Learning with Identity and Uniqueness through Social Constraint

Hao Sun, Jiankai Sun, Zhenghao Peng, Dahua Lin, Bolei Zhou
The Chinese University of Hong Kong
{sh018,sj019,pengzh,dhlin,bzhou}@ie.cuhk.edu.hk

Abstract

The evolution of humans not only relies on their skills developed from interactions with the environment but also sparked by social influences. Humans pursue their fulfillment with both social identity and uniqueness. In this work, we introduce social influence into the scheme of reinforcement learning. Specifically, we consider the common goal of achieving high performance as social identity and employ social uniqueness as another motivation in learning. We take both social identity and uniqueness as constraints in our learning scheme and demonstrate the proposed method on a series of locomotion tasks. Empirical results show our agents learn to move forward with different behaviors, and in some cases, our method results in performance improvement as byproducts.

1 Introduction

Darwin introduced the theory of *survival of the fittest* [1], showing the evolution of lives comes from interactions between lives and the environments they live in, during which time the direction of evolution is determined. And to survive in different circumstances, lives must learn to master new skills. e.g. mammals learn to breathe with their lungs so that then they are able to move from ocean to land. In the ancient past, Homo erectus learns to live in caves in winter and learns to fire in order to survive in cold winters [2, 3]. As for the sociality, some of them are good at hunting while some are good at cultivation [4, 5]. Undeniably, mastering different strategies is crucial for the survival of humans in different situations.

The mechanism of reinforcement learning (RL) is quite similar: an agent learns by interacting with the environments to survive and gain as much reward as possible [6]. The work of Heess et al. shows rich environments help RL algorithms to learn powerful locomotion skills [7], demonstrating the RL paradigm can be regarded as the survival of the fittest. Another essential point in the evolution of human is the social influence [8–12]. While previous work focuses on applying social motivation to Multi-Agent Reinforcement Learning (MARL) settings [13–16], how to motivate single RL agent learning with social motivation is still an open question. We are interested in the question that how and to what extent can social identity and uniqueness motivate RL agent to pursue its goal in a characteristic way.

In this work, we draw the key insight of social influence in human evolution, and address the challenge of motivating RL agents by both identity (i.e. achieve their primal goal) and uniqueness (i.e. achieve their goal in different ways). Intuitively, the identity can help our agents to perform well in the environment, on the other hand, the uniqueness can inspire agents to be different, explore more solutions and pursue more excellent behaviors.

Our proposed method changes the vanilla RL scheme by combining the consideration of social identity and uniqueness of agents with the primal task environment, so that agents in our scheme learns from interactions not only with the environment but also with social influences. The learning scheme is illustrated in Fig 1. To sum up our contributions:

33rd Conference on Neural Information Processing Systems (DeepRL Workshop2019), Vancouver, Canada.

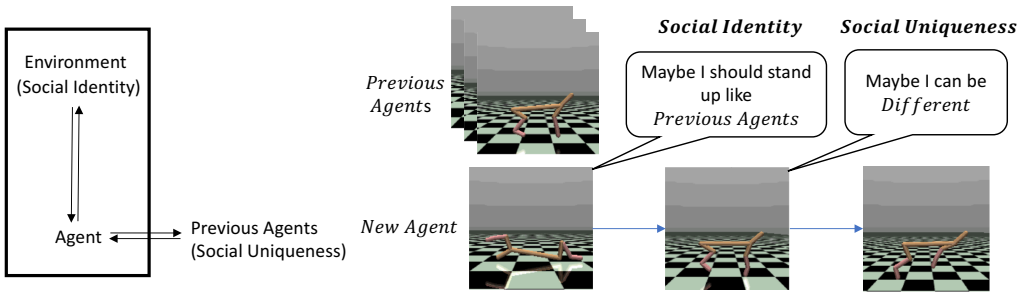

Figure 1: Our learning scheme and intuitively illustration. Our agents will learn to perform well in the given environment with social identity motivation, and learn to perform differently with each other with social uniqueness motivation. The right figure illustrates learning with social identity and uniqueness can be interpreted as a kind of implicit curriculum.

- We propose a new RL scheme where agents interact not only with the environment but also with previous policies. The tough multi-objective optimization problem is reformed into a single objective optimization problem with constraints.
- We introduce an effective policy difference evaluation metric based on the accumulative effect in RL. The metric is based on accumulative action Total Variance Divergence (TVD) between policies, which is more portable than previous neural network based approaches. Moreover, our metric results in immediately intrinsic rewards instead of episodic rewards and therefore substantially motivates the learning process.
- We apply our method in several benchmark locomotion tasks and show our method can learn various well performed policies based on PPO [17]. Furthermore, our approach can help agents to avoid local minimum and can be interpreted as a kind of implicit curriculum learning in certain cases.

2 Related Work

Intrinsic motivation methods. The Variational Information Maximizing Exploration (VIME) method [18] is designed by Houthooft et al. to tackle the sparse reward problems. In VIME, an intrinsic reward term based on the maximization of information gains is added to contemporary RL algorithms to encourage exploration. The curiosity-driven methods, proposed by Pathak et al. [19] and Burda et al. [20] define intrinsic rewards according to prediction errors of neural networks. i.e., when taking previous unseen states as inputs, networks trained with previous states will tend to predict with low accuracy, so that such prediction errors can be viewed as rewards. Burda et al. proposed Random Network Distillation (RND) to quantify intrinsic reward by prediction differences between a fixed random initialized network and another randomly initialized network trained with previous state information [21]. Liu et al. proposed Competitive Experience Replay (CER) [22], in which they use two actors and a centralized critic, and defined an intrinsic reward by the state coincidence of two actors. The values of intrinsic rewards are fixed to be ± 1 for the two actors separately. All of those approaches leverage the weighted sum of the external rewards, i.e., the primal rewards provided by environments, and intrinsic rewards that provided by different heuristics. A challenging problem is the trade-off between external rewards and intrinsic rewards. The Task-Novelty Bisector (TNB) learning method introduced by Zhang et al. aims to solve such problem by jointly optimize the extrinsic rewards and intrinsic rewards [23]. Specifically, TNB updates the policy in the direction of the angular bisector of the two gradients, i.e., gradients of the extrinsic and intrinsic objective functions. However, the foundation of such joint optimization is not solid. Besides, creating an extra intrinsic reward function and evaluating the novelty of states or policies always requires additional neural networks such as auto-encoders. Thus extra computation expenses are needed.

Diverse behaviors from rich environments and algorithms. Heess et al. introduce the Distributed Proximal Policy Optimization (DPPO) method and enable agents with simulated bodies to learn complex locomotion skills in a diverse set of challenging environments [7]. Although the learning reward they utilize is straightforward, the skills their policy learned are quite impressive and effective

in traveling terrains and obstacles. Their work shows that rich environments can encourage the emergence of different locomotion behaviors, but extra manual efforts are required in designing such environments. The research of Such et al. shows that different RL algorithms may converge to different policies for the same task [24]. The authors find that algorithms based on policy gradient tend to converge to the same local optimum in the game of Pitfall, while off-policy and value-based algorithms are prone to learn sophisticated strategies. On the contrary, in this paper, we are more interested in how to learn different policies through a single learning algorithm and learn the capability of avoiding local optimum.

3 Quantifying the Distance between Policies

To encourage the emergence of behavioral diversity in RL, we first define a metric to measure the difference between policies, which is the foundation for the later algorithm we propose. We denote the learned 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 the following, we omit π and denote a policy π_{θ_i} as θ_i for simplicity unless stated otherwise.

3.1 Definition of Policy Uniqueness

A metric should satisfy three important properties, namely the identity, the symmetry as well as the triangle inequality.

Definition 1 *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. $d(x, z) \leq d(x, y) + d(y, z)$.

We use the Total Variance Divergence D_{TV} [25] to measure the distance between policies. Concretely, for discrete probability distributions p and q, this distance is defined as $D_{TV}(p, q) = \sum_i |p_i - q_i|$. [1][2]

Theorem 1 (Metric Space $(\Theta, \overline{D}^{\rho}_{TV})$) *The expectation of $D_{TV}(\cdot, \cdot)$ of two policies over any state distribution $\rho(s)$:*

$$\overline{D}^{\rho}_{TV}(\theta_i, \theta_j) := {}_{s\sim\rho(s)}[D_{TV}(\theta_i(s), \theta_j(s))], \tag{1}$$

is a metric on Θ, thus $(\Theta, \overline{D}^{\rho}_{TV})$ is a metric space.

The proof of Theorem 1 is in Appendix A. It is worth mentioning that, although TVD is used in our work, we can easily extend the result to use other distance between distributions as substitutes of TVD (e.g. Jensen Shannon divergence D_{JS} or Wasserstein metric D_W) [26–28], and similar results can be get

Corollary 1 *Let $\overline{D}^{\rho}_{JS} := {}_{s\sim\rho(s)}[D_{JS}(\theta_i(s), \theta_j(s))]$ and $\overline{D}^{\rho}_{W} := {}_{s\sim\rho(s)}[D_W(\theta_i(s), \theta_j(s))]$, $(\Theta, \overline{D}^{\rho}_{JS})$ and $(\Theta, \overline{D}^{\rho}_{W})$ are also metric spaces.*

On top of the metric space $(\Theta, \overline{D}^{\rho}_{TV})$, we could then compute the uniqueness of a policy.

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

$$\mathrm{U}(\theta|\Theta_{ref}) := \min_{\theta_j \in \Theta_{ref}} \overline{D}^{\rho}_{TV}(\theta, \theta_j). \tag{2}$$

Consequently, to motivate RL with the social uniqueness, we hope our method can maximize the uniqueness of a new policy, i.e., $\max_\theta \mathrm{U}(\theta|\Theta_{ref})$, where the Θ_{ref} includes all the existing policies.

[1] It can be extended to continuous state and action spaces by replacing the sums with integrals.
[2] The factor $\frac{1}{2}$ in [25] is omitted in our work for conciseness.

3.2 Estimation of $\overline{D}^{\rho}_{TV}(\theta_i, \theta_j)$

In practice, the calculation of $\overline{D}^{\rho}_{TV}(\theta_i, \theta_j)$ is based on Monte Carlo estimation. i.e., we need to sample s from $\rho(s)$. Although in finite state space we can get precise estimation after establishing ergodicity, problem arises when we are facing continuous state cases. i.e. it is difficult to efficiently get enough samples.

Formally, we denote the domain of $\rho(s)$ as $\mathcal{S}$ and denote the domain of $\rho_\theta(s)$ as $\mathcal{S}_\theta \subset \mathcal{S}$, where $\rho_\theta(s) := \rho(s|s \sim \theta)$ and in finite time horizon problems $\rho(s|s \sim \theta) = P(s_0 = s|\theta) + P(s_1 = s|\theta) + ... + P(s_T = s|\theta)$. As we only care about the reachable regions, the domain $\mathcal{S}$ can be divided by $\mathcal{S} = \lim_{N\to\infty} \bigcup_{i=1}^{N} \mathcal{S}_{\theta_i}$.

In order to improve the sample efficiency, we propose to approximate $\overline{D}^{\rho}_{TV}(\theta_i, \theta_j)$ with $\overline{D}^{\rho_\theta}_{TV}(\theta_i, \theta_j)$, where θ is a certain fixed behavior policy that irrelevant to θ_i, θ_j. Such approximation requires a necessary condition:

Condition 1 *The domain of possible states are similar between different policies:*

$$\sum_{s\in\mathcal{S}} P(s \in (\mathcal{S}_\theta \cup \mathcal{S}_{\theta_j}) \setminus (\mathcal{S}_\theta \cap \mathcal{S}_{\theta_j})) \ll \sum_{s\in\mathcal{S}} P(s \in (\mathcal{S}_\theta \cap \mathcal{S}_{\theta_j})), \forall j. \tag{3}$$

When such condition holds, we can use $\rho(s|s \sim \theta)$ as our choice of $\rho(s)$, and the properties in Definition 1 still holds.

In practice, the Condition 1 always holds as we can ensure this by adding sufficiently large noise on θ, while the permitted state space is always limited. And for more general cases, to satisfy the properties in Definition 1, we must sample s from $\mathcal{S}_\theta \cup \mathcal{S}_{\theta_j}$, accordingly,

$$\begin{aligned}
\overline{D}^{\rho}_{TV}(\theta, \theta_j) &= \mathbb{E}_{s\sim(\mathcal{S}_\theta\cup\mathcal{S}_j)}[D_{TV}(\theta(s), \theta_j(s))] \\
&= \mathbb{E}_{s\sim(\mathcal{S}_\theta\cap\mathcal{S}_{\theta_j})}[D_{TV}(\theta(s), \theta_j(s))] + \mathbb{E}_{s\sim(\mathcal{S}_\theta\cup\mathcal{S}_{\theta_j})\setminus\mathcal{S}_{\theta_j}}[D_{TV}(\theta(s), \mathcal{N})] \\
&+ \mathbb{E}_{s\sim(\mathcal{S}_\theta\cup\mathcal{S}_{\theta_j})\setminus\mathcal{S}_\theta}[D_{TV}(\mathcal{N}, \theta_j(s))]
\end{aligned} \tag{4}$$

where $\mathcal{N}$ represents random action when a policy have never been trained or visited such state domain. Plugging Eq.(4) into Eq.(2), the objective function of policy differentiation is

$$\begin{aligned}
&\max_\theta \min_{\theta_j\in\Theta_{ref}} \overline{D}^{\rho}_{TV}(\theta, \theta_j) \\
&= \mathbb{E}_{s\sim(\mathcal{S}_\theta\cap\mathcal{S}_{\theta_j})}[D_{TV}(\theta(s), \theta_j(s))] + \mathbb{E}_{s\sim(\mathcal{S}_\theta\cup\mathcal{S}_{\theta_j})\setminus\mathcal{S}_{\theta_j}}[D_{TV}(\theta(s), \mathcal{N})] \\
&+ \mathbb{E}_{s\sim(\mathcal{S}_\theta\cup\mathcal{S}_{\theta_j})\setminus\mathcal{S}_\theta}[D_{TV}(\mathcal{N}, \theta_j(s))]
\end{aligned} \tag{5}$$

While the first two terms are related to the policy θ, the last term is only related to the domain $\mathcal{S}_\theta$. If we enable sufficient exploration in training as well as in the initialization of θ, the last term will disappear (i.e. $\mathcal{S}_{\theta_j} \subset \mathcal{S}_\theta$). Hence we can also use $\overline{D}^{\rho_{\theta_i}}_{TV}(\theta_i, \theta_j)$ as an approximation of $\overline{D}^{\rho}_{TV}(\theta_i, \theta_j)$ in training of θ_i as long as sufficient exploration is guaranteed.

Proposition 1 (Unbiased Single Trajectory Estimation) *The estimation of $\rho_\theta(s)$ using a single trajectory τ is unbiased.*

The proof of Proposition 1 is in Appendix B. Given the definition of uniqueness and a practically unbiased sampling method, the next step is to develop an efficient learning algorithm.

4 Constrained Optimization Perspectives for Uniqueness Seeking

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

To improve the behavioral diversity of different agents, the learning objective must take both reward from the primal task and the policy uniqueness into consideration. Previous approaches [18–22] often directly write the weighted sum of the reward from the primal task and the intrinsic reward $g_{\text{int}} = \sum_{t=0} \gamma^t r_{\text{int},t}$, where $r_{\text{int},t}$ denotes the *intrinsic reward* (e.g., $r_{\text{int}} = \min_{\theta_j \in \Theta_{ref}} \overline{D}^{\rho}_{TV}(\theta, \theta_j)$ as the uniqueness reward in our case) as follows,

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{\tau \sim \theta}[g_{\text{total}}] = \max_{\theta \in \Theta} \quad \mathbb{E}_{\tau \sim \theta}[\alpha \cdot g_{\text{task}} + (1 - \alpha) \cdot g_{\text{int}}], \tag{6}$$

where $0 < \alpha < 1$ is a weight parameter. Such an objective is sensitive to the selection of α as well as the formulation of r_{int}. For example, formulating the intrinsic reward r_{int} as $\min_{\theta_j} \overline{D}^{\rho}_{TV}(\theta, \theta_j)$, $\exp\left[\min_{\theta_j} \overline{D}^{\rho}_{TV}(\theta, \theta_j)\right]$ and $-\exp\left[-\min_{\theta_j} \overline{D}^{\rho}_{TV}(\theta, \theta_j)\right]$ will result in 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 reward, leading to the failure of agent in solving the primal task.

To tackle such issue, the crux is to deal with the conflict between different objectives. The work of Zhang et al. proposes the Task Novel Bisector (TNB) [23], where the task reward is regarded as the dominant one while the task novelty reward is regarded as subordinate. As we will show later, the TNB actually acts as a heuristic approximation of the Feasible Direction Methods (FDMs) [29] in constrained optimization problems, where the primal task is regarded as the objective and the novelty is the constraint.

However, in TNB do not fully resemble the FDMs as the novelty term still perform as a reward term, i.e., the larger the better. Such formulation will impact the learning of primal task in several ways: first, the learning rate is determined by both rewards, leading the final result rely on both the shape of extrinsic reward and novelty reward. Second, the learning rate will be extremely low when $\cos(\vec{g}_{novel}, \vec{g}_{task}) \approx -1$ so that the learning process prone to be trapped in local minima. Third, the result policy will tend to be as different as previous policies as possible, sometimes it will hinder the performance of learned novel policies.

In order to tackle the above problems and be able to adjust the extend of novelty in new policies, we propose to solve the novelty seeking problem in constrained optimization perspective. We also draw inspiration from the observation of social uniqueness motivation in human society. While people trying to pursue their goal in different ways, they do not push themselves to the extreme. The uniqueness motivation, as an intrinsic motivation in human decision making, plays more like a constraint rather than an additional target. Apply this insight in novel policy seeking, we change the multi-objective optimization problem in Eq.(6) into a constrained optimization problem as:

$$\begin{aligned} \max_{\theta \in \Theta} \quad & \mathbb{E}_{\tau \sim \theta}[g_{\text{task}}], \\ s.t. \quad & \overline{r}_{\text{int},t} - r_0 \geq 0, \forall t = 1, 2, ..., T, \end{aligned} \tag{7}$$

where r_0 is a threshold indicating minimal permitted uniqueness, 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 are more care about the long term differences. Therefore, we use the cumulative uniqueness as constraints. Moreover, the constraints can be 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 $\sum_{t=t_S}^{t=\tau}(r_{\text{int},t} - r_0) \geq 0, \tau = S, ..., T,$.

5 Relation between Different Approaches and Constrained Optimization Methods

We note here, the WSR, TNB and IPD methods correspond to three approaches in constrained optimization problem. For simplicity, we consider Eq.(7) with a more concise notion $g_{\text{int},t} - g_{0,t} \geq 0$, where $g_{\text{int},t} = \sum_{t=0}^{t} r_{\text{int},t}$, i.e.,

$$\begin{aligned} \max_{\theta \in \Theta} \quad & f(\theta) = \mathbb{E}_{\tau \sim \theta}[g_{\text{task}}] \\ s.t. \quad & g_t(\theta) = g_{\text{int},t} - g_{0,t} \geq 0, t = 1, 2, ..., T \end{aligned} \tag{8}$$

As the optimization of policy is based on batches of trajectory samples and is implemented with stochastic gradient descent, Eq.(8) can be further simplified as:

$$\begin{aligned} \max_{\theta\in\Theta} \quad & f(\theta) = \mathbb{E}_{\tau\sim\theta}[g_{\text{task}}] \\ s.t. \quad & g(\theta) = \overline{g}_t(\theta) \geq 0 \end{aligned} \tag{9}$$

where $\overline{g}_t(\theta)$ denotes the average over a trajectory.

WSR: Penalty Method The Penalty Method considers the constraints of Eq.(9) by putting constraint $g(\theta)$ into a penalty term, and then solve the unconstrained problem

$$\max_{\theta\in\Theta} \quad f(\theta) + \frac{1-\alpha}{\alpha}\min\{g(\theta), 0\} \tag{10}$$

using an iterative manner, and the limit when $\alpha \to 0$ lead to the solution of the primal constrained problem. As an approximation, WSR choose a fixed weight term α, and use 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.

TNB: Feasible Direction Method The Taylor series of $g(\theta)$ at point $\bar{\theta}$ is

$$g(\bar{\theta} + \lambda\vec{p}) = g(\bar{\theta}) + \nabla_\theta g(\bar{\theta})^{\mathrm{T}}\lambda\vec{p} + O(||\lambda\vec{p}||) \tag{11}$$

The Feasible Direction Method (FDM) considers the constraints of Eq.(9) by first finding a direction $\vec{p}$ satisfies

$$\begin{aligned} &\nabla_\theta f^{\mathrm{T}} \cdot \vec{p} > 0 \\ &\nabla_\theta g^{\mathrm{T}} \cdot \vec{p} > 0 \quad \text{if} \quad g = 0 \end{aligned} \tag{12}$$

so that for small λ, we have

$$g(\bar{\theta} + \lambda\vec{p}) = g(\bar{\theta}) + \lambda\nabla_\theta g(\bar{\theta})^{\mathrm{T}}\vec{p} > g(\bar{\theta}) = 0 \quad \text{if} \quad g(\bar{\theta}) = 0 \tag{13}$$

and

$$g(\bar{\theta} + \lambda\vec{p}) = g(\bar{\theta}) + \lambda\nabla_\theta g(\bar{\theta})^{\mathrm{T}}\vec{p} > 0 \quad \text{if} \quad g(\bar{\theta}) > 0 \tag{14}$$

The TNB method, by using the bisector of gradients $\nabla_\theta f$ and $\nabla_\theta g$, select $\vec{p}$ to be

$$\vec{p} = \begin{cases} \nabla_\theta f + \frac{|\nabla_\theta f|}{|\nabla_\theta g|}\nabla_\theta g \cdot \cos\left(\nabla_\theta f, \nabla_\theta g\right) & \text{if } \cos\left(\nabla_\theta f, \nabla_\theta g\right) \leq 0 \\ \nabla_\theta f + \frac{|\nabla_\theta f|}{|\nabla_\theta g|}\nabla_\theta g & \text{if } \cos\left(\nabla_\theta f, \nabla_\theta g\right) > 0 \end{cases} \tag{15}$$

Clearly, Eq.(15) satisfies Eq.(12), but it is more strict than Eq.(12) as the $\nabla_\theta g$ term always exists during the optimization of TNB. In TNB, the learning stride is fixed to be $\frac{|\nabla_\theta f|+|\nabla_\theta g|}{2}$, leading to problem when $\nabla_\theta f \to 0$, which shows the final optimization result will heavily rely on the selection of g. i.e., the shape of g is crucial for the success of TNB.

IPD: Interior Point Methods (IPMs) In vanilla IPMs, the constrained optimization problem in Eq.(9) is solved by reforming it to an unconstrained form with an additional barrier term $\alpha\frac{1}{g(\theta)}$ in the objective as

$$\max_{\theta\in\Theta} \quad f(\theta) + \alpha\frac{1}{g(\theta)} \tag{16}$$

or use the barrier term of $-\alpha\log g(\theta)$ instead:

$$\max_{\theta\in\Theta} \quad f(\theta) - \alpha\log g(\theta) \tag{17}$$

where α, the barrier factor, is a small positive number. 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 fast. It is clear that the solution of the objective with barrier term will get closer to the primal objective as α getting smaller. Thus in practice, such methods will choose a sequence of $\{\alpha_i\}$ such that $0 < \alpha_i < \alpha_{k+1}$ and $\alpha_i \to 0$ as $k \to \infty$ The limit of Eq.(16), Eq.(17) when $\alpha \to 0$ then leads to the solution of Eq.(9). The work of Conn et al. and Wright et al. provide proofs of the convergence [30, 31].

Directly applying this method is computationally challenging and numerically unstable, especially when α is small. A more natural way can be used: since the learning process is based on sampled transitions, we can simply bound the collected transitions in the feasible region by permitting previous 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 then naturally obtain a new policy that has sufficient uniqueness. In this way, we no longer need to consider the trade-off problem between intrinsic and extrinsic rewards deliberately. The learning process of our method is thus more robust and no longer suffer from objective inconsistency. Algorithm.1 shows the pseudo code of IPD based on PPO, where the blue lines show the addition to primal PPO algorithm.

Algorithm 1 IPD with PPO, Actor-Critic Style

Require

- a behavior policy θ_{old}
- a set of previous policies $\{_j\}, j = 1, 2, ..., M$
- a uniqueness metric $U(\theta, \{\theta_j\}|\rho) = U(\theta, \{\theta_j\}|\tau) = \min_{\theta_j} \overline{D}^{\tau}_{TV}(\theta, \theta_j)$
- a uniqueness threshold r_0, starting point t_S

Initialize θ_{old}
for iteration $= 1, 2, ...$ **do**
  for actor $= 1, 2, ..., N$ **do**
    for t $= 1, 2, ..., T$ **do**
      Run policy θ_{old} in environment, get trajectory τ
      if $U(\theta_{\text{old}}, \{\theta_j\}|\tau) - r_0 < 0$, AND $t > t_S$ **then**
        done = True
      end if
      if done **then**
        break
      end if
    end for
    Compute advantage estimates $\hat{A}_1, ..., \hat{A}_T$
  end for
  Optimize surrogate $\mathcal{L}^{\text{CLIP}}$ w.r.t. θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{old} \leftarrow \theta$
end for

Instead, in this work we propose to solve the constrained optimization problem Eq.(7) by resembling the Interior Point Methods (IPMs) [32, 33]. In vanilla IPMs, the constrained optimization problem in Eq.(7) is solved by reforming it to an unconstrained form with an additional barrier term in the objective as

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{\tau \sim \theta}[g_{\text{task}} + \sum_{t=0}^{T} \alpha \log (r_{\text{int},t} - r_0)]. \tag{18}$$

The limit of Eq.(18) when $\alpha \to 0$ then leads to the solution of Eq.(7). Readers please refer to Appendix 5 for more discussion on the correspondence between those novel policy seeking methods and constrained optimization methods.

However, directly applying the IPMs is computationally challenging and numerically unstable, especially when α is small. Luckily, in our proposed RL paradigm where the behavior of an agent

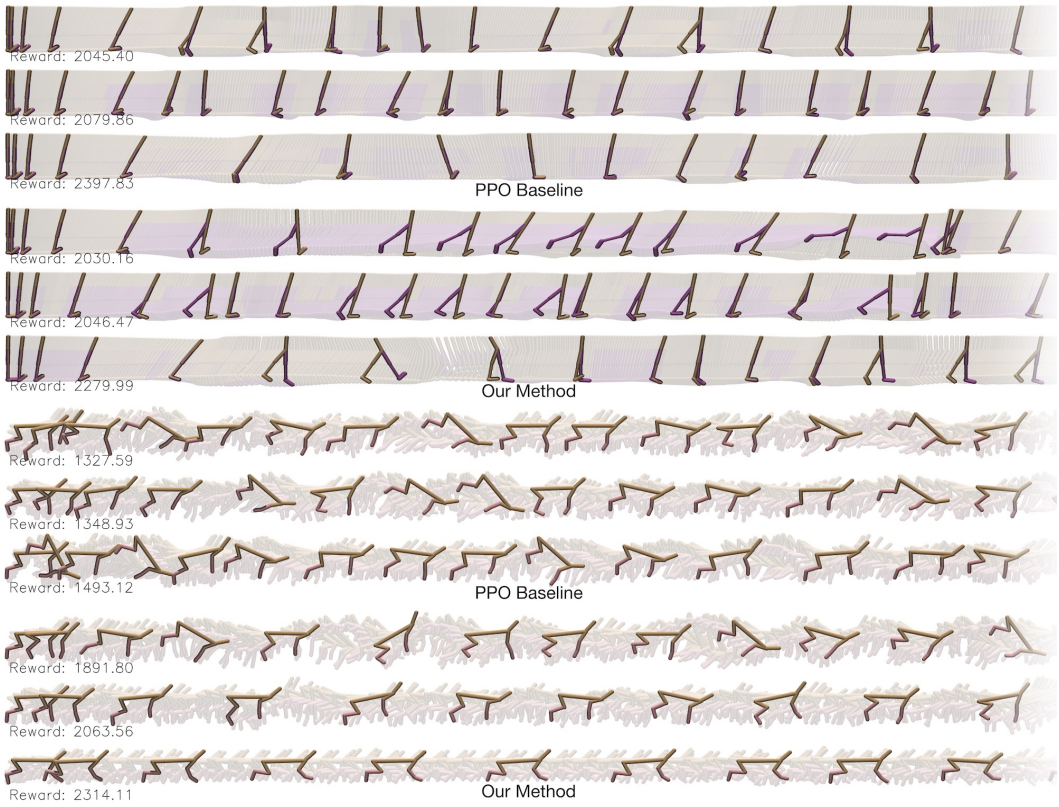

Figure 2: Results of policy differentiation on Walker2d-v3 and HalfCheetah-v3. Compared to the PPO baseline, our method significantly diversifies trained policies while maintaining their performances.

is influenced by its peers, a more natural way can be used. Precisely, since the learning process is based on sampled transitions, we can simply bound the collected transitions in the feasible region by permitting previous 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 then naturally obtain a new policy that has sufficient uniqueness. In this way, we no longer need to consider the trade-off problem between intrinsic and extrinsic rewards deliberately. The learning process of our method is thus more robust and no longer suffer from objective inconsistency. As our formulation of the constrained optimization problem Eq.(7) is inspired by IPMs, we name our approach as Interior Policy Differentiation (IPD) method.

6 Experiments

The MuJoCo environment We demonstrate our proposed method on the OpenAI Gym where the physics engine is based on MuJoCo [34, 35]. Concretely, we test on three locomotion environments, the Hopper-v3 (11 observations and 3 actions), Walker2d-v3 (11 observations and 2 actions), and HalfCheetah-v3 (17 observations and 6 actions). In our experiments, all the environment parameters are set as default values.

Uniqueness beyond intrinsic stochasticity Experiments in Henderson et al. show that policies that perform differently can be produced by simply selecting different random seeds before training [36]. Before applying our method to improve behavior diversity, we firstly benchmark how much uniqueness can be generated from the stochasticity in the training process of vanilla RL algorithms as well as the random weight initialization. In this work, we mainly demonstrate our proposed method based on PPO [17]. The extension to other popular algorithms is straightforward. We also compare our

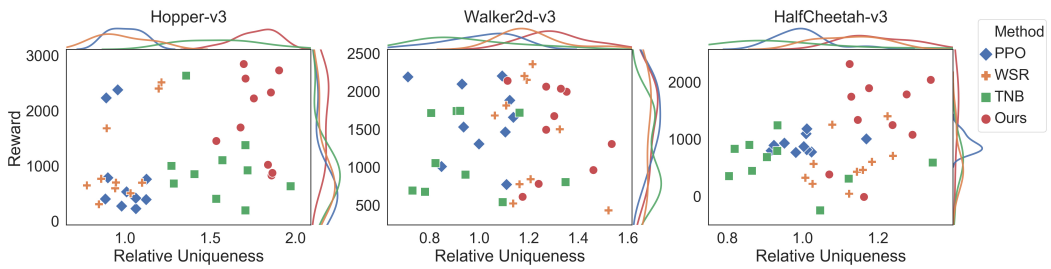

Figure 3: The scatter plot of different policies in terms of Uniqueness and Performance in Hopper-v3, Walker2d-v3 and HalfCheetah-v3 environments. The value of uniqueness is normalized to relative uniqueness by regarding the averaged uniqueness of PPO policies as the baseline.

proposed method with the TNB and weighted sum reward (WSR) approaches as different ways to combine the goal of the task and the uniqueness motivation [23]. More implementation details are depicted in Appendix D.

6.1 Uniqueness and Performance Comparison

According to Theorem 2, the uniqueness r_{int} in equation (7) under our uniqueness metric can be unbiased approximated by $r_{\text{int}} = \min_{\theta_j \in \Theta_{ref}} \overline{D}_{TV}^{\rho_\theta}(\theta(s_t), \theta_j(s_t))$. i.e., we utilize the metric directly in learning new policies instead of applying any kind of reshaping.

We implement WSR, TNB, and our method in the same experimental settings and for each method, 10 different policies are trained and try to be unique with regard to all previously trained policies sequentially. Concretely, the $1st$ policy is trained by ordinary PPO without any social influence. The $2nd$ policy should be different from $1st$ policy, and the $3rd$ should be different from the previous two policies, and so on. Fig.2 shows the qualitative results of our method. We visualize the motion of agents by drawing multiple frames representing the pose of agents at different time steps in the same row. The horizontal interval between consecutive frames is proportional to the velocity of agents. The settings of the frequency of highlighted frames and the correlation between interval and velocity are fixed for each environment. The visualization starts from the beginning of each episode and therefore the readers can get sense of the process of acceleration as well as the pattern of motion of agents clearly.

Fig. 3 shows our experimental results in terms of uniqueness (the x-axis) and the performance (the y-axis). Policies in the upper right are the more unique ones with higher performance. In Hopper and HalfCheetah, our proposed method distinctively outperforms other methods. In Walker2d, both WSR and our method work well in improving the uniqueness of policies, but none of the three methods can find way to surpass the performance of PPO apparently. Detailed comparison on the task related rewards are carried out in Table 1. A box figure depicting the performance of each trained policy and their reward gaining curve are disposed in Fig.4 and Fig.5 in Appendix C. And Fig.6 in Appendix C provides more detailed results from the view of uniqueness.

6.2 Success Rate of Each Method

In addition to averaged reward, we also use success rate as another metrics to compare the performance of different approaches. In this work, we consider a policy is success when its performance is at least as good as the averaged performance of policies trained without social influences. To be specific, we use the averaged final performance of PPO as the baseline. If a new 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. Through the success rate, we know the policy does not learn unique behavior at the expense of performance. Table 1 shows the success rate of all the methods, including the PPO baseline. The results show that our method can always surpass the average baseline during training. Thus the performance of our method can always be insured.

Table 1: The reward and success rate of learned 10 policies using different methods

Method	Reward			Success Rate		
	Hopper	Walker2d	HalfCheetah	Hopper	Walker2d	HalfCheetah
PPO	839 ± 753	$\mathbf{1611 \pm 467}$	913 ± 134	1.0	1.0	0.7
PPO+WSR	1083 ± 768	1429 ± 692	603 ± 407	1.0	0.7	0.4
PPO+TNB	1064 ± 644	1160 ± 484	592 ± 384	1.0	0.9	0.5
PPO+Ours	$\mathbf{1858 \pm 744}$	1506 ± 541	$\mathbf{1442 \pm 588}$	**1.0**	**1.0**	**0.9**

6.3 Better Policy Discovery

In our experiments, we observed noticeable performance improvements in the Hopper and the HalfCheetah environments. For the environment of Hopper, in many cases, the agents trained with PPO tend to learn a policy that jumps as far as possible and then fall to the ground and terminate this episode (please refer to Fig.10 in Appendix E). Our proposed method can prevent new policies from always falling into the same local minimum. After the first policy being trapped in a local minimum, the following policies will try other approaches to avoid the same behavior, explore other feasible action patterns, and thereafter the performance may get improved. Such property shows that our method can be a helpful enhancement of the traditional RL scheme, which can be epitomized as policies could make mistakes, but they should explore more instead of hanging around the same local minimum. The similar feature attributes to the reward growth in the environment of HalfCheetah.

Moreover, we can illuminate the performance improvement of HalfCheetah from another perspective. The environment of HalfCheetah is quite different from the other two for there is no explicit termination signal in its default settings (i.e., no explicit action like falling to the ground would trigger termination). At the beginning of the learning process, an agent will act randomly, resulting in massive repeat, trivial samples as well as large control costs. In our learning scheme, since the agent also interacts with the peers, it can receive termination signals from the peers to prevent wasting too much effort acting randomly. During the learning process in our method, an agent will first learn to terminate itself as soon as possible to avoid heavy control costs by imitating previous policies and then learns to behave differently to pursue higher reward. From this point of view, such learning process can be regarded as a kind of implicit curriculum.

7 Conclusion

In this work, we develop an efficient approach to motivate RL to learn diverse strategies inspired by social influence. After defining the distance between policies, we introduce the definition of policy uniqueness. Regarding the problem as constrained optimization problem, our proposed method, Interior Policy Differentiation (IPD), draws the key insight of the Interior Point Methods. And our experimental results demonstrate IPD can learn various well-behaved policies, and our approach can help agents to avoid local minimum and can be interpreted as a kind of implicit curriculum learning in certain cases.

Acknowledgement: We acknowledge discussions with Smart Ying Zeng, Thegreatestsis Jie Yi for the discussions on social influence motivations. This work was partially supported by SenseTime Group (CUHK Agreement No.7051699) and CUHK direct fund (No.4055098).

References

[1] Charles Darwin. *On the origin of species, 1859*. Routledge, 1859.

[2] Guanjun Shen, Xing Gao, Bin Gao, and Darryl E Granger. Age of zhoukoudian homo erectus determined with 26 al/10 be burial dating. *Nature*, 458(7235):198, 2009.

[3] Steven R James, RW Dennell, Allan S Gilbert, Henry T Lewis, JAJ Gowlett, Thomas F Lynch, WC McGrew, Charles R Peters, Geoffrey G Pope, Ann B Stahl, et al. Hominid use of fire in the lower and middle pleistocene: A review of the evidence [and comments and replies]. *Current Anthropology*, 30(1):1–26, 1989.

[4] Yongqiong Zong, Zhongyuan Chen, James B Innes, Chun Chen, Zhanghua Wang, and Hui Wang. Fire and flood management of coastal swamp enabled first rice paddy cultivation in east china. *Nature*, 449(7161):459, 2007.

[5] Christopher Boehm and Christopher Boehm. *Hierarchy in the forest: The evolution of egalitarian behavior*. Harvard University Press, 2009.

[6] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.

[7] Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, Martin Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

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

[9] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[10] Carel P van Schaik and Judith M Burkart. Social learning and evolution: the cultural intelligence hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1567):1008–1016, 2011.

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

[12] Yuval Noah Harari. *Sapiens: A brief history of humankind.* Random House, 2014.

[13] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, Dj Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049, 2019.

[14] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in neural information processing systems*, pages 3326–3336, 2018.

[15] Pedro Sequeira, Francisco S Melo, Rui Prada, and Ana Paiva. Emerging social awareness: Exploring intrinsic motivation in multiagent learning. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE, 2011.

[16] Alexander Peysakhovich and Adam Lerer. Consequentialist conditional cooperation in social dilemmas with imperfect information. *arXiv preprint arXiv:1710.06975*, 2017.

[17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[18] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Variational information maximizing exploration. 2016.

[19] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[20] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[21] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[22] Hao Liu, Alexander Trott, Richard Socher, and Caiming Xiong. Competitive experience replay. *CoRR*, abs/1902.00528, 2019.

[23] Yunbo Zhang, Wenhao Yu, and Greg Turk. Learning novel policies for tasks. *CoRR*, abs/1905.05252, 2019.

[24] Felipe Petroski Such, Vashisht Madhavan, Rosanne Liu, Rui Wang, Pablo Samuel Castro, Yulun Li, Ludwig Schubert, Marc Bellemare, Jeff Clune, and Joel Lehman. An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents. *arXiv preprint arXiv:1812.07069*, 2018.

[25] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[26] Dominik Maria Endres and Johannes E Schindelin. A new metric for probability distributions. *IEEE Transactions on Information theory*, 2003.

[27] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, page 31. IEEE, 2004.

[28] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[29] Guus Zoutendijk. *Methods of feasible directions: a study in linear and non-linear programming*. Elsevier, 1960.

[30] A Conn, Nick Gould, and Ph Toint. A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Mathematics of Computation of the American Mathematical Society*, 66(217):261–288, 1997.

[31] Stephen J Wright. On the convergence of the newton/log-barrier method. *Mathematical Programming*, 90(1):71–100, 2001.

[32] Florian A Potra and Stephen J Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1-2):281–302, 2000.

[33] George B Dantzig and Mukund N Thapa. *Linear programming 2: theory and extensions*. Springer Science & Business Media, 2006.

[34] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[35] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pages 5026–5033. IEEE, 2012.

[36] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A Proof of Theorem 1

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

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

B Proof of Proposition 1

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

C Details of Uniqueness and Performance

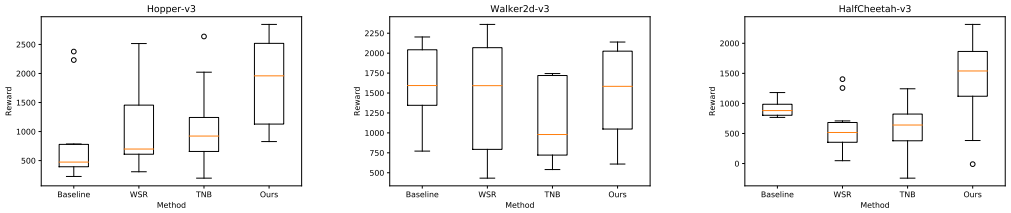

Figure 4: The performance of different methods in the Hopper, Walker and HalfCheetah environments. The results are collected from 10 learned policies based on PPO. The box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to show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

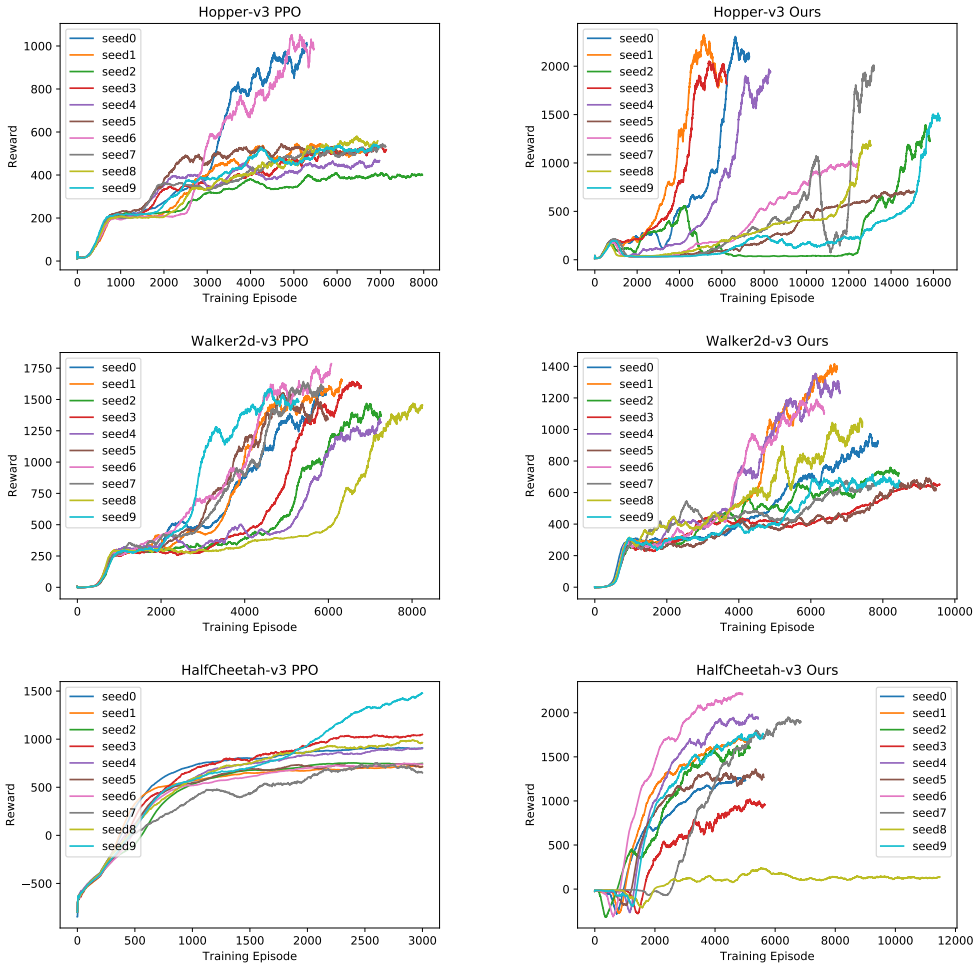

Figure 5: Avoid local minimum in Hopper and HalfCheetah: the left two figures show 10 policies generated by PPO in each environment, the right two figures show 10 policies generated by our method in each environment

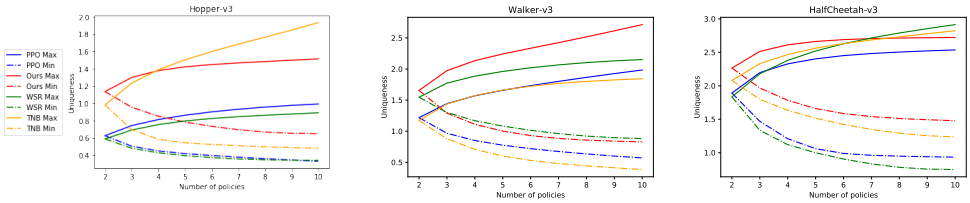

Figure 6: Maximal and minimal between policy uniqueness in Hopper, Walker2d and HalfCheetah environments. The results are averaged over all possible combinations of 10 policies. As TNB and WSR optimize the uniqueness reward directly, their uniqueness sometimes can exceed our proposed method. However, such direct optimization will lead to decreasing in task related performance as cost. To tackle the trade-off problem, carefully hyper-parameter tuning and reward shaping is always a must. Detailed comparison on the task related rewards are carried out in Table 1

D Implementation Details

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

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. Our ablation study on the choice of unit number in the second layer is detailed in Table.2, Table3 and Fig.7. Moreover, 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 (Table.2), performance (Table.3) and computation expense (i.e. the preference to use less unit when the other two factors are similar) into consideration.

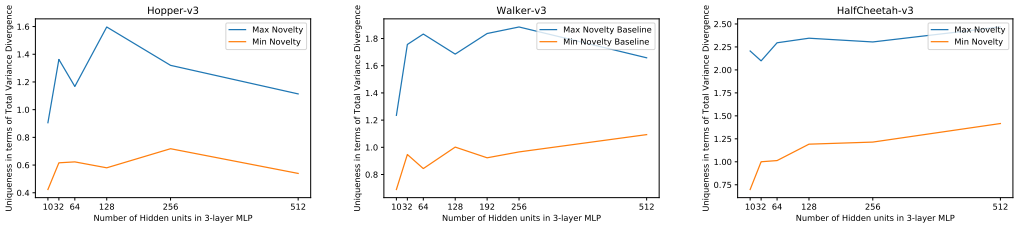

Figure 7: Uniqueness increases a little as network complexity rises

Table 2: The success rate of learned 10 novel policies by our method in different environments under different thresholds

Environment	Threshold	10 hidden	64 hidden	256 hidden	512 hidden
Hopper	0.5	0.7	0.8	0.7	–
	0.6	1.0	0.7	1.0	–
	0.7	0.9	0.4	0.7	–
	0.8	0.5	0.5	0.3	–
	0.9	0.3	0.1	0.4	–
Walker2d	1.0	0.2	0.8	1.0	–
	1.1	0.3	0.8	1.0	–
	1.2	0.1	0.9	1.0	–
	1.3	0.2	0.8	0.8	–
	1.4	0.1	0.6	1.0	–
HalfCheetah	1.1	–	0.3	0.9	0.8
	1.2	–	0.8	0.8	0.7
	1.3	–	1.0	1.0	1.0
	1.4	–	0.4	0.9	1.0
	1.5	–	0.2	0.2	0.7

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.

Table 3: The final training performance of learned 10 novel policies by our method in different environments under different thresholds, for Hopper and Walker, $h_1 = 10, h_2 = 64, h_3 = 256$; for HalfCheetah, $h_1 = 64, h_2 = 256, h_3 = 512$

Environment	Threshold	Average Performance			Top 30% Performance		
hid num		h_1	h_2	h_3	h_1	h_2	h_3
Hopper	0	450 ± 135	914 ± 735	704 ± 598	607 ± 30	1665 ± 726	1202 ± 737
	0.6	$\mathbf{1858 \pm 744}$	$\mathbf{1040 \pm 914}$	$\mathbf{1626 \pm 869}$	$\mathbf{2719 \pm 108}$	$\mathbf{2263 \pm 764}$	$\mathbf{2628 \pm 207}$
	0.7	1180 ± 740	593 ± 159	785 ± 462	2188 ± 510	769 ± 17	1287 ± 590
	0.8	397 ± 283	767 ± 743	950 ± 929	744 ± 152	1594 ± 875	2129 ± 896
	0.9	154 ± 142	235 ± 279	604 ± 554	335 ± 122	645 ± 113	1347 ± 387
	1.0	187 ± 229	298 ± 256	499 ± 754	490 ± 194	648 ± 141	1294 ± 979
Walker	0	$\mathbf{1611 \pm 467}$	1504 ± 502	$\mathbf{1724 \pm 584}$	$\mathbf{2163 \pm 48}$	2018 ± 54	$\mathbf{2311 \pm 45}$
	1.0	725 ± 487	1174 ± 599	1571 ± 692	1346 ± 468	2042 ± 77	2270 ± 37
	1.1	725 ± 654	$\mathbf{1506 \pm 541}$	1453 ± 480	1561 ± 598	$\mathbf{2079 \pm 44}$	1903 ± 104
	1.2	487 ± 375	1061 ± 346	1211 ± 657	880 ± 451	1552 ± 114	2114 ± 47
	1.3	405 ± 647	1138 ± 591	995 ± 420	1124 ± 795	1984 ± 197	1523 ± 313
	1.4	393 ± 518	831 ± 400	1333 ± 558	945 ± 652	1352 ± 273	2004 ± 106
HalfCheetah	0	1210 ± 391	1278 ± 373	1235 ± 317	1655 ± 296	1728 ± 179	1601 ± 241
	1.1	434 ± 415	1055 ± 265	914 ± 427	967 ± 365	1275 ± 42	1330 ± 297
	1.2	1167 ± 491	988 ± 446	948 ± 476	1679 ± 241	1441 ± 254	1466 ± 265
	1.3	$\mathbf{1506 \pm 552}$	$\mathbf{1442 \pm 588}$	830 ± 611	$\mathbf{2097 \pm 213}$	$\mathbf{2081 \pm 175}$	1408 ± 68
	1.4	379 ± 555	1224 ± 412	$\mathbf{1302 \pm 300}$	1187 ± 305	1534 ± 40	$\mathbf{1659 \pm 107}$
	1.5	257 ± 442	527 ± 727	780 ± 593	755 ± 542	1480 ± 646	1550 ± 92

Threshold Selection In our proposed method, we can control the magnitude of policy uniqueness flexibly by adjusting the constraint threshold r_0. Choosing different thresholds will lead to different policy behaviors. Concretely., a larger threshold may drive the agent to perform more differently while smaller threshold imposes a lighter constraint on the behavior of the agent. Intuitively, a larger threshold will lead to relatively poor performance for the learning algorithm is less likely to find a feasible solution to Eq.(7).

Besides, we do not use constraints in the form of Eq.(7) as we need not force every single action of a new agent to be different from others. Instead, we are more care about the long term differences. Therefore, we use the cumulative uniqueness as constraints,

$$\begin{aligned} \max_{\theta \in \Theta} \quad & \mathbb{E}_{\tau \sim \theta}[g_{\text{task}}], \\ s.t. \quad & \textstyle\sum_{t=0}^{t=\tau}(r_{\text{int},t} - r_0) \geq 0, \forall \tau = 1, 2, ..., T, \end{aligned}$$

We test our method with different choices of threshold values. The performance of agents under different thresholds are shown in Fig. 8 and more detailed analysis of their success rate is presented in Table. 2.

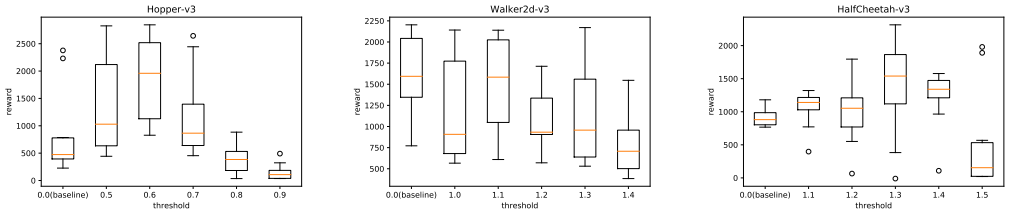

Figure 8: The performance our methods under different threshold selection, in the Hopper, Walker and HalfCheetah environments. the results are collected from 10 learned policies.

E More Qualitative Results

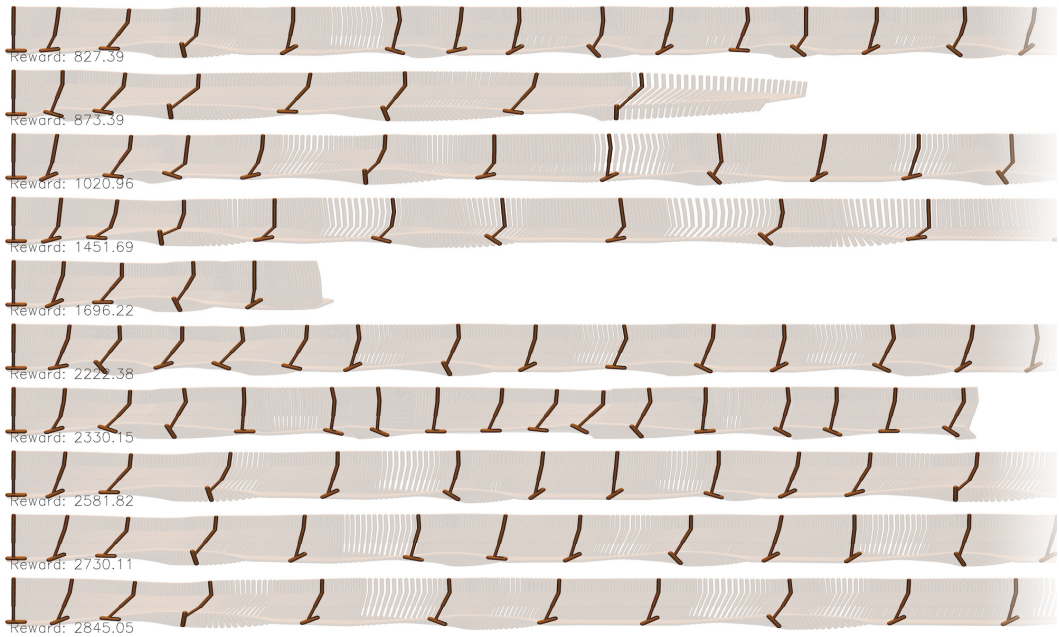

Figure 9: The visualization of policy behaviors of agents trained by our method in Hopper-v3 environment. Agents learn to jump with different strides.

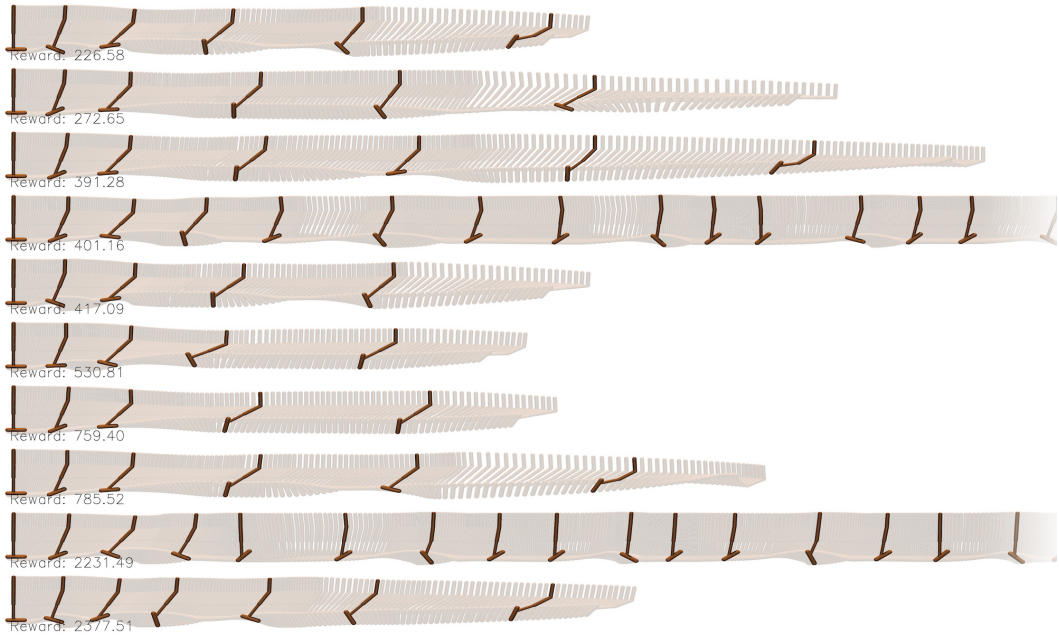

Figure 10: The visualization of policy behaviors of agents trained by PPO in Hopper-v3 environment. Most agents learn a policy that can be described as *Jump as far as possible and fall down*, leading to relative poor performance.

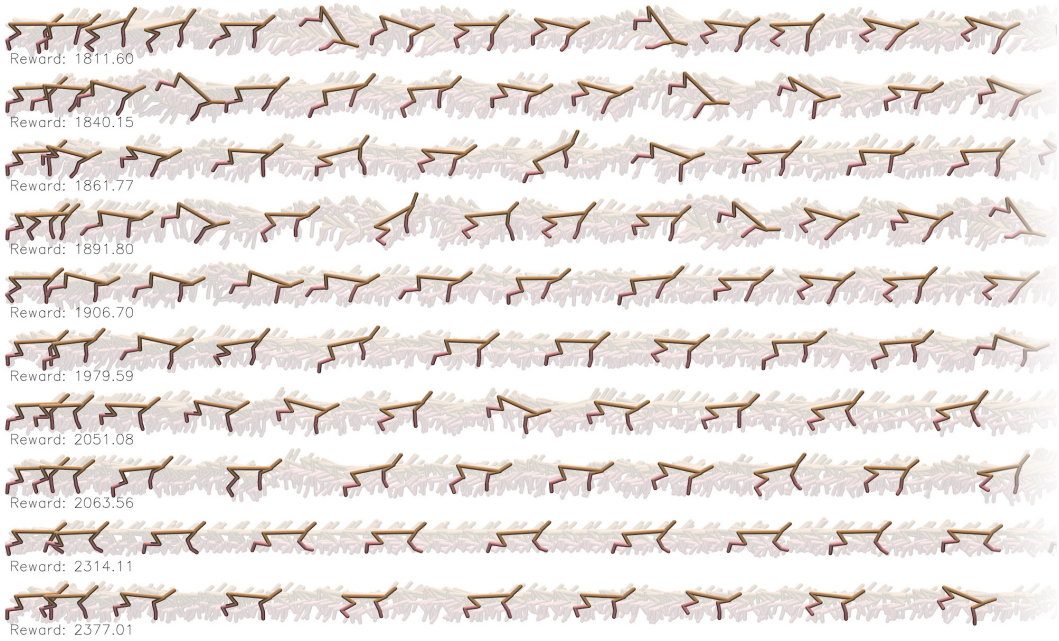

Figure 11: The visualization of policy behaviors of agents trained by our method in HalfCheetah-v3 environment. Our agents run much faster compared to PPO agents and at the mean time several patterns of motion have emerged.

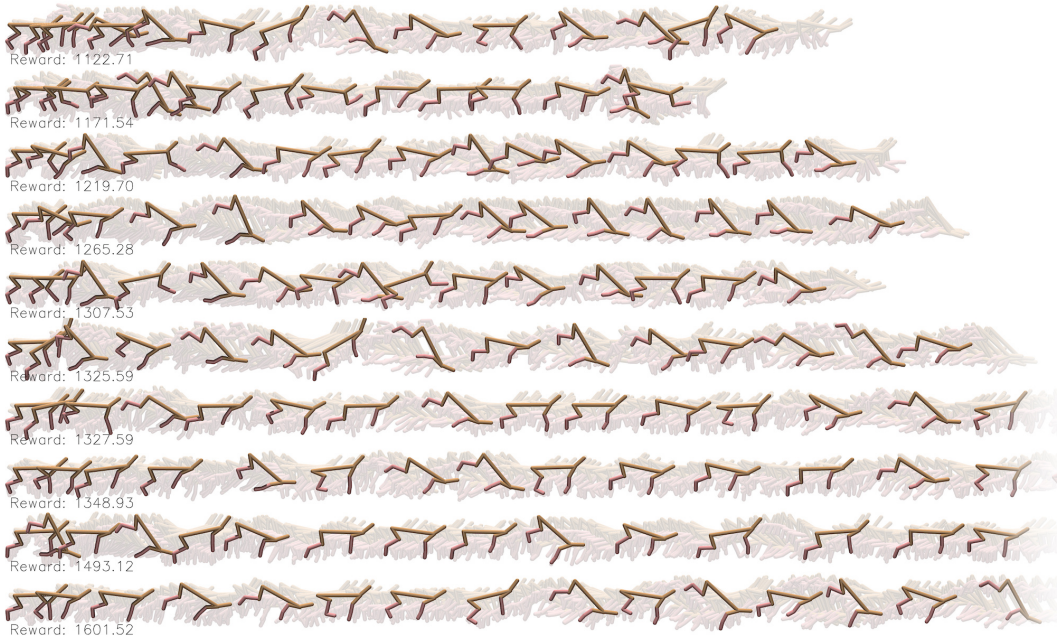

Figure 12: The visualization of policy behaviors of agents trained by PPO in HalfCheetah-v3 environment. Since we only draw fixed number of frames in each line, in the limited time steps the PPO agents can not run enough distance to leave the range of our drawing, which shows that our agents run much faster.

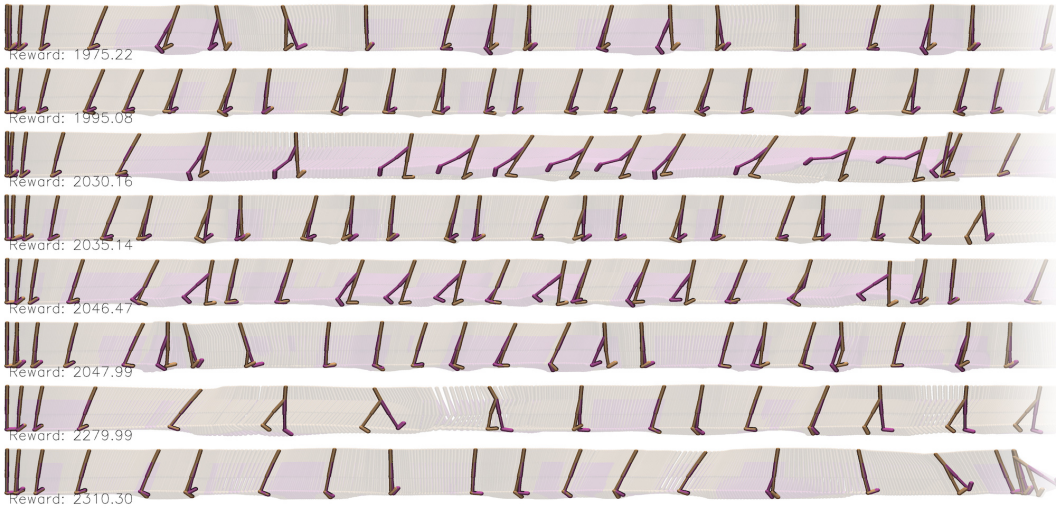

Figure 13: The visualization of policy behaviors of agents trained by our method in Walker2d-v3 environment. Instead of bouncing at the ground using both legs, our agents learns to use both legs to step forward.

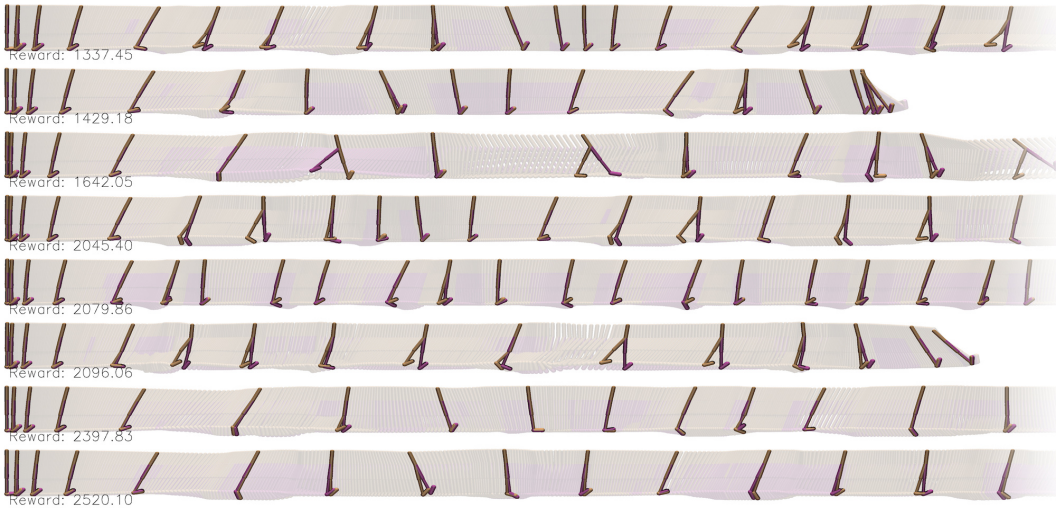

Figure 14: The visualization of policy behaviors of agents trained by PPO in Walker2d-v3 environment. Most of the PPO agents only learn to use both legs to support the body and jump forward.