 0.0093**	0.1295 ± 0.0114
LLC-v2	f_Θ	0.0018 ± 0.0005	0.0016 ± 0.0003	0.1898 ± 0.0237	0.0926 ± 0.1592
	f_{RE}	0.0028 ± 0.0007	0.0025 ± 0.0007	0.0729 ± 0.0197	0.0718 ± 0.0196
	f_{EL}	0.0017 ± 0.0004	0.0016 ± 0.0004	0.0656 ± 0.0088	0.0646 ± 0.0092
	f_{CL}	0.0035 ± 0.0005	0.0032 ± 0.0004	0.0589 ± 0.0176	0.0572 ± 0.0188
	f_π	0.0015 ± 0.0005	0.0013 ± 0.0005	0.1365 ± 0.0367	0.1318 ± 0.0332
	f_{P^π}	0.0015 ± 0.0004	0.0013 ± 0.0004	0.0905 ± 0.0402	0.0900 ± 0.0404
	f_{V^π}	**0.0014 ± 0.0003**	**0.0011 ± 0.0003**	**0.0473 ± 0.0043**	**0.0470 ± 0.0042**

policy metrics. In general, the value generalization results among our abstractions $f_\pi, f_{P^\pi}, f_{V^\pi}$ do not differ much. This is mainly because in the weak generalization setting, the unseen policies obey the same distribution as the training policies, thus posing less difficulty of value generalization.

Strong Generalization Scenario in OPE Now we move to the study in the strong generalization scenario and similarly the results are reported in Table 2 (Strong Generalization). Compared to the weak generalization scenario, the overall T-error and G-gap are significantly higher in the strong generalization scenario. This is reasonable because there is a larger performance difference between the training and unseen policies. In other words, the unseen policies belong to out-of-distribution data. The f_{V^π} obtains the lowest evaluation error on the two environments, which indicates the value-irrelevance abstraction with higher task relevance may be best suited for the strong generalization setting. For the explanation, since the objective of OPE lies at the value function approximation and generalization, we consider that the value-irrelevance principle of f_{V^π} is consistent to the objective and thus fits naturally.

With only low-performance policies for the training data at the 20% sampling ratio (*hardest*), the results of other abstractions including our proposed f_π and f_{P^π} on the task are poor. The main reason may be that under the strong generalization setting, there is a large data-shift between the training and unseen policies. f_π and f_{P^π} fails to learn a policy abstraction with generalization ability in the absence of diversity policies. Nevertheless, from the Table9, 10, as the sampling ratio increase and training policies become more diverse, the advantage of f_π and f_{P^π} over other policy abstractions gradually emerge. Moreover, in the hardest case, f_{P^π} is better than f_π, which shows that the influence-irrelevance policy abstraction may be a general policy abstraction option.

For the other baselines, f_Θ still shows few competition. For f_{RE}, f_{EL}, f_{CL}, they falls behind f_{V^π} while slightly outperforms f_{P^π} and f_π in Table 2. Such slight advantages no long holds in the settings of higher sampling ratios (i.e., 40% in Table 9 and 80% in Table 10). Unlike weak generalization scenario, f_{CL} is not so bad in strong generalization scenario. The main reason is that encountering hard policy evaluation tasks (Strong Generalization), other methods suffer from performance degradation and are no longer superior to contrastive learning. In contrast, contrastive learning based on policy instance-level comparison maintains a relatively good result.

Other Experiments In addition to Table 2, we provide more results under different settings of data amount in Appendix G.1, for both the weak and strong generalization scenarios. Other comprehensive studies (e.g., extrapolation behaviors, visualization) can be found in Appendix G.2,G.3.

7 CONCLUSION & LIMITATIONS

In this work, we introduce a unified policy abstraction theory, including three major types of policy abstraction, and corresponding policy metrics derived from the abstraction, as well as the analysis of their properties. We further propose a policy representation learning approach based on deep metric learning. We empirically evaluate the efficacy of different policy abstraction in both policy optimization (i.e., TRPO, DGES) and off-policy evaluation (OPE). For limitations and future work, we only provide the theory on the fineness of policy abstraction, while provide no theory on the optimality, although the optimality ought to depend on the downstream problem considered. For policy representation learning, the alignment loss and MMD metric are not the only choices; besides, other representation learning principles (Bardes et al., 2021) are potential.

REFERENCES

A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

K. Choromanski, M. Rowland, V. Sindhwani, R. Turner, and A. Weller. Structured evolution with compact architectures for scalable policy optimization. In *ICML*, pp. 970–978. PMLR, 2018.

F. Faccio, L. Kirsch, and J. Schmidhuber. Parameter-based value functions. *arXiv preprint arXiv:2006.09226*, 2020.

J. Fu, M. Norouzi, O. Nachum, G. Tucker, Z. Wang, A. Novikov, M. Yang, M. R. Zhang, Y. Chen, A. Kumar, et al. Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*, 2021.

F. Giunchiglia and T. Walsh. A theory of abstraction. *Artificial intelligence*, 57(2-3):323–389, 1992.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel two-sample test. *JMLR*, 13:723–773, 2012.

A. Grover, M. Al-Shedivat, J. K. Gupta, Y. Burda, and H. Edwards. Learning policy representations in multiagent systems. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1797–1806, 2018.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

J. Harb, T. Schaul, D. Precup, and P. Bacon. Policy evaluation networks. *arXiv preprint arXiv:2002.11833*, 2020.

J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

A. Kanervisto, T. Kinnunen, and V. Hautamäki. General characterization of agents by states they visit. *arXiv preprint arXiv:2012.01244*, 2020.

M. Kaya and H. S. Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for mdps. *ISAIM*, 4:5, 2006.

X. Ma, J. K. Gupta, and M. J. Kochenderfer. Normalizing flow model for policy representation in continuous action multi-agent systems. In *AAMAS*, pp. 1916–1918, 2020.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

M. Mutti, S. D. Col, and M. Restelli. Reward-free policy space compression for reinforcement learning. In *Unsupervised Reinforcement Learning Workshop at ICML*, 2021.

T. Nguyen-Tang, S. Gupta, and S. Venkatesh. Distributional reinforcement learning via moment matching. In *AAAI*, pp. 9144–9152, 2021.

A. Pacchiano, J. Parker-Holder, Y. Tang, K. Choromanski, A. Choromanska, and M. I. Jordan. Learning to score behaviors for guided policy optimization. In *ICML*, volume 119, pp. 7445–7454, 2020.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

R. Raileanu, M. Goldstein, A. Szlam, and R. Fergus. Fast adaptation to new environments via policy-dynamics value functions. In *ICML*, volume 119, pp. 7920–7931, 2020.

H. L. Royden. Real analysis / h. l. roiden. 1968.

T. Sang, H. Tang, Yi Ma, J. Hao, Y. Zheng, Z. Meng, B. Li, and Z. Wang. Pandr: Fast adaptation to new environments from offline experiences via decoupling policy and environment representations. *arXiv preprint arXiv:2204.02877*, 2022.

J. S Schreck, C. W Coley, and K. JM Bishop. Learning retrosynthetic planning through simulated experience. *ACS central science*, 5(6):970–981, 2019.

J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In *ICML*, volume 37, pp. 1889–1897, 2015.

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

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

D. Sejdinovic, A. Gretton, B. K. Sriperumbudur, and K. Fukumizu. Hypothesis testing using pairwise distances and associated kernels (with appendix). *arXiv preprint arXiv:1205.0411*, 2012.

L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine. AVID: learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019.

A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. A hilbert space embedding for distributions. In *ALT*, volume 4754 of *Lecture Notes in Computer Science*, pp. 13–31, 2007.

H. Tang, Z. Meng, J. Hao, C. Chen, D. Graves, D. Li, C. Yu, H. Mao, W. Liu, Y. Yang, W. Tao, and L. Wang. What about inputing policy in value function: Policy representation and policy-extended value function approximator. *arXiv preprint arXiv:2010.09536*, 2020.

J. Urain, D. Tateo, T. Ren, and J. Peters. Structured policy representation: Imposing stability in arbitrarily conditioned dynamic systems. *arXiv preprint arXiv:2012.06224*, 2020.

L V. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Y. Zheng, Z. Meng, J. Hao, Z. Zhang, T. Yang, and C. Fan. A deep bayesian policy reuse approach against non-stationary agents. In *NeurIPS 2018*, pp. 962–972, 2018.