Novel Policy Seeking with Constrained Optimization

Hao Sun[*] DAMTP, Cambridge
Zhenghao Peng CS, UCLA
Bo Dai Shanghai AI Lab
Dahua Lin IE, CUHK
Bolei Zhou CS, UCLA

Abstract

In problem-solving, we humans tend to come up with different novel solutions to the same problem. However, conventional reinforcement learning algorithms ignore such a feat and only aim at producing a set of monotonous policies that maximize the cumulative reward. The resulting policies usually lack diversity and novelty. In this work, we aim at enabling the learning algorithms with the capacity of solving the task with multiple solutions through a practical novel policy generation workflow that can generate a set of diverse and well-performing policies. Specifically, we begin by introducing a new metric to evaluate the difference between policies. On top of this well-defined novelty metric, we propose to rethink the novelty-seeking problem through the lens of constrained optimization, to address the dilemma between the task performance and the behavioral novelty in existing multi-objective optimization approaches, we then propose a practical novel policy-seeking algorithm, Interior Policy Differentiation (IPD), which is derived from the interior point method commonly known in the constrained optimization literature. Experimental comparisons on benchmark environments show IPD can achieve a substantial improvement over previous novelty-seeking methods in terms of both novelties of generated policies and their performances in the primal task. [2]

1 Introduction

In the sense of learning through interactions with the environment, the scheme of reinforcement learning (RL) is conceptually similar to the emergence of intelligence [1]: an agent explores and exploits information of a given environment, learns to master some certain skills through trials and errors to gain as much reward as possible. When solving a problem, we humans could be creative to come up with multiple different solutions and gain insights from searching for diverse solutions. e.g., a self-containing example is the various approaches in RL research.

While the state-of-the-art algorithms have achieved superhuman performance in a variety of challenging tasks [2–6], the task of encouraging individualized diversity [3] of learned agents, on the other hand, is relatively under-explored. Different from conventional RL agents that are only learned through interactions with the external environment, novel policy generation is a task considering the differentiation among individual policies. The differentiation among policies can be explained as the social influence [7–11] in social science literature. Although many works have been proposed applying social motivation to Multi-Agent Reinforcement Learning (MARL) settings [12–15], how to motivate a single RL agent to perform differently against existing agents is still an open question.

In previous attempts for novel policy generation, there are three main challenges: (1) heuristically defined metric for novelty estimation is computational expensive [16], (2) defining novelty reward for an entire episode yields additional challenge in credit assignment, and (3) solving the problem under

[*] hs789@cam.ac.uk

[2] Code is open-sourced at https://github.com/holarissun/NPSCO

[3] Note that this is different from diversity-driven exploration. While the latter focuses on the inner-diversity of a policy, in our work we focus on the inter-diversity between policies

Deep RL Workshop at 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

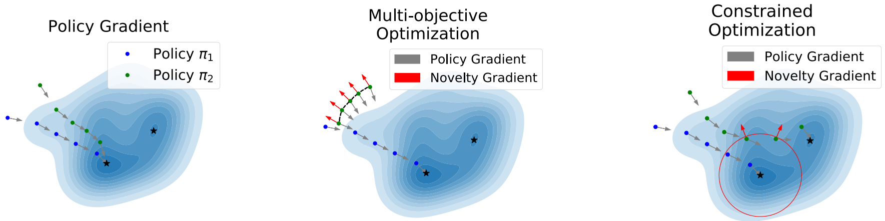

Figure 1: The comparison of the standard policy gradient method without novelty seeking (left), multi-objective optimization method (middle), and our constrained optimization approach (right) for generating novel policies. The standard policy gradient method does not try actively to find novel solutions. The multi-objective optimization method may impede the learning procedure when the novelty gradient is being applied all the time [17], e.g., a random initialized policy will be penalized from getting closer to the previous policy due to the conflict of gradients, which limits the learning efficiency and the final performance. On the contrary, the novelty gradient of our constrained optimization approach will only be considered within a certain region to keep the policy being optimized away from highly similar solutions. Such an approach is more flexible and includes the multi-objective optimization method as its special case.

the formulation of multi-objective optimization leads to the performance decay in the original task. Fig. 1 compares the policy gradients of three cases, namely the one without novel policy seeking, novelty-seeking with multi-objective optimization, and novelty-seeking with constrained optimization methods, respectively. In this work we take into consideration not only the novelty of a set of learned policies but also the performance of those novel policies in the primal task, when addressing the problem of novel-policy-generation.

Our contributions can be summarized as follows:

1. Mathematically, we introduce a lightweight metric to compute the difference between policies with *instant feedback* at every timestep, to address the first two drawbacks of previous novel policy seeking methods discussed above;

2. Practically, we propose a constrained optimization formulation for novel policy generation to avoid hindering the primal task performance while seeking cross-policy diversity. We further design an efficient *Novelty-Reward-Scale-Agnostic* algorithm dubbed as IPD, resembling the interior point method in constrained optimization literature;

3. Empirically, we evaluate IPD on several continuous control benchmarks to generate groups of diverse policies, showing the strengths of our constrained optimization solution for novelty-seeking can generate a series of diverse and well-performing policies, compared to previous multi-objective novel policy generation methods.

2 Related Work

Intrinsic motivation methods. In previous work, different approaches are proposed to provide intrinsic motivation or intrinsic reward as a supplementary to the primal task reward for better exploration [18–22]. All those approaches use the weighted sum of two rewards, the primal rewards provided by environments and the intrinsic rewards provided by different heuristics. On the other hand, the work of DIAYN and DADS [23, 24] learn diverse skills without extrinsic reward. Those approaches focus on decomposing diverse skills of a single policy, while our work focuses on learning diverse behaviors among a batch of policies for the same task.

Diverse policy generation methods. The work of Such et al. shows that different RL algorithms may converge to different policies for the same task [25]. On the contrary, we are interested in learning different policies through a single algorithm with the capability of avoiding local optimum. The work of Pugh et al. establishes a standard framework for understanding and comparing different approaches to search for quality diversity (QD) [26]. Conti et al. proposes a solution which avoids local optima as well as achieves higher performance by adding novelty search and QD to evolution strategies [27]. The Task-Novelty Bisector (TNB) [16] aims to solve novel policy generation problem by jointly optimize the extrinsic rewards and novelty rewards defined by an auto-encoder. In this work, we first adopt TNB in the constrained optimization framework, resulting in Contrained TNB, to demonstrate the dilemma between the task performance and novelty pursuance.

Constrained Markov Decision Process. The Constrained Markov Decision Process (CMDP) [28] considers the situation where an agent interacts with the environment under certain constraints. Formally, the CMDP can be defined as a tuple $(\mathcal{S}, \mathcal{A}, \gamma, r, c, C, P, s_0)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action space; $\gamma \in [0, 1)$ is a discount factor; $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and $c : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denote the reward function and cost function; $C \in \mathbb{R}^+$ is the upper bound of permitted expected cumulative cost; $P(\cdot|s, a) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the transition dynamics, and s_0 is the initial state. Denote the Markovian policy class as Π, where $\Pi = \{\pi : \mathcal{S} \times \mathcal{A} \to [0, 1], \sum_a \pi(a|\pi) = 1\}$ The learning objective of a policy for CMDP is to find a $\pi^* \in \Pi$, such that

$$\pi^* = \max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \pi, s' \sim P}[\sum_{t=0}^{\infty} \gamma^t r(s, a, s')], \text{ s.t. } \mathbb{E}_{\tau \sim \pi, s' \sim P}[\sum_{t=0}^{\infty} \gamma^t c(s, a, s')] \leq C, \tag{1}$$

where τ indicates a trajectory $(s_0, a_0, s_1, ...)$ and $\tau \sim \pi$ represents the distribution over trajectories following policy π: $a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t); t = 0, 1, 2, ...$ Previous literature provide several approaches to solve CMDP [29–32].

3 Methods

In Sec.3.1, we define a metric space that measures the difference between policies, which is the fundamental element for the proposed methods. In Sec.3.2, we develop a practical estimation method for this metric. Sec.3.3 describes the formulation of constrained optimization on novel policy generation. The implementations of two practical algorithms are further introduced in Sec.3.4.

We denote the policies as $\{\pi_{\theta_i}; \theta_i \in \Theta, i = 1, 2, ...\}$, wherein θ_i represents parameters of the i-th policy, Θ denotes the whole parameter space. In this work, we focus on improving the behavioral diversity of policies from PPO [33], thus we use Θ to represent Θ_{PPO} in this paper. It is worth noting that the proposed methods can be easily extended to other RL algorithms [34–37]. To simplify the notation, we omit π and denote a policy π_{θ_i} as θ_i unless stated otherwise.

3.1 Measuring the Difference between Policies

In this work, we use the Wasserstein metric W_p [38–40] to measure the distance between policies. Concretely, in this work we consider the Gaussian-parameterized policies, where the W_p over two policies can be written in the closed form $W_2^2(\mathcal{N}(m_1, \Sigma_1), \mathcal{N}(m_2, \Sigma_2)) = ||m_1 - m_2||^2 + \mathrm{tr}[\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}]$ as $p = 2$, where $m_1, \Sigma_1, m_2, \Sigma_2$ are mean and covariance metrics of the two normal distributions. In the following of this paper, we use D_W to denote the W_2 and it is worth noting that when the covariance matrics are identical, the trace term disappears and only the term involving the means remains, i.e., $D_W = |m_1 - m_2|$ for Dirac delta distributions located at points m_1 and m_2. This diversity metric satisfies the three properties of a metric, namely identity, symmetry as well as triangle inequality.

Proposition 1 (Metric Space $(\Theta, \overline{D}_W^q)$)**.** *The expectation of $D_W(\cdot, \cdot)$ of two policies over any state distribution $q(s)$:*

$$\overline{D}_W^q(\theta_i, \theta_j) := \mathbb{E}_{s \sim q(s)}[D_W(\theta_i(a|s), \theta_j(a|s))], \tag{2}$$

is a metric on Θ, thus $(\Theta, \overline{D}_W^q)$ is a metric space.

The proof of Proposition 1 is straightforward. It is worth mentioning that Jensen Shannon divergence D_{JS} or Total Variance Distance D_{TV} [41, 42, 34] can also be applied as alternative metric spaces, we choose D_W in our work for that the Wasserstein metric better preserves the continuity [40].

On top of the metric space $(\Theta, \overline{D}_W^q)$, we can then compute the novelty of a policy as follows.

Definition 1 (Novelty of Policy)**.** *Given a reference policy set Θ_{ref} such that $\Theta_{ref} = \{\theta_i^{ref}, i = 1, 2, ...\}, \Theta_{ref} \subset \Theta$, the novelty $\mathrm{U}(\theta|\Theta_{ref})$ of policy θ is the minimal difference between θ and all policies in the reference policy set, i.e.,*

$$\mathrm{U}(\theta|\Theta_{ref}) := \min_{\theta_j \in \Theta_{ref}} \overline{D}_W^q(\theta, \theta_j). \tag{3}$$

Consequently, to encourage the discovery of novel policies discovery, typical novel policy generation methods tend to directly maximize the novelty of a new policy, i.e., $\max_\theta \mathrm{U}(\theta|\Theta_{ref})$, where the Θ_{ref} includes all existing policies.

3.2 Estimation of $\overline{D}^q_W(\theta_i, \theta_j)$ and the Selection of $q(s)$

In practice, the calculation of $\overline{D}^q_W(\theta_i, \theta_j)$ is based on Monte Carlo estimation where we need to sample s from $q(s)$. Although in Eq.(2) $q(s)$ can be selected simply as a uniform distribution over the state space, there remains two obstacles: first, in a finite state space we can get precise estimation after establishing ergodicity, but problem arises when facing continuous state spaces due to the difficulty of efficiently obtaining enough samples; second, when s is sampled from a uniform distribution q, we can only get *sparse* episodic reward instead of *dense* online reward which is more useful in learning. Therefore, we make an approximation here based on importance sampling.

Formally, we denote the domain of $q(s)$ as $\mathcal{S}_q \subset \mathcal{S}$ and assume $q(s)$ to be a uniform distribution over $\mathcal{S}_q$, without loss of generality in later analysis. Notice $\mathcal{S}_q$ is closely related to the algorithm being used in generating trajectories [43]. As we only care about the reachable regions of a certain algorithm (in this work, PPO), the domain $\mathcal{S}_q$ can be decomposed by $\mathcal{S}_q = \lim_{N\to\infty} \bigcup_{i=1}^N \mathcal{S}_{\theta_i}$, where $\mathcal{S}_{\theta_i}$ denotes all the possible states a policy θ_i can visit given a starting state distribution.

In order to get online-reward, we estimate Eq.(2) with

$$\overline{D}^q_W(\theta_i, \theta_j) = \mathbb{E}_{s\sim q(s)}[D_W(\theta_i(a|s), \theta_j(a|s))] = \mathbb{E}_{s\sim \rho_{\theta_i}(s)}[\frac{q(s)}{\rho_{\theta_i}(s)} D_W(\theta_i(a|s), \theta_j(a|s))], \tag{4}$$

where we use $\rho_\theta(s)$ to denote the stationary state visitation frequency under policy θ, i.e., $\rho_\theta(s) = P(s_0 = s|\theta) + P(s_1 = s|\theta) + ... + P(s_T = s|\theta)$ in finite horizon problems. We propose to use the averaged stationary visitation frequency as $q(s)$, e.g., for PPO, $q(s) = \overline{\rho}(s) = \mathbb{E}_{\theta\sim\Theta_{\mathrm{PPO}}}[\rho_\theta(s)]$. Clearly, choosing $q(s) = \overline{\rho}(s)$ will be much better than choosing a uniform distribution as the importance weight will be closer to 1. Such an importance sampling process requires a necessary condition that $\rho_{\theta_i}(s)$ and $q(s)$ have the same domain, which can be guaranteed by applying a sufficient exploration noise on θ.

Another difficulty lies in the estimation of $\overline{\rho}(s)$, which is always intractable given a limited number of trajectories. However, during training, θ_i is a policy to be optimized and $\theta_j \in \Theta_{ref}$ is a fixed reference policy. The error introduced by approximating the importance weight as 1 will get larger when θ_i becomes more distinct from normal policies, at least in terms of the state visitation frequency. We may just regard increasing of the approximation error as the discovery of novel policies.

Proposition 2 (Unbiased Single Trajectory Estimation)**.** *The estimation of $\rho_\theta(s)$ using a single trajectory τ is unbiased.*

The Proposition 2 follows the usual trick in RL that uses a single trajectory to estimate the stationary state visitation frequency. Given the definition of novelty and a practically unbiased sampling method, the next step is to develop an efficient learning algorithm.

3.3 Constrained Optimization Formulation for Novel Policy Generation

In the traditional RL paradigm, maximizing the expectation of cumulative rewards is commonly used as the objective. i.e., $\max_{\theta\in\Theta} \mathbb{E}_{\tau\sim\theta}[g]$, where $g = \sum_{t=0} \gamma^t r_t$ and $\tau \sim \theta$ denotes a trajectory τ sampled from the policy θ.

To improve the diversity of different agents' behaviors, the learning objective must take both the reward from the primal task and the policy novelty into consideration. Previous approaches [18–22] often directly use the weighted sum of these two terms as the objective:

$$\max_{\theta\in\Theta} \mathbb{E}_{\tau\sim\theta}[g_{\mathrm{total}}] = \max_{\theta\in\Theta} \mathbb{E}_{\tau\sim\theta}[\alpha \cdot g_{\mathrm{task}} + (1-\alpha) \cdot g_{\mathrm{int}}], \tag{5}$$

where $0 < \alpha < 1$ is a weight hyper-parameter, g_{task} is the reward from the primary task, and $g_{\mathrm{int}} = \sum_{t=0} \gamma^t r_{\mathrm{int},t}$ is the cumulative intrinsic reward of the *intrinsic reward* $r_{\mathrm{int},t}$. In our case, the intrinsic reward is the novelty reward $r_{\mathrm{int}} = \min_{\theta_j\in\Theta_{ref}} \overline{D}^{\overline{\rho}}_W(\theta, \theta_j)$. These methods can be summarized as Weighted Sum Reward (WSR) methods [17]. Such an objective is sensitive to the

Algorithm 1 IPD

```
Input: (1) a behavior policy $\theta_{old}$, (2) a set of previous policies $\{\theta_j\}_{j=1}^M$, (3) a novelty metric $U(\theta, \{\theta_j\}|\rho) = U(\theta, \{\theta_j\}|\tau) = \min_{\theta_j} \overline{D}_W^{\tau}(\theta, \theta_j)$, (4) a novelty threshold $r_0$ and starting point $t_S$.
Initialize $\theta_{old}$.
for iteration $= 1, 2, ...$ do
    for t $= 1, 2, ..., T$ do
        Step the environment by taking action $a_t \sim \theta_{\text{old}}$ and collect transitions.
        if $U(\theta_{old}, \{\theta_j\}|\tau) - r_0 < 0$ AND $t > t_S$ then
            Break this episode.
        end if
    end for
    Update policy parameters based on sampled data.
end for
```

Algorithm 2 Constrained TNB

```
Input: (1-4) same as IPD, (5) a value network for cost $V_c$.
Initialize $\theta_{old}$.
for iteration $= 1, 2, ...$ do
    for t $= 1, 2, ..., T$ do
        Step the environment by taking action $a_t \sim \theta_{\text{old}}$ and collect transitions.
    end for
    Compute advantage of reward $\hat{A}_{r,1}, ..., \hat{A}_{r,T}$.
    Compute advantage of cost $\hat{A}_{c,1}, ..., \hat{A}_{c,T}$.
    Optimize objective for reward $\mathcal{L}_r^{\text{CLIP}}$, with gradient $g_r = \nabla_\theta \mathcal{L}_r^{\text{CLIP}}$.
    Optimize objective for cost $\mathcal{L}_c^{\text{CLIP}}$, with gradient $g_c = -\nabla_\theta \mathcal{L}_c^{\text{CLIP}}$
    if $U(\theta_{old}, \{\theta_j\}|\tau) - r_0 < 0$ then
        Calculate $\vec{p}$ according to Eq.(7) with $g_r$ and $g_c$
    else
        Calculate $\vec{p}$ with $g_r$
    end if
    Update policy parameters
end for
```

selection of α as well as the formulation of r_{int}. For example, in our case formulating the novelty reward r_{int} as $\min_{\theta_j} \overline{D}_W^{\rho}(\theta, \theta_j)$, $\exp\left[\min_{\theta_j} \overline{D}_W^{\rho}(\theta, \theta_j)\right]$ and $-\exp\left[-\min_{\theta_j} \overline{D}_W^{\rho}(\theta, \theta_j)\right]$ will lead to significantly different results as they determine the trade-offs in the two terms given α. Besides, dilemma also arises in the selection of α: while a large α may undermine the contribution of intrinsic reward, a small α could ignore the importance of the primal task, leading to the failure of an agent in solving the task.

The crux of tackling such an issue is to deal with the conflict between different objectives. The work of Zhang et al. proposes the TNB, where the task reward is regarded as the dominant one while the novelty reward is regarded as subordinate [17]. However, as TNB considers the novelty gradient all the time, it may hinder the learning process. Intuitively, well-performing policies should be more similar to each other than to random initialized policies. As a new random initialized policy is different enough from previous policies, considering the novelty gradient at beginning of training will result in a much slower learning process.

In order to tackle the above problems and adjust the extent of novelty in new policies, we propose to solve the novelty-seeking problem under the perspective of constrained optimization. The intuition is as follows: while the task reward is considered as a learning objective, the novelty reward should be considered as a bonus instead of another objective, thus should not impede the learning of the primal task. Fig. 1 illustrates how novelty gradients impede the learning of a policy: at the beginning of learning, a random initialized policy should learn to be more similar to a well-performing policy rather than be different. The seeking of novelty should not be taken into consideration all the time during learning. With such an insight, we update the multi-objective optimization problem in Eq.(5) into a constrained optimization problem as:

$$\max_{\theta \in \Theta} f(\theta) = \mathbb{E}_{\tau \sim \theta}[g_{\text{task}}] \quad \text{s.t. } g_t(\theta) = \overline{r}_{\text{int},t} - r_0 \geq 0, \forall t = 1, 2, ..., T, \tag{6}$$

where r_0 is a threshold indicating minimal permitted novelty, and $\overline{r}_{\text{int},t}$ denotes a moving average of $r_{\text{int},t}$. as we need not force every single action of a new agent to be different from others. Instead, we care more about the long-term differences. Therefore, we use cumulative novelty terms as constraints. Moreover, the constraints can be flexibly applied after the first t_S timesteps (e.g., $t_S = 20$) for the consideration of similar starting sequences, so that the constraints can be written as $g_t(\theta) \geq 0, \forall t = t_S, ..., T$.

3.4 Practical Novel Policy Generation Methods

One thing to note is that WSR and TNB proposed in the prior work [16] correspond to different approaches in constrained optimization problems, yet some important ingredients are missing. We in this section adopt TNB according to the Feasible Direction Method in constrained optimization

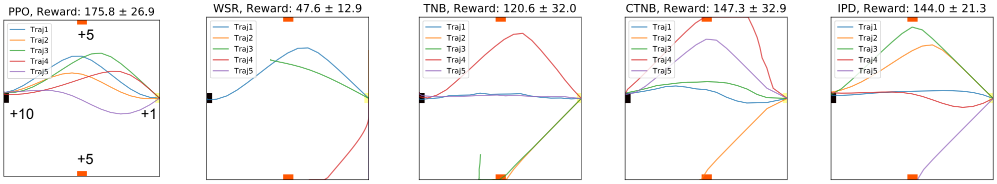

Figure 2: Experimental results on the Four Reward Maze Problem. We generate 5 policies with different novel policy generation methods, and use the PPO with different random seeds as baseline. In each figure, the 5 lines indicate 5 trajectories when the game is started from the right hand side. It worth noting that the results of WSR, CTNB and IPD are associated with the parameters of weights or threshold. We set the weight parameter in WSR as 10 to make the two reward terms comparable, and set the thresholds in CTNB and IPD as the averaged novelty between policies trained with PPO. All policies are trained with 6.1×10^3 episodes.

and then propose our method of Interior Policy Differentiation (IPD), according to the Interior Point Method in constrained optimization literature. A detailed discussion on WSR is provided in Appendix C.1.

TNB: Feasible Direction Method The Feasible Direction Method (FDM) [44, 45] solves the constrained optimization problem by finding a direction $\vec{p}$ where taking gradient upon will lead to the increment of the objective function as well as constraints satisfaction, i.e., $\nabla_\theta f^{\mathrm{T}} \cdot \vec{p} > 0$, if $g > 0$ and $\nabla_\theta g^{\mathrm{T}} \cdot \vec{p} > 0$ otherwise. The TNB proposes to use a revised bisector of gradients $\nabla_\theta f$ and $\nabla_\theta g$ as $\vec{p}$,

$$\vec{p} = \begin{cases} \nabla_\theta f + \frac{|\nabla_\theta f|}{|\nabla_\theta g|} \nabla_\theta g, & \text{if } \cos(\nabla_\theta f, \nabla_\theta g) > 0 \\ \nabla_\theta f + \frac{|\nabla_\theta f|}{|\nabla_\theta g|} \nabla_\theta g \cdot \cos(\nabla_\theta f, \nabla_\theta g), & \text{otherwise} \end{cases} \tag{7}$$

Clearly, Eq.(7) satisfies the constraints but it is more strict than it as the $\nabla_\theta g$ term always exists during the optimization of TNB. Based on TNB, we provide a revised approach, named Constrained Task Novel Bisector (CTNB), which resembles better with FDM. Specifically, when $g > 0$, CTNB will not apply $\nabla_\theta g$ on g. It is clear that TNB is a special case of CTNB when the novelty threshold r_0 is set to infinity. We note that in both TNB and CTNB, the learning stride is fixed to be $\frac{|\nabla_\theta f| + |\nabla_\theta g|}{2}$ and may lead to problem when $\nabla_\theta f \to 0$, where the final optimization result will rely heavily on the selection of g, i.e., the shape of g is crucial for the success of this approach. We propose CTNB in our work, as a constrained optimization vairant of TNB, to demonstrate the importance of the constrained optimization perspective in novelty seeking, however, we do not in practice observe such a method achieves satisfactory performance.

IPD: Interior Point Method The Interior Point Method [46, 47] is another approach used to solve the constrained optimization problem. Thus here we solve Eq.(6) using the Interior Policy Differentiation (IPD), which can be regarded as an analogy of the Interior Point Method. In the vanilla Interior Point Method, the constrained optimization problem in Eq.(6) is solved by reforming it to an unconstrained form with an additional barrier term $-\alpha \log g(\theta)$ in the objective as $\max_{\theta \in \Theta} f(\theta) - \alpha \log g(\theta)$, or more precisely in our problem with the formulation with Eq.(6) we have $\max_{\theta \in \Theta} \mathbb{E}_{\tau \sim \theta}[g_{\text{task}} - \sum_{t=0}^{T} \alpha \log(\overline{r}_{\text{int},t} - r_0)]$, where $\alpha > 0$ is the barrier factor. Besides the log barrier term, there are other choices like $\alpha \frac{1}{g(\theta)}$ can be used and the objective becomes $\max_{\theta \in \Theta} \ f(\theta) + \alpha \frac{1}{g(\theta)}$. As α is small, the barrier term will introduce only minuscule influence on the objective. On the other hand, when θ get closer to the barrier, the objective will increase rapidly. The limits when $\alpha \to 0$ then lead to the solution of Eq.(6). The convergence of such methods are provided in previous works [48, 49].

However, directly applying IPM is computationally expensive and numerically unstable. In this work, we propose a simple yet novel heuristic method that resembles the idea of barrier methods: we implicitly apply such barrier terms by providing termination signals in interactions with the environments. Our method can be regarded as revising the primal task MDP into a new one in which the behaviors of agents must satisfy novelty constraints. Specifically, in the RL paradigm, the learning procedure of an agent is determined by the experiences collected during interactions with the environment and the sampling strategy used to filter experiences in the calculation of policy gradients.

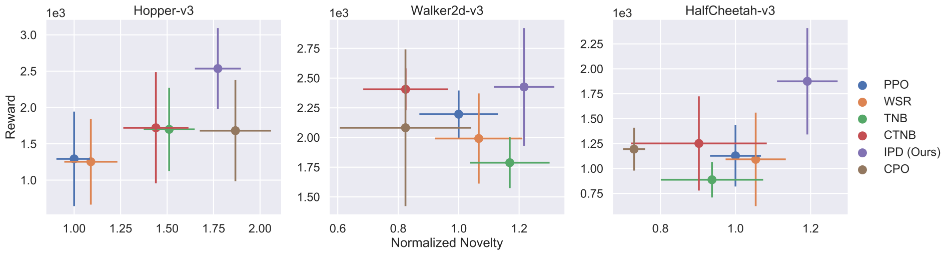

Figure 3: The performance and novelty comparison of different methods in Hopper-v3, Walker2d-v3 and HalfCheetah-v3 environments. The value of novelty is normalized to relative novelty by regarding the averaged novelty of PPO policies as the baseline. The results are from 10 policies of each method, with the points showing their mean and lines showing their standard deviation.

Since the learning process is based on sampled transitions, a more natural way can thus be used to perform the constrained optimization. We can simply bound the collected transitions in the feasible region by permitting previously trained M policies $\theta_i \in \Theta_{\text{ref}}, i = 1, 2, ..., M$ sending termination signals during the training process of new agents. In other words, we implicitly bound the feasible region by terminating any new agent that steps outside it.

Consequently, during the training process, all valid samples we collected are inside the feasible region, which means these samples are less likely to appear in previously trained policies. At the end of the training, we obtain a new policy that has sufficient novelty. In this way, we no longer need to consider the trade-off between intrinsic and extrinsic rewards deliberately. The learning process of IPD is thus more robust and no longer suffers from the objective inconsistency.

Remark 1 (Reward-Shaping-Agnostic Novelty Seeking)**.** *IPD is a gradient-free method with regard to the novelty reward.*

Remark 1 is an important property that only IPD owns. For other approaches, including both multi-objective approaches and constrained optimization approaches, an elaborated design of the novelty reward function is needed, e.g., it can be any monotonic increasing function of $\overline{D}_W^{\rho}$. While it is well-known that reward shaping [50–52] is in general a non-trivial work that requires domain knowledge [53, 2–4, 6], such an additional novelty-reward term further increases the burden of proper reward design.

Differently, the seeking of novelty in IPD does not require any (policy) gradient information that flows from the novelty reward, therefore, the selection of reward scaling function is agnostic to the performance of IPD. IPD learns to become novel in a passive manner, i.e., an episode will be terminated whenever the averaged step-wise novelty is lower than a given threshold. Searching in a constant hyper-parameter space is at least tractable while searching in a monotonically increasing functional reward shaping class [16] is not – let alone IPD works well with a default novelty threshold parameter $r_0 =$ averaged differences between PPO policies.

4 Experiments

According to Proposition 2, the novelty reward r_{int} in Eq.(6) under our novelty metric can be unbiasedly approximated by $r_{\text{int}} = \min_{\theta_j \in \Theta_{ref}} \overline{D}_W^{\rho_\theta}(\theta(a|s_t), \theta_j(a_j|s_t))$. We thus utilize this novelty metric directly throughout our experiments. We apply different novel policy generation methods, namely WSR, TNB, CTNB, and IPD, to the backbone RL algorithm PPO [33]. The extension to other popular RL algorithms is straightforward. More implementation details are depicted in Appendix C.

Experiments in the work of [43] show that one can simply change the random seeds before training to get policies that perform differently. Therefore, we use PPO with varying random seeds as a baseline method for novel policy generation and use the averaged differences between policies learned by this baseline as the **default threshold** r_0 in CTNB and IPD. Algorithm 1 and Algorithm 2 show the pseudo code of IPD and CTNB based on PPO, where the blue lines show the additional code added to the standard PPO. Qualitative results can be found in Appendix D.

Table 1: Reward and Success Rate of 10 Policies. IPD beat CTNB, CPO, TNB and WSR in all three environments. Constrained optimization approaches outperforms multi-objective methods.

	Reward			Success Rate		
Environment	Hopper	Walker2d	HalfCheetah	Hopper	Walker2d	HalfCheetah
PPO	1292 ± 650	2196 ± 200	1127 ± 308	0.5	0.5	0.5
WSR	1253 ± 591	1992 ± 380	1091 ± 469	0.6	0.3	0.3
TNB	1699 ± 573	1788 ± 214	887 ± 178	0.8	0.0	0.1
CPO	1681 ± 696	2082 ± 660	1194 ± 215	0.8	0.6	0.8
CTNB	1721 ± 765	$\mathbf{2405 \pm 177}$	1251 ± 473	0.8	**0.9**	0.5
IPD (Ours)	$\mathbf{2536 \pm 557}$	2282 ± 206	$\mathbf{1875 \pm 533}$	**1.0**	0.6	**0.9**

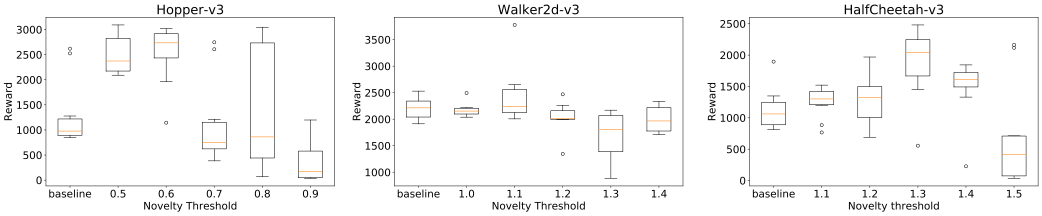

Figure 4: The performance under different novelty thresholds in the Hopper, Walker and HalfCheetah environments. The results are collected from 10 learned policies based on PPO. The box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to show the range of the data. Flier points are those past the end of the whiskers.

4.1 The Four Reward Maze Problem

We first utilize a basic 2-D environment named Four Reward Maze as a diagnostic environment where we can visualize learned policies directly. In this environment, four positive rewards of different values (e.g., $+5, +5, +10, +1$ for top, down, left and right respectively) are assigned to four middle points with radius 1 on each edge in a 2-D $N \times N$ square map. We use $N = 16$ in our experiments. The observation of a policy is the current position and the agent will receive a negative reward of -0.01 at each timestep except stepping into the reward regions. Each episode starts from a randomly initialized position and the action space is limited to $[-1, 1]$. The performance of each agent is evaluated by the averaged performances over 100 trials.

Results are shown in Fig. 2, where the behaviors of the PPO agents are quite similar, suggesting the diversity provided by random seeds is limited. WSR and TNB solve the novel policy generation problem from the multi-objective optimization formulation, they thus suffer from the unbalance between performance and novelty. While WSR and TNB both provide sufficient novelty, performances of agents learned by WSR decay significantly, so did TNB due to an encumbered learning process, as we analyzed in Sec.3.3. Both CTNB and IPD, solving the task with novelty-seeking from the constrained optimization formulation, provide evident behavior diversity and perform recognizably better than TNB and WSR.

4.2 The MuJoCo Benchmark

We evaluate our proposed method on three locomotion tasks [54, 55]: the Hopper-v3 (11 observations and 3 actions), Walker2d-v3 (11 observations and 6 actions), and HalfCheetah-v3 (17 observations and 6 actions). Although relaxing the healthy termination thresholds in Hopper and Walker may permit more visible behavior diversity, all the environment parameters are set as default values in our experiments to demonstrate the generality of our method.

Comparison on Novelty and Performance We implement WSR, TNB, CTNB, and IPD using the same hyper-parameter settings per environment. And we also apply CPO [29] as a baseline as a solution of CMDP. For each method, we first train 10 policies using PPO with different random seeds. Those PPO policies are used as the primal reference policies, and then we train 10 novel policies that try to be different from previous reference policies. Concretely, in each method, the $1st$ novel policy is trained to be different from the previous 10 PPO policies, and the $2nd$ should be different from the previous 11 policies, and so on. More implementation details are depicted in Appendix C.

Fig. 3 shows our experimental results in terms of novelty (the x-axis) and the performance (the y-axis). Policies close to the upper right corner are the more novel ones with higher performance. In all environments, the performance of CTNB, IPD and CPO outperforms WSR and TNB, showing the advantage of constrained optimization approaches in novel policy generation. Specifically, the results of CTNB are all better than their multi-objective counterparts from TNB, showing the superiority of generating novel policies with constrained optimization. In all experiments we use a linear novelty reward function, i.e., $r_{\text{int}} = \min_{\theta_j} \overline{D}_W^{\rho}(\theta, \theta_j)$. We attribute the failure of CPO, TNB and CTNB in Walker and HalfCheetah in finding novel policies to that their convergence behavior is fully controlled by the reward scaling function. Whereas in IPD, there is no novelty-gradient controlled by such a scaling function.

Comparisons of the task-related rewards are carried out in Table 1, where among all the four methods, IPD provides sufficient diversity with minimum loss of performance. Instead of performance decay, we find IPD is able to find better policies in the environment of Hopper and HalfCheetah. Moreover, in the Hopper environment, while the agents trained with PPO tend to fall into the same local minimum. (e.g., they all jump as far as possible and then terminate this episode. On the contrary, PPO with IPD keeps new agents away from falling into the same local minimum, because once an agent has reached some local minimum, agents learned later will try to avoid this region due to the novelty constraints. Such property shows that IPD can enhance the traditional RL schemes to tackle the local exploration challenge [56, 57]. A similar feature brings about reward growth in the environment of HalfCheetah.

Success Rate of Each Method In addition to averaged reward, we also use the success rate as another metric to compare the performance of different approaches. Roughly speaking, the success rate evaluates the stability of each method in terms of generating a policy that performs as good as the policies PPO generates. In this work, we regard a policy successful when its performance achieves at least as good as the median performance of policies trained with PPO. To be specific, we use the median of the final performance of PPO as the baseline, and if a novel policy, which aims at performing differently to solve the same task, surpasses the baseline during its training process, it will be regarded as a successful policy. By definition, the success rate of PPO is 0.5 as a baseline for every environment. Table 1 shows the success rate of all the methods. The results show that all constrained novel policy generation methods (CTNB, IPD, CPO) can surpass the average baseline during training, while the multi-objective optimization approaches normally can not.

4.3 Novel Policy Generation without Performance Decay

Multi-objective formulation of novel policy generation has the risk of sacrificing the primal performance as the overall objective needs to consider both novelty and primal task rewards. On the contrary, under the perspective of constrained optimization, there will be no more trade-off between novelty and final reward as the only objective is the task reward. Given a certain novelty threshold, the algorithms tend to find the optimal solution in terms of task reward under constraints, thus the learning process becomes more controllable and reliable, i.e., one can utilize the novelty threshold to control the degree of novelty. The proper magnitude of the novelty threshold leads to more exploration among a population of policies, thus the performance of latterly found policies may be better than or at least as good as those trained without novelty seeking. However, when a larger magnitude of novelty threshold is applied, the performance of found novel policies will decrease because finding a feasible solution will get harder under more strict constraints. Fig. 4 shows experimental results on adjusting the thresholds, which supports our intuition.

5 Conclusion

In this work, we rethink the novel policy seeking problem under the perspective of constrained optimization. We first introduce a new metric to measure the distances between policies, on top of which we define the novelty of a policy. Based on our formulation of constrained optimization, we provide practical algorithms for constrained novel policy learning, we evaluate several constrained policy optimization methods: namely the CPO, Constrained TNB, and the Interior Policy Differentiation (IPD) proposed in this work. Our experimental results demonstrate IPD, as a novelty (policy) gradient-free approach, can effectively learn various well-performing yet diverse policies, outperforming previous multi-objective methods, as well as constrained optimization baselines.

References

[1] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*, vol. 2. MIT press Cambridge, 1998.

[2] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, “Grandmaster level in starcraft ii using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[3] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, *et al.*, “Solving rubik’s cube with a robot hand,” *arXiv preprint arXiv:1910.07113*, 2019.

[4] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, *et al.*, “Dota 2 with large scale deep reinforcement learning,” *arXiv preprint arXiv:1912.06680*, 2019.

[5] A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, Z. D. Guo, and C. Blundell, “Agent57: Outperforming the atari human benchmark,” in *International Conference on Machine Learning*, pp. 507–517, PMLR, 2020.

[6] M. Elbarbari, K. Efthymiadis, B. Vanderborght, and A. Nowé, “Ltlf-based reward shaping for reinforcement learning,” in *Adaptive and Learning Agents Workshop 2021*, 2021.

[7] B. Rogoff, *Apprenticeship in thinking: Cognitive development in social context.* Oxford university press, 1990.

[8] R. M. Ryan and E. L. Deci, “Intrinsic and extrinsic motivations: Classic definitions and new directions,” *Contemporary educational psychology*, vol. 25, no. 1, pp. 54–67, 2000.

[9] C. P. van Schaik and J. M. Burkart, “Social learning and evolution: the cultural intelligence hypothesis,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1567, pp. 1008–1016, 2011.

[10] J. Henrich, *The secret of our success: How culture is driving human evolution, domesticating our species, and making us smarter.* Princeton University Press, 2017.

[11] Y. N. Harari, *Sapiens: A brief history of humankind.* Random House, 2014.

[12] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas, “Social influence as intrinsic motivation for multi-agent deep reinforcement learning,” in *International Conference on Machine Learning*, pp. 3040–3049, 2019.

[13] E. Hughes, J. Z. Leibo, M. Phillips, K. Tuyls, E. Dueñez-Guzman, A. G. Castañeda, I. Dunning, T. Zhu, K. McKee, R. Koster, *et al.*, “Inequity aversion improves cooperation in intertemporal social dilemmas,” in *Advances in neural information processing systems*, pp. 3326–3336, 2018.

[14] P. Sequeira, F. S. Melo, R. Prada, and A. Paiva, “Emerging social awareness: Exploring intrinsic motivation in multiagent learning,” in *2011 IEEE International Conference on Development and Learning (ICDL)*, vol. 2, pp. 1–6, IEEE, 2011.

[15] A. Peysakhovich and A. Lerer, “Consequentialist conditional cooperation in social dilemmas with imperfect information,” *arXiv preprint arXiv:1710.06975*, 2017.

[16] Y. Zhang, W. Yu, and G. Turk, “Learning novel policies for tasks,” in *International Conference on Machine Learning*, pp. 7483–7492, PMLR, 2019.

[17] Y. Zhang, W. Yu, and G. Turk, “Learning novel policies for tasks,” *CoRR*, vol. abs/1905.05252, 2019.

[18] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, “Variational information maximizing exploration,” 2016.

[19] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

[20] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," *arXiv preprint arXiv:1808.04355*, 2018.

[21] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.

[22] H. Liu, A. Trott, R. Socher, and C. Xiong, "Competitive experience replay," *CoRR*, vol. abs/1902.00528, 2019.

[23] 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.

[24] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman, "Dynamics-aware unsupervised discovery of skills," *arXiv preprint arXiv:1907.01657*, 2019.

[25] F. P. Such, V. Madhavan, R. Liu, R. Wang, P. S. Castro, Y. Li, L. Schubert, M. Bellemare, J. Clune, and J. Lehman, "An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents," *arXiv preprint arXiv:1812.07069*, 2018.

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

[27] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. Stanley, and J. Clune, "Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents," in *Advances in Neural Information Processing Systems*, pp. 5027–5038, 2018.

[28] E. Altman, *Constrained Markov decision processes*, vol. 7. CRC Press, 1999.

[29] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 22–31, JMLR. org, 2017.

[30] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," in *Advances in neural information processing systems*, pp. 8092–8101, 2018.

[31] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *openai*, 2019.

[32] H. Sun, Z. Xu, M. Fang, Z. Peng, J. Guo, B. Dai, and B. Zhou, "Safe exploration by solving early terminated mdp," *arXiv preprint arXiv:2107.04200*, 2021.

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

[34] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, 2015.

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

[36] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv preprint arXiv:1802.09477*, 2018.

[37] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[38] L. Rüschendorf, "The wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, vol. 70, no. 1, pp. 117–129, 1985.

[39] C. Villani, *Optimal transport: old and new*, vol. 338. Springer Science & Business Media, 2008.

[40] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 214–223, PMLR, 06–11 Aug 2017.

[41] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions," *IEEE Transactions on Information theory*, 2003.

[42] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, p. 31, IEEE, 2004.

[43] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[44] A. Ruszczyński, "Feasible direction methods for stochastic programming problems," *Mathematical Programming*, vol. 19, no. 1, pp. 220–229, 1980.

[45] J. Herskovits, "Feasible direction interior-point technique for nonlinear optimization," *Journal of optimization theory and applications*, vol. 99, no. 1, pp. 121–146, 1998.

[46] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 281–302, 2000.

[47] G. B. Dantzig and M. N. Thapa, *Linear programming 2: theory and extensions*. Springer Science & Business Media, 2006.

[48] A. Conn, N. Gould, and P. Toint, "A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds," *Mathematics of Computation of the American Mathematical Society*, vol. 66, no. 217, pp. 261–288, 1997.

[49] S. J. Wright, "On the convergence of the newton/log-barrier method," *Mathematical Programming*, vol. 90, no. 1, pp. 71–100, 2001.

[50] J. Randløv and P. Alstrøm, "Learning to drive a bicycle using reinforcement learning and shaping.," in *ICML*, vol. 98, pp. 463–471, Citeseer, 1998.

[51] A. D. Laud, *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[52] H. Sun, L. Han, R. Yang, X. Ma, J. Guo, and B. Zhou, "Exploiting reward shifting in value-based deep rl," *arXiv preprint arXiv:2209.07288*, 2022.

[53] H. Sun, Z. Li, X. Liu, B. Zhou, and D. Lin, "Policy continuation with hindsight inverse dynamics," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[54] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[55] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control.," in *IROS*, pp. 5026–5033, IEEE, 2012.

[56] C. Tessler, G. Tennenholtz, and S. Mannor, "Distributional policy optimization: An alternative approach for continuous control," *arXiv preprint arXiv:1905.09855*, 2019.

[57] K. Ciosek, Q. Vuong, R. Loftin, and K. Hofmann, "Better exploration with optimistic actor critic," in *Advances in Neural Information Processing Systems*, pp. 1785–1796, 2019.

A Missing Proofs

A.1 Proof of Proposition 1

Definition 2. *A metric space is an ordered pair (M, d) where M is a set and d is a metric on M, i.e., a function $d\colon M \times M \to \mathbb{R}$ such that for any $x, y, z \in M$, the following holds:*
1. $d(x, y) \geq 0, d(x, y) = 0 \Leftrightarrow x = y,$
2. $d(x, y) = d(y, x),$
3. $d(x, z) \leq d(x, y) + d(y, z).$

The first two properties are obviously guaranteed by $\overline{D}_W^{\rho}$. As for the triangle inequality,

$$\begin{aligned}
&\mathbb{E}_{s\sim\rho(s)}[D_W(\theta_i(s), \theta_k(s)]\\
&= \mathbb{E}_{s\sim\rho(s)}[\sum_{l=1}^{|\mathcal{A}|} |\theta_i(s) - \theta_k(s)|]\\
&= \mathbb{E}_{s\sim\rho(s)}[\sum_{l=1}^{|\mathcal{A}|} |\theta_i(s) - \theta_j(s) + \theta_j(s) - \theta_k(s)|]\\
&\leq \mathbb{E}_{s\sim\rho(s)}[\sum_{l=1}^{(|\mathcal{A}|} |\theta_i(s) - \theta_j(s)| + |\theta_j(s) - \theta_k(s)|)]\\
&= \mathbb{E}_{s\sim\rho(s)}[\sum_{l=1}^{|\mathcal{A}|} |\theta_i(s) - \theta_j(s)|] + \mathbb{E}_{s\sim\rho(s)}[\sum_{l=1}^{|\mathcal{A}|} |\theta_j(s) - \theta_k(s)|]\\
&= \mathbb{E}_{s\sim\rho(s)}[D_W(\theta_i(s), \theta_j(s)] + \mathbb{E}_{s\sim\rho(s)}[D_W(\theta_j(s), \theta_k(s)]
\end{aligned}$$

B Proof of Proposition 2

$$\begin{aligned}
\rho_\theta(s) &= P(s_0 = s|\theta) + P(s_1 = s|\theta) + ... + P(s_T = s|\theta)\\
&\overset{L.L.N.}{=} \lim_{N\to\infty} \frac{\sum_{i=1}^{N} I(s_0 = s|\tau_i)}{N} + \frac{\sum_{i=1}^{N} I(s_1 = s|\tau_i)}{N} + ... + \frac{\sum_{i=1}^{N} I(s_T = s|\tau_i)}{N}\\
&= \lim_{N\to\infty} \frac{\sum_{j=0}^{T}\sum_{i=1}^{N} I(s_j = s|\tau_i)}{N}\\
\overline{\rho}_\theta(s) &= \sum_{i=1}^{N}\sum_{j=0}^{T} \frac{I(s_j = s|\tau_i)}{N}\\
\mathbb{E}[\overline{\rho}_\theta(s) &- \rho_\theta(s)] = 0
\end{aligned}$$

C Implementation Details

C.1 More Details on WSR

WSR: Penalty Method The Penalty Method considers the constraints of Eq.(6) by putting constraint $g(\theta)$ into a penalty term, followed by solving the following unconstrained problem in an iterative manner,

$$\max_{\theta\in\Theta} \quad f(\theta) + \frac{1-\alpha}{\alpha}\min\{g(\theta), 0\}, \tag{8}$$

The limit of the above unconstrained problem when $\alpha \to 0$ then leads to the solution of the original constrained problem. As an approximation, WSR chooses a fixed weight α, and uses the gradient of $\nabla_\theta f + \frac{1-\alpha}{\alpha}\nabla_\theta g$ instead of $\nabla_\theta f + \frac{1-\alpha}{\alpha}\nabla_\theta \min\{g(\theta), 0\}$, thus the final solution will intensely rely on the selection of α.

C.2 Calculation of D_W

We use deterministic part of policies in the calculation of D_W, i.e., we remove the Gaussian noise on the action space in PPO and use $D_W(a_1, a_2) = |a_1 - a_2|$.

C.3 Network Structure

We use MLP with 2 hidden layers as our actor models in PPO. The first hidden layer is fixed to have 32 units. We choose to use 10, 64 and 256 hidden units for the three tasks respectively in all of the main experiments, after taking the success rate, performance and computation expense (i.e. the preference to use less unit when the other two factors are similar) into consideration.

C.4 Training Timesteps

We fix the training timesteps in our experiments. The timesteps are fixed to be 1M in Hopper-v3, 1.6M for Walker2d-v3 and 3M for HalfCheetah-v3.

D Visualize Diversity

D.1 Mujoco Locomotion

In this section, we provide some qualitative results of IPD on the Mujoco locomotion tasks. In all of our experiments we use the vanilla Mujoco locomotion benchmarks, with the default settings on defining healthy states. Although otherwise the visualization of learned policies might become more diverse (e.g., a Hopper agent may learn to stand-up after falling down while another agent may learn to move forward on the ground if we set the z-axis healthy threshold as 0).

With the method of IPD, the Hopper policies (Figure 5) learns to jump further and avoids falling down rather instead of just jumping and falling down (Figure 6). In the Walker2d environment, the color of purple indicates the left leg is visible. It can be seen that the IPD policies (Figure 7) learn to use both left and right legs in walking, while the PPO policies usually learn jumping. (Figure 8). In HalfCheetah, the IPD policies (Figure 9) perform much better than the PPO policies (Figure 10). The IPD policies leran to run with head-downward (Figure 9 line 1), head-upward (Figure 9 line 3), and forward (Figure 9 line 5) while the PPO policies are always head-downward.

In Hopper and HalfCheetah, IPD is able to improve the primal task performance by avoiding always getting trapped in some certain sub-optimal behaviors.

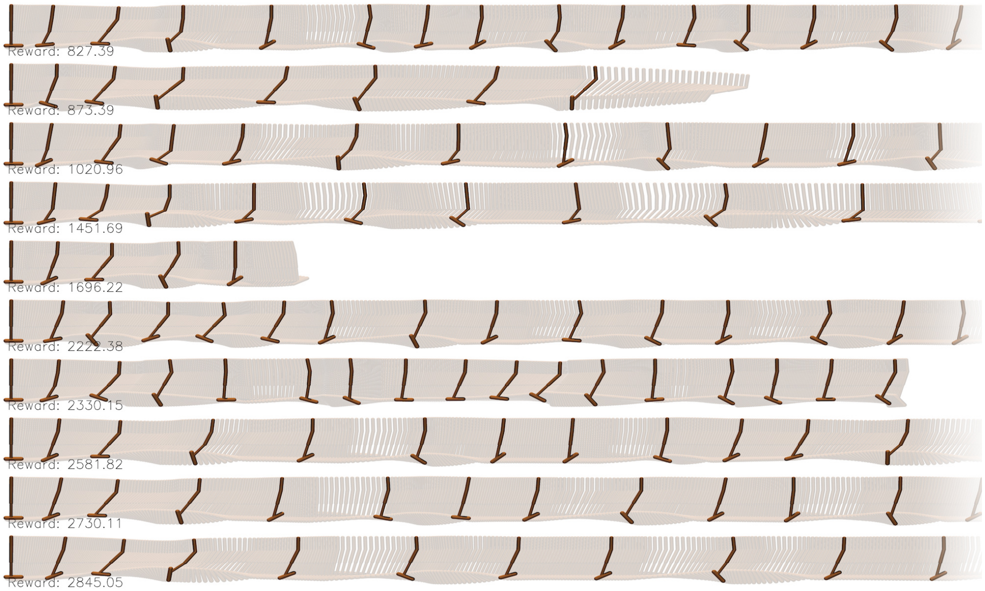

Figure 5: The visualization of policy behaviors of agents trained by our method in Hopper-v3 environment. Agents learn to jump with different strides.

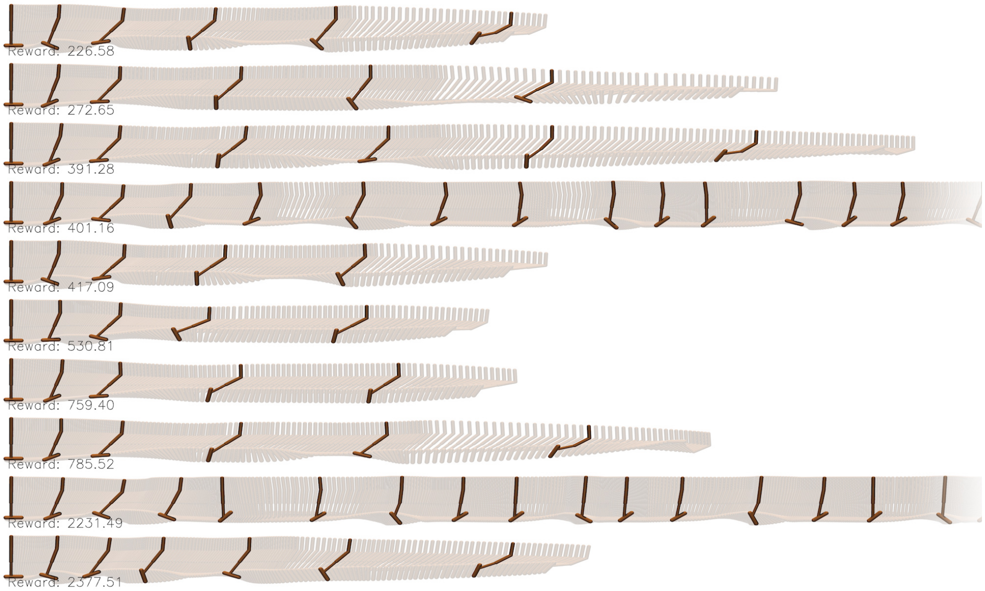

Figure 6: The visualization of policy behaviors of agents trained by PPO in Hopper-v3 environment. Most agents learn a policy that can be described as *Jump as far as possible and fall down*, leading to relative poor performance.

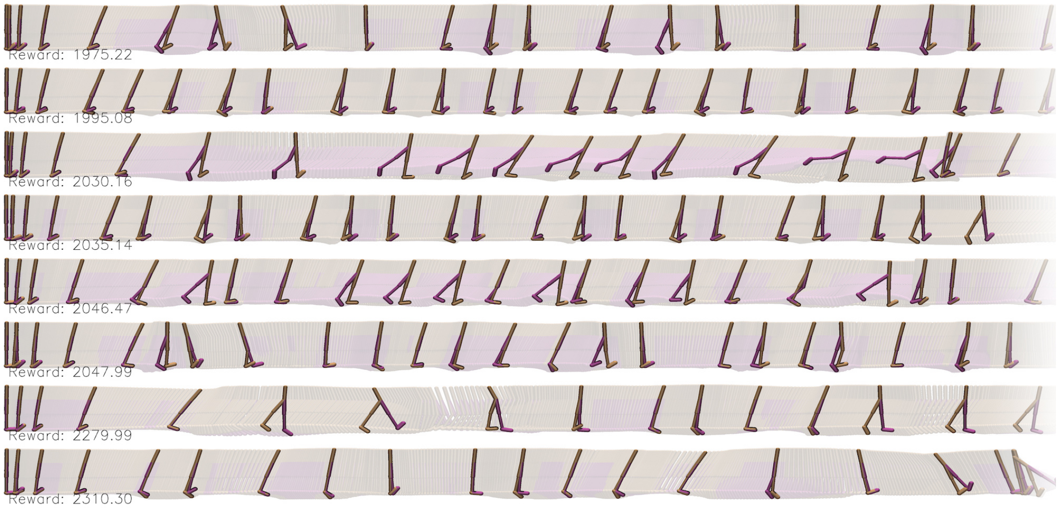

Figure 7: The visualization of policy behaviors of agents trained by our method in Walker2d-v3 environment. Instead of bouncing at the ground using both legs, our agents learns to use both legs to step forward.

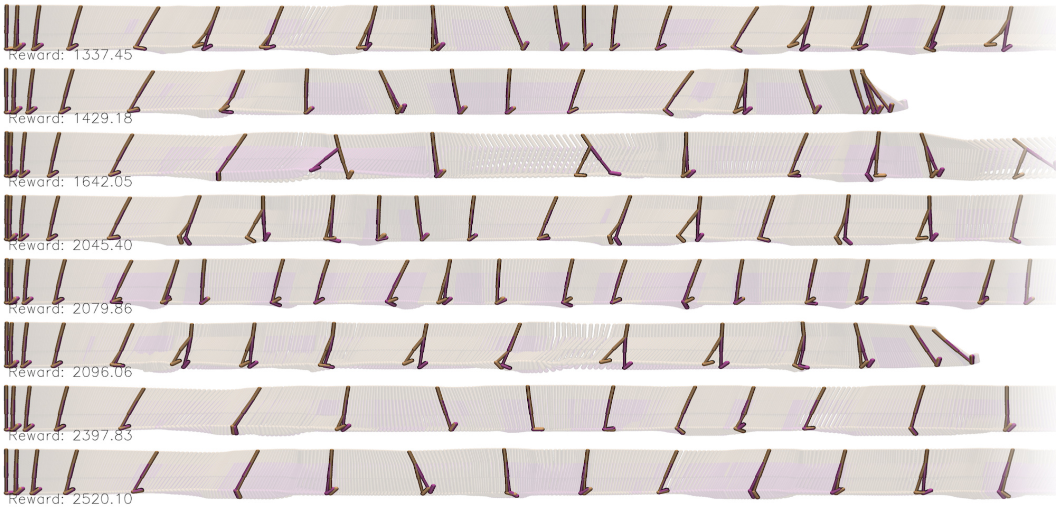

Figure 8: The visualization of policy behaviors of agents trained by PPO in Walker2d-v3 environment. Most of the PPO agents only learn to use their right leg to support the body and jump forward.

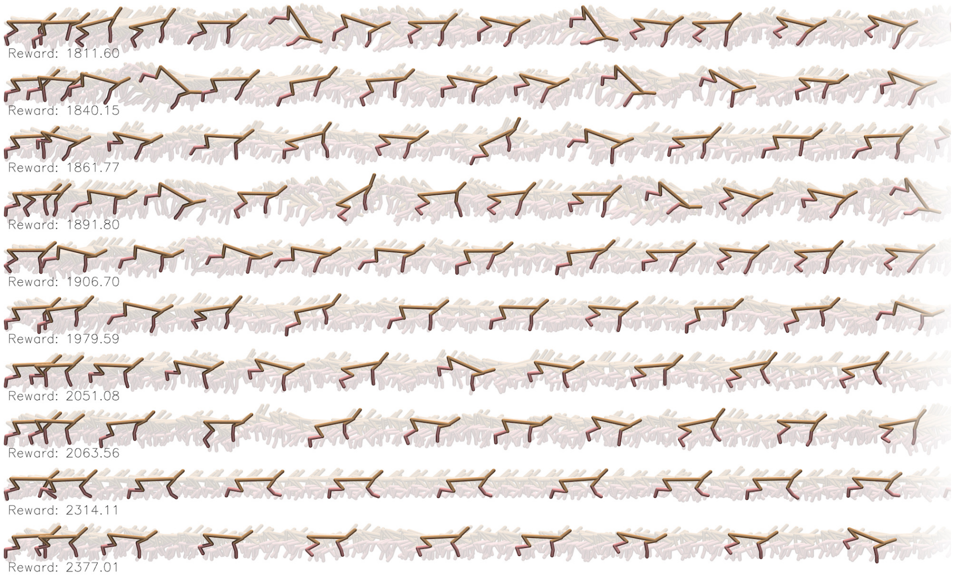

Figure 9: The visualization of policy behaviors of agents trained by our method in HalfCheetah-v3 environment. Our agents run much faster compared to PPO agents and at the mean time several patterns of motion have emerged.

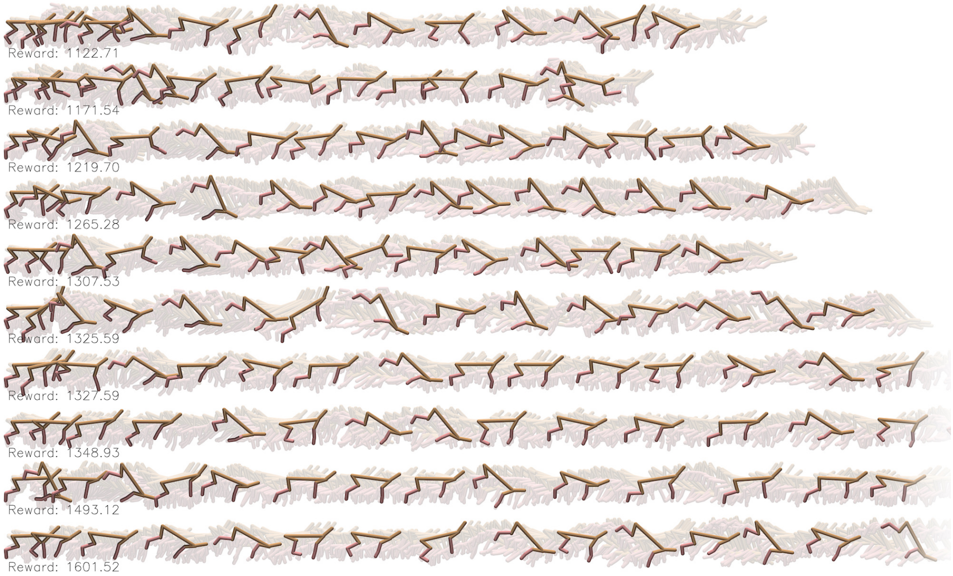

Figure 10: The visualization of policy behaviors of agents trained by PPO in HalfCheetah-v3 environment. Since we only draw fixed number of frames in each line, in the limited time steps the PPO agents can not run enough distance to leave the range of our drawing, which shows that our agents run much faster.