Harnessing Distribution Ratio Estimators for Learning Agents with Quality and Diversity

Tanmay Gangwani
Dept. of Computer Science
UIUC
gangwan2@illinois.edu

Jian Peng
Dept. of Computer Science
UIUC
jianpeng@illinois.edu

Yuan Zhou
Dept. of ISE
UIUC
yuanz@illinois.edu

Abstract: Quality-Diversity (QD) is a concept from Neuroevolution with some intriguing applications to Reinforcement Learning. It facilitates learning a population of agents where each member is optimized to simultaneously accumulate high task-returns and exhibit behavioral diversity compared to other members. In this paper, we build on a recent kernel-based method for training a QD policy ensemble with Stein variational gradient descent. With kernels based on f-divergence between the stationary distributions of policies, we convert the problem to that of efficient estimation of the ratio of these stationary distributions. We then study various distribution ratio estimators used previously for off-policy evaluation and imitation and re-purpose them to compute the gradients for policies in an ensemble such that the resultant population is diverse and of high-quality [1].

Keywords: Reinforcement Learning, Quality-Diversity, Exploration-Exploitation

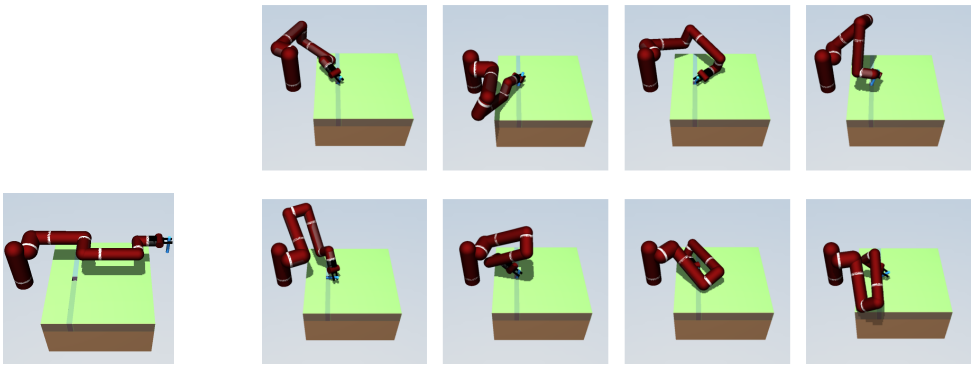
1 Introduction

The goal in Reinforcement Learning (RL) is to learn agents that maximize long-term environmental rewards. Deep RL, which uses deep neural networks as function approximators for the policy and value-functions, has achieved outstanding results on a wide variety of sequential decision making problems, with the barometer of success usually being the total returns accumulated by the final policy. Due to the intrinsic nature of direct reward maximization, seldom is the focus on how the *behavioral characteristics* of the trained agent compare with the other possible behaviors in the solution space. For instance, consider the robotic manipulator arm in Figure 1a and the peg-insertion task. Though the task description is simple, for a sufficiently flexible arm, there are numerous ways (positions of the joints and the end-effector) to insert the peg in the hole (Figure 1b). For reasons argued below, it is beneficial to learn these varied behaviors rather than aiming for the single most efficient solution dictated by the reward function. Quality-Diversity (QD) algorithms [1, 2] are prominent in the Neuroevolution literature as a means to generate many diverse behavioral *niches*, while ensuring that each niche is populated with individuals of the highest possible quality for that niche. When applied to RL, QD offers a principled approach for learning policies that are diverse, yet achieve high returns [3, 4].

Prior works have examined the benefits of uncovering diversity in how the task can be solved [5, 6, 7]. In these, an explicit diversity-maximization objective is incorporated into the RL algorithm to facilitate the learning of diverse *skills*. There are several important benefits of training a population of agents with diverse skills. Firstly, this is an efficient exploration strategy in sparse-reward environments as the agents can collectively achieve much wider coverage of the state-space, while reducing the susceptibility of RL to local optimal solutions caused by deceptive rewards [4, 8]. Secondly, the acquired skills could be leveraged for accelerated learning in downstream tasks, for example, by composing the skills to solve long-horizon tasks via hierarchical RL [9, 6]. Diversity also helps in the transfer learning of RL policies across environments that may have discrepancies such as system dynamics mismatch. Having a repertoire of skills is useful when knowledege transfer is done to a target environment that has constraints on the set of feasible behaviors [10].

[1] Code for this paper is available at https://github.com/tgangwani/QDAgents

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.



(a) (b)

Figure 1: (a) MuJoCo model of a 7 DOF arm based on the Sawyer robot, inspired by Chen et al. [21]; (b) Policies that achieve the peg-insertion task in different ways. These policies are sampled from a single ensemble trained with the algorithm QD-*DualDICE*-JS (explained in Section 5).

A policy π is characterized by its occupancy measure ρ_π [11], which is the stationary distribution over the state-action pairs that π encounters when navigating the environment. Given two policies π, β, the ratio of their stationary distributions $\zeta = \rho_\pi / \rho_\beta$ is a well-studied quantity in RL. Estimates of the distribution ratio are useful for off-policy evaluation [12, 13] (where the goal is to evaluate the performance of π using fixed data generated from β), policy optimization [14, 15] and off-policy imitation learning [16]. In this paper, we examine the use of distribution ratio estimators for learning a diverse policy ensemble with high returns (a QD ensemble). We build on the approach introduced by Liu et al. [7]. Using Stein variational gradient descent (SVGD) [17] as the inference method, the authors construct an update rule that includes the policy-gradient on the environmental rewards (for high quality) and a kernel-induced repulsive force gradient (for high diversity). This kernel-based algorithm is naturally impacted by the choice of the kernel. We begin with generalizing the Stein variational policy gradient (SVPG) objective [7] by using as kernels the negative exponents of an f-divergence between the stationary distributions of two policies, and discuss key properties such as positive-definiteness of kernels. For kernels based on the Jenson-Shannon and Symmetric Kullback-Leibler divergences, we show how the complete SVPG gradient can be recast in terms of the *ratio of the stationary distributions* (ζ) between policies. Then, to estimate these ratios, and hence the SVPG gradient, we study three recently proposed distribution ratio estimators for off-policy evaluation and imitation learning. These are *DualDICE* [18], *ValueDICE* [16] and *GenDICE* [19]. Additionally, we describe a fourth estimator based on Noise-Constrastive Estimation [20].

We perform experiments on various tasks to get a measure of the effectiveness of our proposed approach in generating diverse behaviors with high returns. We also evaluate on tasks with deceptive rewards and those which lack an external reward signal to further illuminate the benefits of QD.

2 Preliminaries

RL Notations. The environment is modeled as an infinite-horizon, discrete-time Markov Decision Process (MDP), represented by the tuple ($\mathcal{S}$, $\mathcal{A}$, μ_0, r, p, γ), where $\mathcal{S}$ is the state-space, $\mathcal{A}$ is the action-space, $\gamma \in [0, 1)$ is the discount factor, and μ_0 denotes the initial state distribution. Given an action a_t sampled from a stochastic policy $\pi_\theta(a_t|s_t)$, the next state is sampled from the transition dynamics distribution, $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$, and the agent receives a reward $r(s_t, a_t)$ determined by the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The RL objective is to maximize the expected discounted sum of rewards, $\eta(\pi) = (1-\gamma)\mathbb{E}_{\mu_0,p,\pi}\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\big]$.

Distribution Ratio (ζ)**.** The occupancy measure [11], or the stationary discounted state-action visitation distribution of the policy π is defined as $\rho_\pi(s, a) = (1-\gamma)\pi(a|s) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t{=}s|\pi)$, where $\mathbb{P}(s_t{=}s|\pi)$ is the probability of being in state s at time t, when starting in state $s_0 \sim \mu_0$ and using π thereafter. The stationary distribution [2] affords a convenient rewriting of the expected policy return as $\eta(\pi) = \mathbb{E}_{\rho_\pi}[r(s, a)]$, and the gradient is provided by the policy gradient theorem [22] as

[2]Throughout, stationary *discounted* distribution is shorthanded with stationary distribution for brevity

Name of f-divergence	Formula D_f(P, Q)	Generator $f(u)$ with $f(1)=0$	Is the kernel $e^{-D_f(P,Q)/T}$ PD?
Jenson-Shannon	$\int_x \frac{p(x)}{2}\log\frac{2p(x)}{p(x)+q(x)} + \frac{q(x)}{2}\log\frac{2q(x)}{p(x)+q(x)}dx$	$\frac{u}{2}\log u - \frac{(1+u)}{2}\log\frac{1+u}{2}$	Yes
Triangular Discrimination	$\int_x \frac{(p(x)-q(x))^2}{p(x)+q(x)}dx$	$\frac{(u-1)^2}{u+1}$	Yes
Squared Hellinger	$\int_x(\sqrt{p(x)}-\sqrt{q(x)})^2dx$	$(\sqrt{u}-1)^2$	Yes
Total Variation	$\frac{1}{2}\int_x \lvert p(x)-q(x)\rvert dx$	$\frac{1}{2}\lvert u-1\rvert$	Yes
Kullback-Leibler	$\int_x p(x)\log\frac{p(x)}{q(x)}dx$	$-\log u$	No
Reverse Kullback-Leibler	$\int_x q(x)\log\frac{q(x)}{p(x)}dx$	$u\log u$	No
Symmetric Kullback-Leibler	$\int_x p(x)\log\frac{p(x)}{q(x)} + q(x)\log\frac{q(x)}{p(x)}dx$	$(u-1)\log u$	No

Table 1: f-divergences and positive-definiteness of the negative exponential kernels.

$\nabla_\theta\eta(\pi) = \mathbb{E}_{\rho_\pi}\big[\nabla_\theta \log \pi_\theta(a|s)Q^\pi(s,a)\big]$, where Q^π is the state-action value function. For two policies π_i and π_j, we denote the ratio of their stationary distributions by $\zeta_{ij}(s,a) = \rho_{\pi_i}(s,a)/\rho_{\pi_j}(s,a)$. This ratio is widely applicable for off-policy evaluation as it enables estimating the expected returns of π_i using a fixed dataset $\mathcal{D}$ of transitions generated from a different behavioral policy π_j, since $\eta(\pi_i) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\zeta_{ij}(s,a)r(s,a)]$, where $\mathcal{D}$ is an empirical estimate of ρ_{π_j}.

3 Methods

3.1 QD objective and its solution based on variational inference

QD when applied to policy search entails learning multiple policies that all accumulate high environmental rewards during an episode, but the agents accomplish this using diversified strategies, such as navigating dissimilar regions of the state-action space. Formally, policy search with QD could be defined as learning a *distribution* over the policy parameters (θ) that maximizes the RL-objective in expectation, while maintaining a high-entropy ($\mathcal{H}$) in the parameter-space:

$$\max_q \mathbb{E}_{\theta\sim q}[\eta(\theta)] + \mathcal{H}(q)\,; \qquad \mathcal{H}(q) = \mathbb{E}_{\theta\sim q}[-\log q(\theta)] \tag{1}$$

Solving the objective in Equation 1 analytically yields the following energy-based optimal parameter distribution: $q^*(\theta) = \exp(\eta(\theta))/Z_{q^*}$, where Z_{q^*} is the normalization constant. Let $p(\theta)$ define a trainable distribution over the policy parameters that we seek to optimize to be close (*w.r.t.* the KL-divergence) to the target distribution q^*. Representing $p(\theta)$ with a mixture of delta distributions, the variational objective is:

$$\min_p D_{\text{KL}}\big[p \;||\; \exp(\eta(\theta))/Z_{q^*}\big]\,; \qquad p(\theta) = \frac{1}{n}\sum_{i=1}^{n}\delta(\theta=\theta_i)$$

Here $\{\theta_i\}_1^n$ denotes a policy ensemble with n discrete policies that constitute the p distribution. Stein variational gradient descent (SVGD; Liu and Wang [17]) provides an efficient solution to obtain an approximate gradient on the p distribution. Suppose we perturb each policy θ_i with $\Delta\theta_i$ such that the induced KL between p and q is reduced. The optimal perturbation direction, in the unit ball of a reproducing kernel Hilbert space associated with a kernel function k, that maximally decreases the KL is given by [17]:

$$\Delta\theta = \mathbb{E}_{\theta'\sim p}\big[\nabla_{\theta'}\log q^*(\theta')k(\theta',\theta) + \nabla_{\theta'}k(\theta',\theta)\big]$$

Using this result and the energy-based form of the target distribution q^*, SVPG [7] iteratively updates the policies with the following rule to learn a policy ensemble with QD behavior:

$$\theta_i \leftarrow \theta_i + \epsilon\Delta\theta_i, \qquad \Delta\theta_i = \frac{1}{n}\sum_{j=1}^{n}\Big[\underbrace{\nabla_{\theta_j}\eta(\pi_{\theta_j})k(\theta_j,\theta_i)}_{\text{Quality-enforcing}} + \underbrace{\nabla_{\theta_j}k(\theta_j,\theta_i)}_{\text{Diversity-enforcing}}\Big] \tag{2}$$

3.2 Negative exponents of f-divergences as kernels

The positive definite (PD) kernel function k in Equation 2 is an algorithmic design choice. There are two considerations. It should be possible to efficiently compute $k(\theta_j,\theta_i)$ for any two policies $(\pi_{\theta_j},\pi_{\theta_i})$ as well as its gradient *w.r.t.* the policy parameters; and the function should be sufficiently

expressive to capture the complex interactions between policy behaviors. Liu et al. [7] employ a Gaussian RBF kernel $k(\theta_j, \theta_i) = \exp(-\|\theta_j - \theta_i\|_2^2/h)$, with a dynamically adapted bandwidth h. Gangwani et al. [8] suggest replacing the Euclidean distance in the parameter space with a statistical distance in the stationary distribution space, and use $k(\theta_j, \theta_i) = \exp(-D_{\text{JS}}(\rho_{\pi_{\theta_j}}, \rho_{\pi_{\theta_i}})/T)$, where D_{JS} is the Jenson-Shannon divergence and T is the temperature. D_{JS} is a member of a broader class of divergences known as Ali-Silvey distances or f-divergences [23]. Given two distributions with continuous densities $p(x)$ and $q(x)$ over the support $\mathcal{X}$, the f-divergence between them is defined as:

$$D_f(p \,||\, q) = \int_{\mathcal{X}} q(x) f(\frac{p(x)}{q(x)}) dx$$

where $f : \mathbb{R}_+ \to \mathbb{R}$ is a convex, lower-semicontinuous function such that $f(1) = 0$. Different choices for the function f recover the well-known divergences. Although generalizing the kernel function as $k_f(\theta_j, \theta_i) = \exp(-D_f(\rho_{\pi_{\theta_j}}, \rho_{\pi_{\theta_i}})/T)$ may seem like a natural extension, for some f-divergences, $k_f(\theta_j, \theta_i)$ is not PD, and hence not a proper kernel from a theoretical standpoint. For instance, while k_{JS} is PD, kernels with other divergences commonly used for policy learning (KL, Reverse-KL) are not. Table 1 provides details on the various divergences that define PD and non-PD kernels after negative exponentiation. The first four divergences in Table 1 are squared metrics (*i.e.* $\sqrt{D_f}$ is a true metric) and the proof of positive-definiteness of the corresponding kernels k_f is provided in Hein and Bousquet [24]. Inserting $k_f(\theta_j, \theta_i)$ in Equation 2, the SVPG gradient becomes:

$$\Delta\theta_i = \frac{1}{n} \sum_{j=1}^{n} \exp(-\underbrace{D_f(\rho_{\pi_{\theta_j}}, \rho_{\pi_{\theta_i}})}_{\text{Divergence value}} / T) \Big[\underbrace{\nabla_{\theta_j} \eta(\pi_{\theta_j})}_{\text{Policy gradient}} - \frac{1}{T} \underbrace{\nabla_{\theta_j} D_f(\rho_{\pi_{\theta_j}}, \rho_{\pi_{\theta_i}})}_{\text{Divergence gradient}} \Big] \tag{3}$$

This provides a general framework to evaluate the SVPG gradient for learning a QD policy ensemble. Depending on the f-divergence and the method for estimating its value and gradient, several approaches are possible, a few of which we will discuss. We use the shorthand notation ρ_i for the stationary distribution of the policy π_{θ_i}. Then $\zeta_{ij}(s, a) = \rho_i(s, a)/\rho_j(s, a)$ is the distribution ratio for two given policies, and would be the pivotal quantity in the exposition that follows. Next, we rewrite two kernels (and their gradient *w.r.t.* the policy parameters) in terms of ζ, before elucidating several methods to estimate ζ for use in a practical RL algorithm to generate a QD policy ensemble.

The k_{JS} and k_{KLS} kernels. While k_{JS} is a PD kernel, k_{KLS} is not since $\sqrt{D_{\text{KLS}}}$ is not a metric as it does not satisfy the triangle inequality. Although positive-definiteness is desirable, non-PD kernels may yet achieve good performance in practice, as shown in Moreno et al. [25], where SVM classification with a non-PD kernel leads to better accuracy than provably PD kernels. Both k_{JS} and k_{KLS} afford the benefit that the divergence value and gradient (in Equation 3) can be evaluated in terms of the distribution ratio ζ_{ij}. Using the definitions from Table 1, we express D_{JS} and D_{KLS} as:

$$D_{\text{JS}}(\rho_i, \rho_j) = \frac{1}{2}\mathbb{E}_{\rho_i(s,a)} \log \frac{\zeta_{ij}(s, a)}{1 + \zeta_{ij}(s, a)} + \frac{1}{2}\mathbb{E}_{\rho_j(s,a)} \log \frac{1}{1 + \zeta_{ij}(s, a)} + \log 2$$

$$D_{\text{KLS}}(\rho_i, \rho_j) = \mathbb{E}_{\rho_i(s,a)} \log \zeta_{ij}(s, a) - \mathbb{E}_{\rho_j(s,a)} \log \zeta_{ij}(s, a)$$

The SVPG gradient involves the gradient of the f-divergence *w.r.t.* the policy parameters (θ). For D_{JS} and D_{KLS}, the gradient can be written using ζ as follows:

$$\nabla_{\theta_j} D_{\text{JS}} = \nabla_{\theta_j} \mathbb{E}_{\rho_j} \underbrace{-(1/2) \log[1 + \zeta_{ij}(s, a)]}_{r(s,a)}; \quad \nabla_{\theta_j} D_{\text{KLS}} = \nabla_{\theta_j} \mathbb{E}_{\rho_j} \underbrace{[-\zeta_{ij}(s, a) - \log \zeta_{ij}(s, a)]}_{r(s,a)} \tag{4}$$

The proofs for these are included in Appendix A.1. In practice, these gradients could be estimated using the policy-gradient theorem [22] with the appropriate term as the reward function (marked as $r(s, a)$ above). It is thus evident that a reasonable estimation of the distribution ratio yields a good approximation of the SVPG gradient (Equation 3), which could then be applied to the policy parameters to learn a QD ensemble. We now discuss methods to estimate ζ efficiently from samples.

3.3 Estimating the distribution ratio ζ

We start with Noise-Constrastive Estimation (NCE) [20] which has found wide applicability in representation learning, natural language processing and image synthesis, among others. We then examine three distribution ratio estimators – *DualDICE* [18] and *GenDICE* [19] were recently proposed

for behavior-agnostic off-policy evaluation, and *ValueDICE* [16] enables imitating expert trajectories without requiring additional policy rollouts in the environment.

NCE. It provides a method to learn an estimator $\widetilde{\rho}_i(s, a; \omega)$ for the stationary distribution of any policy π_i in the ensemble. NCE uses a noise distribution $p_N(s, a)$ and frames the following binary classification objective:

$$\max_{\omega} \mathbb{E}_{\rho_i} \log \frac{\widetilde{\rho}_i(s, a; \omega)}{\widetilde{\rho}_i(s, a; \omega) + p_N(s, a)} + \mathbb{E}_{p_N} \log \frac{p_N(s, a)}{\widetilde{\rho}_i(s, a; \omega) + p_N(s, a)}$$

Gutmann and Hyvärinen [20] show that under mild assumption on the noise distribution, $\widetilde{\rho}_i(\cdot; \omega)$ converges to the true density ρ_i in the limit of infinite amount of samples. They further note that for practical efficiency, it is desirable to select a noise distribution that is easy to sample from, and that is not too far from the true unknown data distribution ρ_i. Consequently, for learning the estimator for policy i, we use a uniform mixture of stationary distributions of the remaining $(n-1)$ policies in the ensemble as the constrastive noise, *i.e.*, $p_N(s, a) = (1/(n-1)) \sum_{j \neq i} \rho_j(s, a)$. The distribution ratio for a pair of policies can then be computed as $\zeta_{ij}(s, a) = \widetilde{\rho}_i(s, a)/\widetilde{\rho}_j(s, a)$.

DualDICE. Nachum et al. [18] propose a convex optimization problem that gives the distribution ratio as its optimal solution:

$$\zeta_{ij} = \arg\min_{x: \mathcal{S} \times \mathcal{A} \to \mathbb{R}} \frac{1}{2} \mathbb{E}_{(s,a) \sim \rho_j}[x(s, a)^2] - \mathbb{E}_{(s,a) \sim \rho_i}[x(s, a)] \tag{5}$$

The expression is then simplified with the following *change-of-variables* trick. Define a variable $\nu(s, a)$ and the operator $\mathcal{B}^{\pi_i} \nu(s, a) = \gamma \mathbb{E}_{s' \sim p(\cdot|s,a), a' \sim \pi_i(s')}[\nu(s', a')]$. Using $x(s, a) = \nu(s, a) - \mathcal{B}^{\pi_i} \nu(s, a)$, the second expectation in Equation 5 telescopes and conveniently reduces into an expectation over the initial states. The transformed objective is:

$$\min_{\nu: \mathcal{S} \times \mathcal{A} \to \mathbb{R}} \frac{1}{2} \mathbb{E}_{(s,a) \sim \rho_j}[(\nu - \mathcal{B}^{\pi_i} \nu)(s, a)^2] - (1 - \gamma) \mathbb{E}_{\substack{s_0 \sim \mu_0 \\ a_0 \sim \pi_i(s_0)}}[\nu(s_0, a_0)]$$

Given an optimal solution ν^* for this equation, the distribution ratio is recovered with $\zeta_{ij}(s, a) = (\nu^* - \mathcal{B}^{\pi_i} \nu^*)(s, a)$. Further, to alleviate the bias in the sample-based Monte-Carlo estimate of the gradient, Nachum et al. [18] suggest the use of Fenchel conjugates. The final objective is a min-max saddle-point optimization that directly provides the distribution ratio $\zeta_{ij}(s, a)$. Appendix A.2 includes the details.

ValueDICE. The Donsker-Varadhan representation [26] of the KL-divergence is given by:

$$D_{\text{KL}}(\rho_i || \rho_j) = \sup_{x: \mathcal{S} \times \mathcal{A} \to \mathbb{R}} \mathbb{E}_{(s,a) \sim \rho_i}[x(s, a)] - \log \mathbb{E}_{(s,a) \sim \rho_j}[e^{x(s,a)}]$$

In *ValueDICE* [16], the authors use the fact that the optimality in the above equation is achieved at $x^*(s, a) = \log \zeta_{ij}(s, a) + C$, for some constant $C \in \mathbb{R}$. The proof is included in Appendix A.3 for completeness. Therefore, a method to obtain ζ_{ij} is to first solve for x^* (written as a minimization):

$$x^* = \arg\min_{x: \mathcal{S} \times \mathcal{A} \to \mathbb{R}} \log \mathbb{E}_{(s,a) \sim \rho_j}[e^{x(s,a)}] - \mathbb{E}_{(s,a) \sim \rho_i}[x(s, a)]$$

With the same *change-of-variables* trick from *DualDICE*, $x(s, a) = \nu(s, a) - \mathcal{B}^{\pi_i} \nu(s, a)$, the second expectation over $\rho_i(s, a)$ is transformed into an expectation over the initial states:

$$\min_{\nu: \mathcal{S} \times \mathcal{A} \to \mathbb{R}} \log \mathbb{E}_{(s,a) \sim \rho_j}[e^{\nu(s,a) - \mathcal{B}^{\pi_i} \nu(s,a)}] - (1 - \gamma) \mathbb{E}_{\substack{s_0 \sim \mu_0 \\ a_0 \sim \pi_i(s_0)}}[\nu(s_0, a_0)]$$

Different from *DualDICE*, *ValueDICE* avoids the min-max saddle-point optimization by eschewing the use of Fenchel conjugates to remove the bias in the sample-based gradient. The log distribution ratio is calculated (up to a constant shift) from ν^* as, $\log \zeta_{ij}(s, a) = \nu^*(s, a) - \mathcal{B}^{\pi_i} \nu^*(s, a)$.

GenDICE. It is known that the stationary distribution for a policy π_i satisfies the following Bellman flow constraint:

$$\rho_i(s', a') = (1 - \gamma) \mu_0(s') \pi_i(a'|s') + \gamma \int \pi_i(a'|s') p(s'|s, a) \rho_i(s, a) ds da; \quad \forall (s', a') \in \mathcal{S} \times \mathcal{A}$$

Algorithm 1: Pseudo code for learning a QD ensemble

Initialize policy ensemble $\{\pi_i\}_1^n$
Initialize networks to estimate ζ_{ij} ▷ Parameterization depends on the method (Section 3.3)
for *each iteration* **do**
  Sample rollouts with $\pi_i \ \forall i$
  Update all ζ_{ij} estimation networks ▷ Objective depends on the method (Section 3.3)
  Use ζ_{ij} to compute the divergence value and divergence gradient ▷ (D_{JS} or D_{KLS})
  Update each π_i with the corresponding SVPG gradient ▷ (Equation 3)
end

This could be re-written using the distribution ratio ζ_{ij} as:

$$\rho_j(s',a')\zeta_{ij}(s',a') = \underbrace{(1-\gamma)\mu_0(s')\pi_i(a'|s') + \gamma \int \pi_i(a'|s')p(s'|s,a)\zeta_{ij}(s,a)\rho_j(s,a)dsda}_{\mathcal{T}_{(\pi_i,\rho_j)}\circ\zeta_{ij}} \quad (6)$$

Zhang et al. [19] parameterize ζ_θ and suggest to estimate it by minimizing the f-divergence between the distributions (with support on $\mathcal{S}\times\mathcal{A}$) on the two sides of Equation 6, namely $\rho_j.\zeta_\theta$ and $\mathcal{T}_{(\pi_i,\rho_j)}\circ\zeta_\theta$, where the notation $\mathcal{T}_{(\pi_i,\rho_j)}$ denotes the distribution operator on the RHS in Equation 6. The objective, which further includes a penalty regularizer on ζ_θ to prevent degenerate solutions, is:

$$\zeta_{ij} = \arg\min_\theta D_f\left(\mathcal{T}_{(\pi_i,\rho_j)}\circ\zeta_\theta \,||\, \rho_j.\zeta_\theta\right) + \frac{\lambda}{2}\left(\mathbb{E}_{\rho_j}[\zeta_\theta]-1\right)^2$$

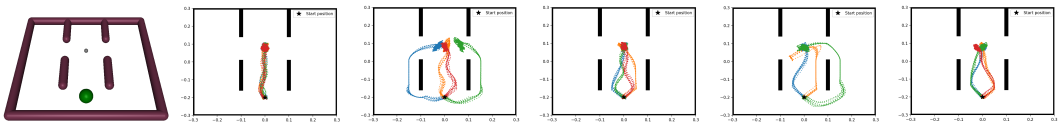
Similar to *DualDICE*, Fenchel conjugates are used to obtain unbiased gradient estimates, resulting in a min-max saddle-point optimization. The final objective, with χ^2 as the f-divergence is included in Appendix A.4.

Overall Algorithm.. We summarize our complete approach for training a QD policy ensemble in Algorithm 1. In each iteration, we sample transitions in the environment using the policies in the ensemble and update the networks that facilitate estimation of the distribution ratios ζ_{ij}. The type of network(s) and the update rule is determined by the estimator choice. To form the SVPG gradient (Equation 3), the current value of ζ is used to compute the divergence value and the divergence gradient. The latter, as shown by Equation 4, is equivalent to the policy gradient with a distinctive reward function. We use the clipped PPO algorithm [27] for the policy gradient, although other on-policy and off-policy RL methods are also applicable.

4 Related work

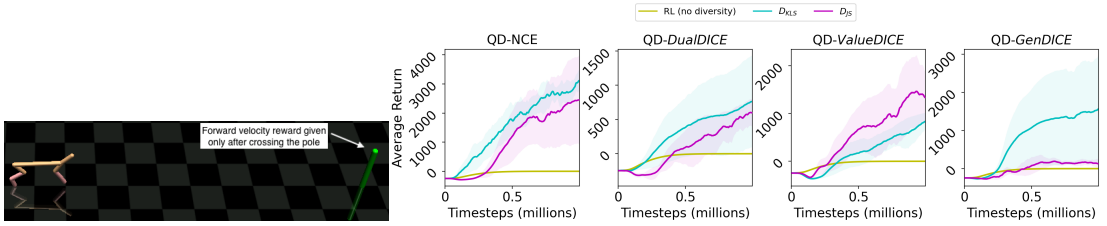
Neuroevolution methods inspired by Quality-Diversity [1] have been proposed to efficiently manage the exploration-exploitation trade-off in RL. Conti et al. [4] augment evolution strategies [28] such that the fitness of a particle is computed by a weighted combination of the performance and novelty components. The novelty is determined based on a chosen behavior characterization (BC) metric. In MAP-Elites [3], the entire behavior space is divided into a discrete grid, where each grid-dimension represents a BC. The algorithm then fills each grid cell with the highest quality solution possible for that cell. Recent methods integrate RL gradients with concepts from evolutionary computation (*e.g. random mutations*) to learn diverse exploratory agents [29, 30] or to discover coordination strategies for multi-agent RL [31].

Diversity Maximization in RL. To aid exploration in sparse-reward tasks, Hong et al. [5] encourage the current policy to be diverse compared to an archive of past policies, by maximizing a distance metric in the action space. Expanding on this idea, Doan et al. [32] ensure sufficient diversity in a population by using *operators* for attraction and repulsion between agents. Towards learning diverse skills even in the absence of an external reward signal, maximization of the mutual information between the latent skill and the state-visitation of the skill-conditioned policy has been proposed [9, 6]. This is achieved by training a neural network to estimate the latent skill posterior, which provides proxy rewards for the policy. Our work aims to broaden the SVPG algorithm [7] for learning a QD policy ensemble. We substitute the parameter-space RBF kernel used in their method with negative



(a) (b) Standard RL (c) QD-NCE-JS (d) QD-NCE-KLS (e) QD-*GenDICE*-JS (f) QD-*GenDICE*-KLS

Figure 2: 2D Maze navigation task along with trajectories (state-visitations) for several methods.



(a) (b)

Figure 3: (a) Modified Half-Cheetah task that introduces multi-modality due to deceptive rewards; (b) Contrasting performance of standard RL (no diversity) with QD method in Algorithm 1.

exponents of f-divergences, and employ distribution ratio estimation techniques to approximate the ensuing gradient on the policy parameters. Gangwani et al. [8] avail SVPG to improve self-imitation learning. Their procedure bears some resemblance to our NCE-based ratio estimation, though, in contrast to them, we use a mixture of stationary distributions as the contrastive noise.

5 Experiments

In this section, we train policy ensembles in various environments with continuous state- and action-space. We evaluate the different distribution ratio estimators and divergence metrics from Section 3. For ease of exposition, the algorithms are abbreviated as **QD-**$\{$***ratio-estimator***$\}$**-**$\{$***divergence***$\}$, hence QD-NCE-JS, for instance, is Algorithm 1 instantiated with $\exp(-D_{\text{JS}}(\rho_j, \rho_i)/T)$ as the kernel for SVPG, and NCE as the estimator for ζ. Our goal is to gauge the effectiveness of our approach in producing diverse, high-quality behaviors and suitably handling tasks with deceptive rewards. We also compare the NCE and DICE-based estimators on a quantitative metric correlated with behavioral diversity. The hyperparameters and implementation details are included in Appendix A.6.

Qualitative Assessment of QD Behavior. We visualize the diversity of the learned skills in two environments – a robotic manipulator arm [21] and a 2D maze goal navigation task. The robotic arm models a 7 DOF Sawyer robot and is implemented in MuJoCo. For the peg-insertion task, we train a QD ensemble of 10 policies using the exponential of the negative Euclidean distance between the peg and the hole as the per-step reward for RL (the quality measure). Figure 1b shows some of the policies from a single ensemble trained with QD-*DualDICE*-JS. We find that while all the policies insert the peg in the hole, the final positions of the joints (marked with white rings) and the end-effector are markedly different. The resultant behaviors have dissimilar torque demands on the various joints, which is advantageous in scenarios such as transfer learning to a robot with system dynamics discrepancies. Figure 2a depicts a 2D navigation task with the start position (green ball at center bottom) and the goal location (small grey circle in the center of the maze). The per-step RL reward is the exponential of the negative Euclidean distance to the goal. We train an ensemble of 10 policies and plot final trajectories from some of them (each policy colored differently). Figure 2b shows results with the standard RL method, *i.e*, no diversity enforcement; the trajectories achieve the best possible cumulative returns but exhibit identical behavior. Figure 2c- 2f plots the paths for policies learned with the QD algorithm (specific instantiation mentioned in the caption). Though the cumulative returns now are lower than those with standard RL, the policies are noticeably more exploratory and cover large portions of the state-space.

Multi-modal Locomotion with Deceptive Rewards. One of the crucial benefits of learning a QD ensemble is that it potentially avoids the local optimum trap in the policy-search landscape due

to deceptive rewards – if one policy gets stuck, the explicit diversity enforcement prevents other policies in the ensemble from the same fate. We evaluate this hypothesis with the Half-Cheetah locomotion task from OpenAI Gym [33]. We modify the task such that the forward velocity reward is only given to the agent once the center-of-mass of the bot is beyond a certain threshold distance (d). Concretely, $r_t = vel_{x(t)} * \mathbb{1}(pos_{x(t)} >= d) - 0.1 * \|a_t\|_2^2$, where the second term penalizes large actions and is the default from Gym. Figure 3a is a rendering of the task. This change introduces multi-modality for policy optimization with a locally optimal solution to stand still at the starting location to avoid any action penalty. We compare the performance of the QD ensembles with a baseline standard-RL ensemble. The standard-RL ensemble has the same size as others but the constituent policies do not have any interactions; they apply independently computed gradients. For all baseline and QD ensembles, we select the policy with the highest cumulative returns after training and plot its learning curve in Figure 3b. We observe that the baseline RL (no diversity) latches onto the deceptive reward of minimizing the action penalty and gets stuck, achieving a cumulative return close to zero. In contrast, the diversity enforcing mechanism in the QD* ensemble enables *at-least* one member to reach the alternative mode where high forward velocity rewards are attained. This is evident in the final score accumulated by the member selected from each ensemble.

Quantitative Comparison of the Estimators. While the previous experiment exhibits that the NCE and DICE-based estimators can provide adequate diversity impetus, it does not provide insights about the comparative efficiency of the estimators in generating behavioral diversity in the trained ensemble. This is because the forward velocity reward is a *quality metric*, which is usually not aligned with the measure of diversity. For instance, an estimator may produce a policy that makes the Half-Cheetah run backwards—this is much desired from the diversity perspective but would perform badly on the quality metric that rewards forward motion. To evaluate the efficacy of our estimators for producing diverse behaviors, and also for meaningful comparison with prior work [6, 8], we define a *diversity metric* as follows. For two locomotion tasks from Gym (Hopper and Walker-2d), we train policy ensembles without any environmental rewards. Thus, the gradient from the quality-enforcing component in Equation 2 is absent and the QD ensemble is trained only to maximize diversity. Post-training, we generate a few trajectories with all the constituent policies and plot a histogram with the velocity of the center-of-mass of the bot on the x-axis and the respective counts on the y-axis. We define the diversity metric to be the *variance* of this histogram. Intuitively, higher variance in the velocity of the bot is indicative of stronger behavioral diversity in the trained ensemble. Table 2 evaluates the various estimator on this diversity metric. We note that DICE-based estimators generally outperform NCE. Our intuition for this observation is that since NCE is an on-policy estimator (in contrast with the DICE-based estimators, which are off-policy), the availability of limited on-policy data in each iteration of Algorithm 1 has an impact on the efficiency of NCE. Lastly, many of the QD* methods compare favorably to the prior methods for learning diverse skills without environmental rewards [6, 8].

	Hist. Variance ↑	
Method	Walker-2d	Hopper
QD-DD-JS	**1.36**	0.45
QD-VD-JS	1.33	**0.50**
QD-GD-JS	0.63	0.14
QD-NCE-JS	0.13	0.11
QD-DD-KLS	0.10	0.10
QD-VD-KLS	0.24	0.45
QD-GD-KLS	0.07	0.40
QD-NCE-KLS	0.14	0.28
Gangwani et al. [8]	0.10	0.08
DIAYN [6]	0.22	0.11

Table 2: Diversity metric (histogram variance) with different estimators. Higher is better. Mnemonic: DD=*DualDICE*, VD=*ValueDICE*, GD=*GenDICE*

6 Conclusion and Future Work

In this paper, we study methods to learn diverse and high-return policies. We extend the kernel-based SVPG algorithm with kernels based on f-divergence between the stationary distributions of policies. For kernels based on D_{JS} and D_{KLS}, we show that the problem reduces to that of efficient estimation of the ratio of the stationary distributions between policies. To compute these ratios, and consequently the SVPG gradient, we harness noise-contrastive estimation and several distribution ratio estimators widely used for off-policy evaluation and imitation learning. Experimental evaluation with continuous state- and action-space environments demonstrates that the approach is capable of generating diverse high-quality skills, assists in multi-modal environments with deceptive rewards, and provides a constructive learning signal when the external rewards are absent. Our algorithmic framework is general enough to accommodate any distribution ratio estimator. Utilizing future research on these estimators for improving the efficiency of QD training is an interesting direction, along with investigating which other f-divergences or integral probability metrics (IPMs such as the Wasserstein distance and the Maximum Mean Discrepancy) between stationary distributions could be incorporated into the framework.

Acknowledgments

This work is supported by the National Science Foundation under grants OAC-1835669 and CCF-2006526. Yuan Zhou is supported in part by a Ye Grant and a JPMorgan Chase AI Research Faculty Research Award.

References

[1] J. K. Pugh, L. B. Soros, and K. O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

[2] A. Cully and Y. Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2017.

[3] J.-B. Mouret and J. Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

[4] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. O. Stanley, and J. Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *arXiv preprint arXiv:1712.06560*, 2017.

[5] Z.-W. Hong, T.-Y. Shann, S.-Y. Su, Y.-H. Chang, T.-J. Fu, and C.-Y. Lee. Diversity-driven exploration strategy for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 10489–10500, 2018.

[6] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[7] Y. Liu, P. Ramachandran, Q. Liu, and J. Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.

[8] T. Gangwani, Q. Liu, and J. Peng. Learning self-imitating diverse policies. *arXiv preprint arXiv:1805.10309*, 2018.

[9] C. Florensa, Y. Duan, and P. Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*, 2017.

[10] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[11] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.

[12] D. Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[13] P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.

[14] R. S. Sutton, A. R. Mahmood, and M. White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1):2603–2631, 2016.

[15] Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill. Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*, 2019.

[16] I. Kostrikov, O. Nachum, and J. Tompson. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032*, 2019.

[17] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

[18] O. Nachum, Y. Chow, B. Dai, and L. Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, pages 2318–2328, 2019.

[19] R. Zhang, B. Dai, L. Li, and D. Schuurmans. Gendice: Generalized offline estimation of stationary values. *arXiv preprint arXiv:2002.09072*, 2020.

[20] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[21] T. Chen, A. Murali, and A. Gupta. Hardware conditioned policies for multi-robot transfer learning. In *Advances in Neural Information Processing Systems*, pages 9333–9344, 2018.

[22] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[23] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

[24] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. 2004.

[25] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *Advances in neural information processing systems*, pages 1385–1392, 2004.

[26] M. D. Donsker and S. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

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

[28] T. Salimans, J. Ho, X. Chen, and I. Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[29] S. Khadka, S. Majumdar, T. Nassar, Z. Dwiel, E. Tumer, S. Miret, Y. Liu, and K. Tumer. Collaborative evolutionary reinforcement learning. *arXiv preprint arXiv:1905.00976*, 2019.

[30] S. Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel. Emergent coordination through competition. *arXiv preprint arXiv:1902.07151*, 2019.

[31] S. Khadka, S. Majumdar, S. Miret, S. McAleer, and K. Tumer. Evolutionary reinforcement learning for sample-efficient multiagent coordination. *arXiv preprint arXiv:1906.07315*, 2019.

[32] T. Doan, B. Mazoure, A. Durand, J. Pineau, and R. D. Hjelm. Attraction-repulsion actor-critic for continuous control reinforcement learning. *arXiv preprint arXiv:1909.07543*, 2019.

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

[34] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[35] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56 (11):5847–5861, 2010.

[36] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.