Bridge the Gap between Deep RL and Human Learning

Hao SUN*

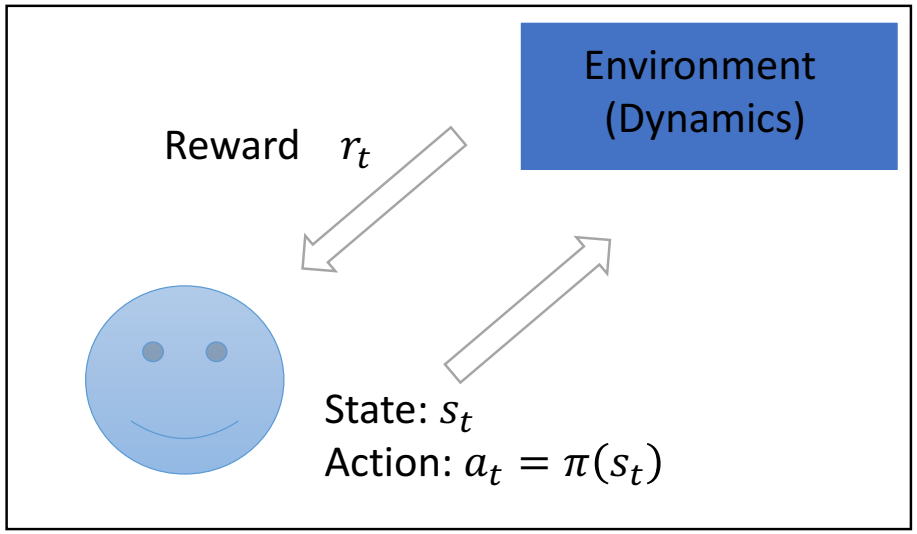
*Dept. Information Engineering @CUHK

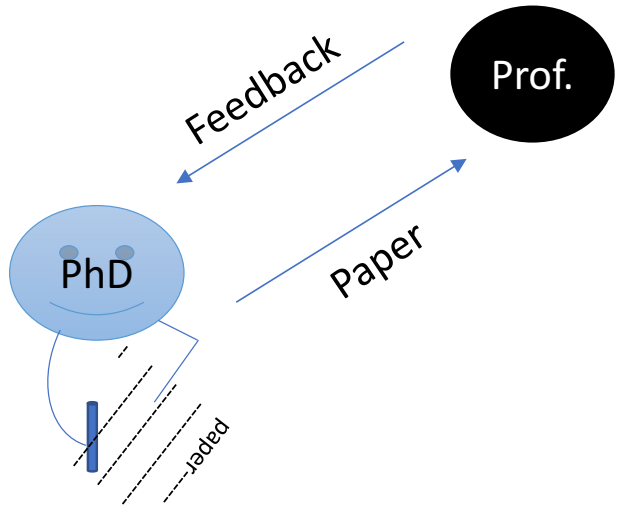
October 10, 2019

Content

Content

An Analogy of RL Learning Paradigm





Learning Objective:

$$Max_{\pi} \Sigma r_t$$

Value Based/ Policy Based/ Actor-Critic

Dense Reward: Once you have a commitment, you receive a feedback
Sparse Reward: Your professor is busy

Model Based/ Model Free

Markov Decision Process (MDP) Perspective

- An agent has its policy π
- For each time step, the agent know about its **state** information s_t, it should react by giving an **action** $a_t = \pi(s_t)$, which is always a distribution over the action space $\mathcal{A}$.
- The **environment** (dynamics) returns a reward r_t
- Learning Objective: $\max_\pi \mathbb{E}_\pi \sum_{t=0}^{T} \gamma^t r_t$
- Markovian Property: $P(s_{t+1}|a_t, s_t, a_{t-1}, s_{t-1}, ...) = P(s_{t+1}|a_t, s_t)$

SOTA (Prevailing) Algorithms

- On-Policy: PPO, TRPO, A3C, A2C (Locomotion)
- Off-Policy: TD3, DDPG (Locomotion, HalfCheetah)
- Off-Policy: DQN, C51, QRDQN, IQN (Atari)
- Extension: Curiosity, HER (For Reward Sparse Tasks), Distributed Training
- Auxiliary: DQNfD, DDPGfD, Inverse RL from Human Preference

Benchmark Tasks

- Games (Atari, SuperMarioBros, Doom): Discrete Action Space
- Robotics, Locomotion based on Mujoco: Continuous Action Space

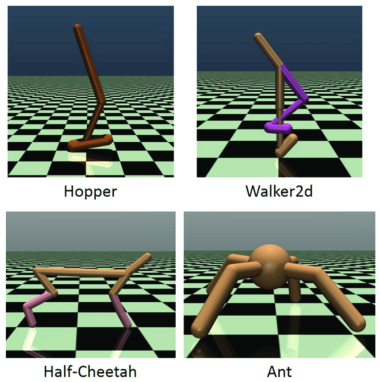


Figure: SuperMarioBro, HandReach and Locomotion Tasks

Content

Policy Continuation with Hindsight Inverse Dynamics

- Objective: Find an effective way to learn goal-oriented reward sparse task
- Challenge: When an agent always fail, it can hardly learn how to be success
- Our Key Insight: Self-imitation learning of human, Curriculum learning, Learn to be success from failures

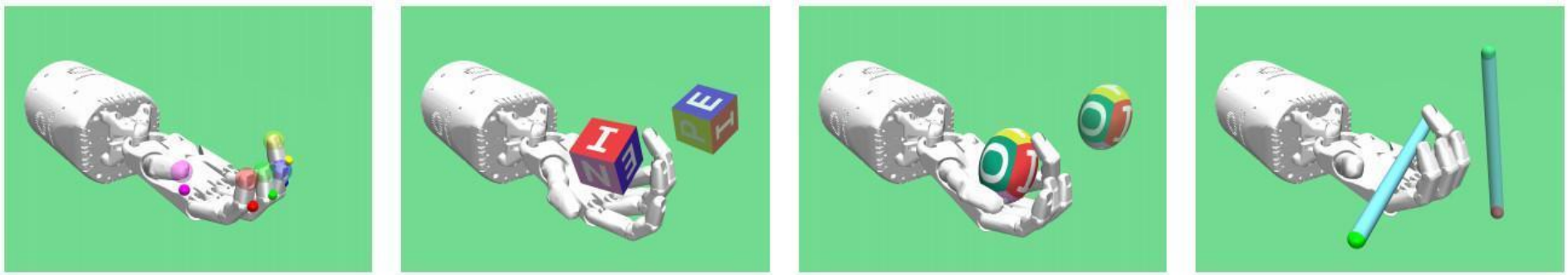


Figure: Hand-Reach, -Block, -Egg, -PenRotate Environment

Policy Continuation and k-step Sovability

Definition 2.1 (Definition 1: Policy Continuation(PC))

Suppose π is a policy function defined on a non-empty sub-state-space $\mathcal{S}_U$ of the state space $\mathcal{S}$, i.e., $\mathcal{S}_U \subset \mathcal{S}$. If $\mathcal{S}_V$ is a larger subset of $\mathcal{S}$, containing $\mathcal{S}_U$, i.e., $\mathcal{S}_U \subset \mathcal{S}_V$ and Π is a policy function defined on $\mathcal{S}_V$ such that

$$\Pi(s) = \pi(s) \qquad \forall s \in \mathcal{S}_U$$

then we call Π a policy continuation of π, or we can say the restriction of Π to $\mathcal{S}_U$ is the policy function π.

- It is clear that complex skills are continuations of simpler skills

Definition 2.2 (Definition 2: k-Step Solvability)

Given a state-goal pair (s, g) as a task of a certain system with deterministic dynamics, if reaching the goal g needs at least k steps under the optimal policy π^* starting from s, i.e., starting from $s_0 = s$ and execute $a_i = \pi^*(s_i, g)$ for $i = \{0, 1, ..., k-1\}$, the state $s_k = \mathcal{T}(s_{k-1}, a_{k-1})$ satisfies $m(s_k) = g$, we call the pair (s, g) has k-step solvability, or (s, g) is k-step solvable.

State Goal Space Partition

Definition 2.3 (Definition 3: Solvable State-Goal Space Partition)

Given a certain environment, any solvable state-goal pairs can be categorized into only one sub state-goal space by the following partition

$$\mathcal{S} \times \mathcal{G} \backslash (\mathcal{S} \times \mathcal{G})_U = \bigcup_{j=0}^{T} (\mathcal{S} \times \mathcal{G})_j \tag{1}$$

Definition 2.4 (Definition 4: Sub Policy on Sub Space)

π_i is a sub-policy defined on the sub-state-goal space $(\mathcal{S} \times \mathcal{G})_i$. We say π_i^* is an optimal sub-policy if it is able to solve all i-step solvable state-goal pair tasks in i steps.

Corollary 2.5

If $\{\pi_i^*\}$ is restricted as a policy continuation of $\{\pi_{i-1}^*\}$ for $\forall i \in \{1, 2, ...k\}$, π_i^* is able to solve any i-step solvable problem for $i \leq k$. By definition, the optimal policy π^* is a policy continuation of the sub policy π_T^*, and π_T^* is already a substitute for the optimal policy π^*.

Hindsight Inverse Dynamics

Inverse Dynamics:

$$\theta_1 = \arg\min_{\theta} \sum_{s_t, g, a_t} ||f_\theta((s_t, g), (s_{t+1}, g)) - a_t||^2 \tag{2}$$

Hindsight Inverse Dynamics:

$$\theta_1 = \arg\min_{\theta} \sum_{s_t, s_{t+1}, a_t} ||f_\theta((s_t, m(s_{t+1})), (s_{t+1}, m(s_{t+1}))) - a_t||^2 \tag{3}$$

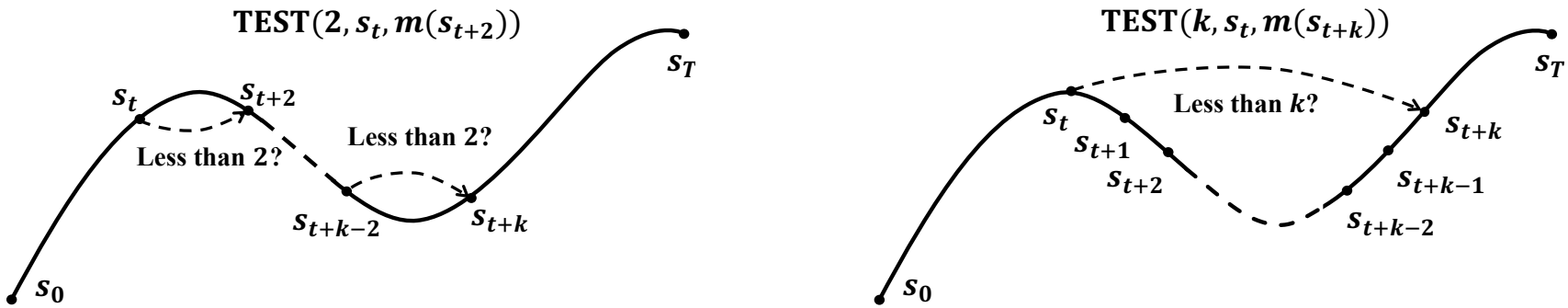
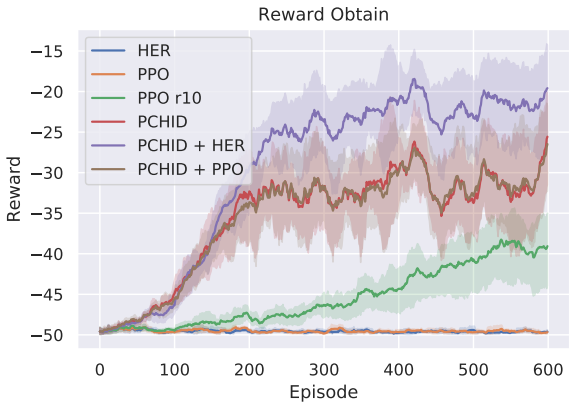


Figure: Test whether the transitions are 2-step (left) or k-step (right) solvable. The TEST function will return True if the transition $s_t \to s_{t+k}$ needs at least k steps.

Empirical Results



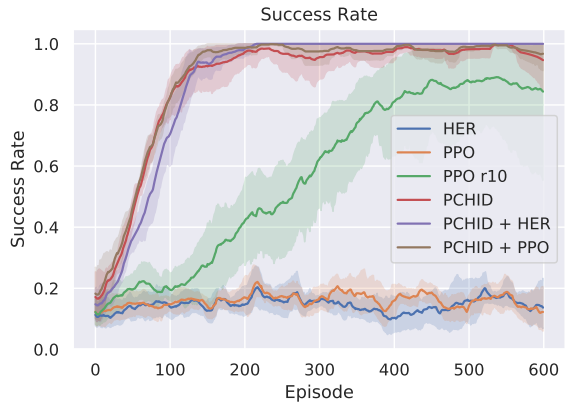
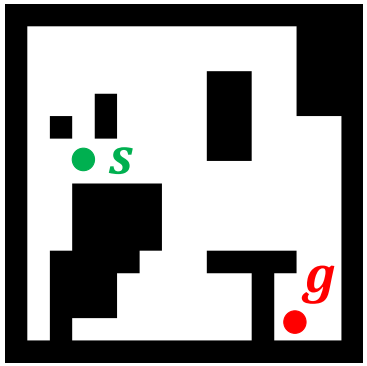


Figure: Experiments on the FetchReach Environment



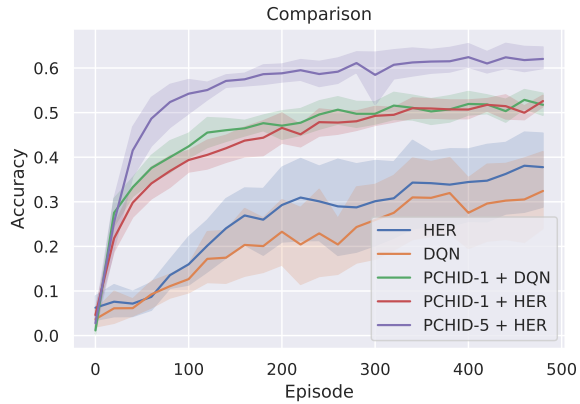


Figure: Experiments on the GridWorld Environment

Content

Policy Evolution with Hindsight Inverse Dynamics

- Motivation: Can we further improve the learning efficiency of PCHID?
- Question: Can we learn all the policies together? Or can the agent learn $1, 2, ..., k$-step transitions simultaneously?
- Yes, from another perspective other than "Policy Continuation"
- Why could 1-step PCHID improves the final performance? Action prediction extrapolation in **flat** state-action spaces!

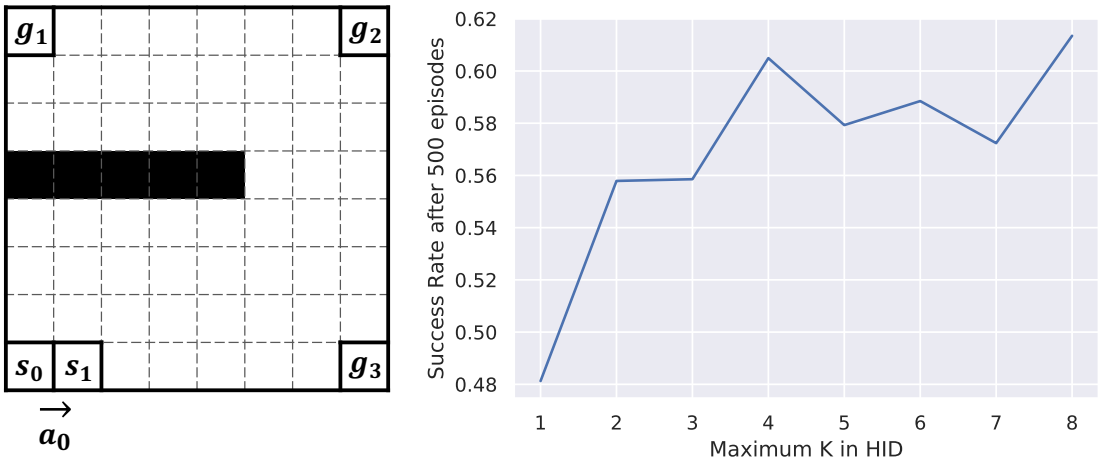


Figure: An simple navigation task; and ablation study results in the GridWorld domain

Ornstein-Uhlenbeck (OU) Process Perspective

An policy equipped with Gaussian noise in the action space $a \sim \mathcal{N}(\mu, \sigma^2)$ lead to a stochastic process in the state space. In the most simple case, the mapping between action space and the corresponding change in state space is an affine transformation, i.e., $\Delta s_t = s_{t+1} - s_t = \alpha a_t + \beta$.
Without loss of generality, we have

$$\Delta s_t \sim \mathcal{N}(\epsilon(g - s_t), \sigma^2) \tag{4}$$

where ϵ indicates the goal awareness of the agent. e.g., for random initialized policies, the actions are unaware of goal thus $\epsilon = 0$, and for optimal policies, the actions are goal-oriented thus $\epsilon = 1$. So we have

$$\mathrm{d}s_t = \epsilon(g - s_t)\mathrm{d}t + \sigma \mathrm{d}W_t \tag{5}$$

Which is a typical OU process.

Intuition of Policy Evolution

The closed form solution of the above OU-process is

$$s_t = s_0 e^{-\epsilon t} + g(1 - e^{-\epsilon t}) + \sigma \int_0^t e^{-\epsilon(t-s)} \mathrm{d}W_s \tag{6}$$

and the expectation is

$$\mathbb{E}(s_t) - g = (s_0 - g)e^{-\epsilon t} \tag{7}$$

Intuitively, Eq. (7) shows that as ϵ increase during learning, it will take less time to reach the goal. More precisely, we are caring about the concept of First Hitting Time (FHT) of OU process, i.e., $\tau = \inf\{t > 0 : s_t = g | s_0 > g\}$.

Calculation of FHT

Without loss of generality, we can normalize the Eq.(5) by the transformation:

$$\tilde{t} = \epsilon t, \quad \tilde{s} = \frac{\sqrt{2\epsilon}}{\sigma}(s-g), \quad \tilde{g} = \frac{\sqrt{2\epsilon}}{\sigma}(g-g) = 0, \quad \tilde{s_0} = \frac{\sqrt{2\epsilon}}{\sigma}(s_0-g) \tag{8}$$

and we consider the FHT problem of

$$\begin{aligned} \mathrm{d}\tilde{s_t} &= -\tilde{s_t}\mathrm{d}\tilde{t} + 2\mathrm{d}W_{\tilde{t}} \\ \tilde{\tau} &= \inf\{\tilde{t} > 0 : \tilde{s_t} = 0 | \tilde{s_0} > 0\} \end{aligned} \tag{9}$$

The probability density function of $\tilde{\tau}$, denoted by $p_{0,\tilde{s_0}}(\tilde{t})$ is

$$p_{0,\tilde{s_0}}(\tilde{t}) = \sqrt{\frac{2}{\pi}} \frac{\tilde{s_0} e^{-\tilde{t}}}{(1-e^{-2\tilde{t}})^{3/2}} \exp\left(\frac{\tilde{s_0}^2 e^{-2\tilde{t}}}{2(1-e^{-2\tilde{t}})}\right) \tag{10}$$

and the expectation is provided as

$$\mathbb{E}[\tilde{\tau}] = \sqrt{\frac{\pi}{2}} \int_{-\tilde{s_0}}^{0} \left(1 + \mathrm{erf}\left(\frac{\tilde{t}}{\sqrt{2}}\right)\right) \exp\left(\frac{\tilde{t}^2}{2}\right) \mathrm{d}\tilde{t} \tag{11}$$

Solution from FHT Perspective: Policy Evolution with HID

Accordingly, the optimization of solving goal-oriented reward sparse tasks can be viewed as minimizing the FHT of OU process. From this perspective, **any action that can reduce the FHT will lead to a better policy**.

1. Large σ in exploration
2. Large ϵ as possible
3. Learning all transitions not yet mastered

Here is a trade-off problem between (1) and (2), we choose to use linear decayed σ in practice.

Table: The successful rate of different methods in the FetchPush, FetchSlide and FetchPickAndPlace environments (trained for 1.25M timesteps)

Method	FetchPush	FetchSlide	FetchPickAndPlace
PPO	0.00	0.00	0.00
DDPG	0.08	0.03	0.05
DDPG + HER	**1.00**	0.30	0.60
PEHID	0.95	**0.38**	**0.75**

Content

Learning with Social Influences

- Motivation: Learning variant policies to solve a given primal task
- Previous Approaches: Optimization to a **heuristically** Novelty Reward directly
- Our Key Insight: People pursue their social uniqueness not as a goal, but a constraint.
- Main Contributions:
  1. define a metric space to measure distance between policies
  2. multi-objective optimization problem $\rightarrow$ constrained optimization problem
  3. result in both different and well-performed policies

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{\tau \sim \theta}[g_{\text{total}}] = \max_{\theta \in \Theta} \quad \mathbb{E}_{\tau \sim \theta}[\alpha \cdot g_{\text{task}} + (1-\alpha) \cdot g_{\text{int}}], \tag{12}$$

$$\begin{aligned} \max_{\theta \in \Theta} \quad & \mathbb{E}_{\tau \sim \theta}[g_{\text{task}}], \\ s.t. \quad & r_{\text{int},t} - r_0 \geq 0, \forall t = 1, 2, ..., T, \end{aligned} \tag{13}$$

Policy Metric Spaces

Theorem 4.1 (Metric Space $(\Theta, \overline{D}^{\rho}_{TV})$)

The expectation of $D_{TV}(\cdot,\cdot)$ of two policies over any state distribution $\rho(s)$:

$$\overline{D}^{\rho}_{TV}(\theta_i, \theta_j) := {}_{s\sim\rho(s)}[D_{TV}(\theta_i(s), \theta_j(s))], \tag{14}$$

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

Definition 4.2 (Uniqueness of Policy)

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

$$\mathrm{U}(\theta|\Theta_{\text{ref}}) := \min_{\theta_j \in \Theta_{\text{ref}}} \overline{D}^{\rho}_{TV}(\theta, \theta_j). \tag{15}$$

Interpretation from Optimization Perspective

Theorem 4.3 (Unbiased Single Trajectory Estimation)

The estimation of $\rho_\theta(s)$ using a single trajectory τ is unbiased.

We note here, the WSR, TNB and ours Interior Policy Differentiation (IPD) methods correspond to three approaches in constrained optimization problems. For simplicity, we consider Eq.(13) with a simpler constraint $g_{\text{int}} - g_0 \geq 0$, where $g_{\text{int}} = \sum_{t=0}^{T} r_{\text{int},t}$, i.e.,

$$\begin{aligned} \max_{\theta \in \Theta} \quad & f(\theta) = \mathbb{E}_{\tau \sim \theta}[g_{\text{task}}] \\ s.t. \quad & g(\theta) = g_{\text{int}} - g_0 \geq 0 \end{aligned}$$

WSR: Penalty Method

The Penalty Method considers the constraints of Eq.(13) by putting constraint $g(\theta)$ into a penalty term, and then solve the unconstrained problem

$$\max_{\theta \in \Theta} \quad f(\theta) + \frac{1-\alpha}{\alpha} \min\{g(\theta), 0\} \tag{16}$$

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

$$g(\bar{\theta}+\lambda\vec{p}) = g(\bar{\theta}) + \nabla_\theta g(\bar{\theta})^{\mathrm{T}}\lambda\vec{p} + O(||\lambda\vec{p}||) \tag{17}$$

The Feasible Direction Method (FDM) considers the constraints of Eq.(13) by first finding a direction $\vec{p}$ satisfies

$$\begin{aligned} &\nabla_\theta f^{\mathrm{T}} \cdot \vec{p} > 0 \\ &\nabla_\theta g^{\mathrm{T}} \cdot \vec{p} > 0 \quad \text{if} \quad g = 0 \end{aligned} \tag{18}$$

so that for small λ, we have

$$g(\bar{\theta}+\lambda\vec{p}) = g(\bar{\theta}) + \lambda\nabla_\theta g(\bar{\theta})^{\mathrm{T}}\vec{p} > g(\bar{\theta}) = 0 \quad \text{if} \quad \mathrm{g}(\bar{\theta}) = 0 \tag{19}$$

and

$$g(\bar{\theta}+\lambda\vec{p}) = g(\bar{\theta}) + \lambda\nabla_\theta g(\bar{\theta})^{\mathrm{T}}\vec{p} > 0 \quad \text{if} \quad g(\bar{\theta}) > 0 \tag{20}$$

TNB (Cont.)

In order to find such feasible direction $\vec{p}$, we can introduce a parameter η in Eq.(18), such that

$$\begin{cases} \nabla_\theta f^{\mathrm{T}} \cdot \vec{p} > \eta \\ \nabla_\theta g^{\mathrm{T}} \cdot \vec{p} > \eta \quad \text{if} \quad g = 0 \\ \eta \geq 0 \end{cases} \tag{21}$$

As Eq.(21) can be reformed as a linear programming (LP) problem over η, i.e.,

$$\max \eta$$
$$s.t. \begin{cases} \nabla_\theta f^{\mathrm{T}} \cdot \vec{p} > \eta \\ \nabla_\theta g^{\mathrm{T}} \cdot \vec{p} > \eta \quad \text{if} \quad g = 0 \\ -1 \leq d_j \leq 1, j = 1, 2, ..., n \end{cases} \tag{22}$$

where $\vec{p} = (d_1, d_2, ..., d_n)^{\mathrm{T}}$, and the last constraint prevent $|\vec{p}|$ from going to $\to \infty$ (as we only care about $\vec{p}$ as a direction). Denote the optimal solution of Eq.(22) as $(\eta^*, \vec{p}^*)$. We can then step to the direction of $\vec{p}$ after a line search for the optimal stride λ^*. The Topkis-Veinott method provides improves FDM by using another auxiliary problem.

TNB (Cont.)

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

$$\vec{p} = \begin{cases} \nabla_\theta f + \frac{|\nabla_\theta f|}{|\nabla_\theta g|} \nabla_\theta g \cdot \cos\left(\nabla_\theta f, \nabla_\theta g\right) & \text{if } \cos\left(\nabla_\theta f, \nabla_\theta g\right) \leq 0 \\ \nabla_\theta f + \frac{|\nabla_\theta f|}{|\nabla_\theta g|} \nabla_\theta g & \text{if } \cos\left(\nabla_\theta f, \nabla_\theta g\right) > 0 \end{cases} \tag{23}$$

Clearly, Eq.(23) satisfies Eq.(18). As the stride of TNB is chosen as $\frac{|\nabla_\theta f| + |\nabla_\theta g|}{2}$, and the $\nabla_\theta g$ term always exists during optimization, problem arises when $\nabla_\theta f \to 0$, therefore 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.

Interior Policy Differentiation (IPD): Interior Method

in this work we propose to solve the constrained optimization problem Eq.(13) by resembling the Interior Point Methods (IPMs). In vanilla IPMs, the constrained optimization problem in Eq.(13) is solved by reforming it to an unconstrained form with an additional barrier term in the objective as

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{\tau \sim \theta}[g_{\text{task}} + \sum_{t=0}^{T} \alpha \log\left(r_{\text{int},t} - r_0\right)]. \tag{24}$$

The limit of Eq.(24) when $\alpha \to 0$ then leads to the solution of Eq.(13). However, directly applying this solution 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.

Empirical Results

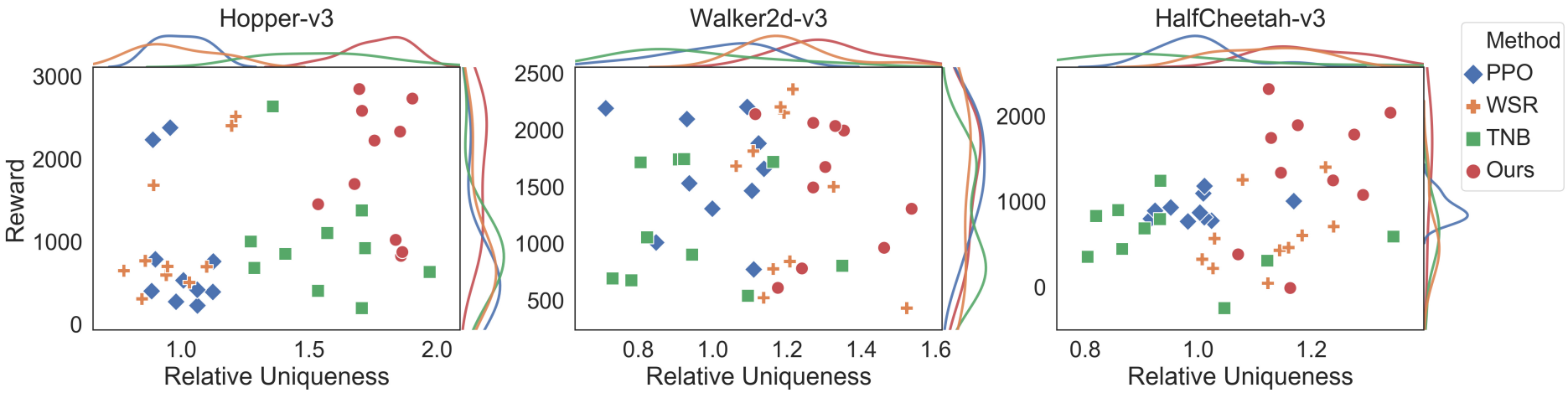


Figure: The comparison between Uniqueness and Performance in Hopper-v3, Walker2d-v3 and HalfCheetah-v3 environments. The value of uniqueness is normalized to relative uniqueness by regarding the averaged uniqueness of PPO policies as the baseline.

Accepted Works

- Policy Continuation with Hindsight Inverse Dynamics (NeurIPS'19)
- Policy Continuation and Policy Evolution with Hindsight Inverse Dynamics (NeurIPS'19 OptRL Workshop)
- Learning with Identity and Uniqueness through Social Constraints (NeurIPS'19 DeepRL Workshop)

Thanks!