Hindsight Trust Region Policy Optimization

Hanbo Zhang[1] , **Site Bai**[1] , **Xuguang Lan**[1*] , **David Hsu**[2] and **Nanning Zheng**[1]
[1]Xi'an Jiaotong University
[2]National University of Singapore
{zhanghanbo163, best99317}@stu.xjtu.edu.cn, xglan@xjtu.edu.cn, dyhsu@comp.nus.edu.sg, nnzheng@xjtu.edu.cn

Abstract

Reinforcement Learning (RL) with sparse rewards is a major challenge. We propose Hindsight Trust Region Policy Optimization (HTRPO), a new RL algorithm that extends the highly successful TRPO algorithm with hindsight to tackle the challenge of sparse rewards. Hindsight refers to the algorithm's ability to learn from information across goals, including past goals not intended for the current task. We derive the hindsight form of TRPO, together with QKL, a quadratic approximation to the KL divergence constraint on the trust region. QKL reduces variance in KL divergence estimation and improves stability in policy updates. We show that HTRPO has similar convergence property as TRPO. We also present Hindsight Goal Filtering (HGF), which further improves the learning performance for suitable tasks. HTRPO has been evaluated on various sparse-reward tasks, including Atari games and simulated robot control. Results show that HTRPO consistently outperforms TRPO, as well as HPG, a state-of-the-art policy gradient algorithm for RL with sparse rewards.[1]

1 Introduction

Reinforcement Learning (RL) has been widely investigated to solve problems from complex strategic games [Mnih *et al.*, 2015] to precise robotic control [Deisenroth *et al.*, 2013]. However, current successful practice of RL in robotics relies heavily on careful and arduous reward shaping[Ng *et al.*, 1999; Grzes, 2017]. **Sparse reward**, in which the agent is rewarded only upon reaching the desired goal, obviates designing a delicate reward mechanism. It also guarantees that the agent focuses on the intended task itself without any deviation. However, sparse reward diminishes the chance for policy to converge, especially in the initial random exploration stage, since the agent can hardly get positive feedbacks.

Recently, several works have been devoted to sparse-reward RL. [Andrychowicz *et al.*, 2017] proposes Hindsight Experience Replay(HER), which trains the agent with hindsight goals generated from the achieved states through the historical interactions. Such hindsight experience substantially alleviates exploration problem caused by sparse-reward settings. [Rauber *et al.*, 2019] proposes Hindsight Policy Gradient (HPG). It introduces hindsight to policy gradient, resulting in an advanced algorithm for RL with sparse reward. However, for HPG, there remain several drawbacks hindering its application in more cases. Firstly, as an extension to "vanilla" policy gradient, its performance level and sample efficiency remain limited. Secondly, it inherits the intrinsic high variance of PG methods, and the combination with hindsight data further exacerbates the learning stability.

In this paper, we propose Hindsight Trust Region Policy Optimization (HTRPO), a hindsight form of TRPO [Schulman *et al.*, 2015b], which is an advanced RL algorithm with approximately monotonic policy improvements. We prove that HTRPO theoretically inherits the convergence property of TRPO, and significantly reduces the variance of policy improvement by introducing Quadratic KL divergence Estimation (QKL) approach. Moreover, to select hindsight goals that better assist the agent to reach the original goals, we design a Hindsight Goal Filtering mechanism.

We demonstrate that in a wide variety of sparse-reward tasks including Atari games and robotic control, HTRPO can consistently outperform TRPO and HPG in both performance and sample efficiency with commendable learning stability. We also provide a comprehensive comparison with HER, showing that HTRPO achieves much better performance in 6 out of 7 benchmarks. Besides, we also conduct ablation studies to show that Quadratic KL divergence Estimation can effectively lower the variance and constrain the divergence while Hindsight Goal Filtering brings the performance to a higher level especially in more challenging tasks.

2 Preliminaries

2.1 RL Formulation and Notation

Consider the standard infinite-horizon reinforcement learning formulation which can be defined by tuple $(\mathcal{S}, \mathcal{A}, \pi, \rho_0, r, \gamma)$. $\mathcal{S}$ and $\mathcal{A}$ denote the set of states and actions respectively. $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is a policy mapping states to a distribution over actions. ρ_0 is the distribution of the initial state s_0. Reward function $r : \mathcal{S} \to \mathbb{R}$ defines the reward obtained from

[*] Corresponding author.

[1] All detailed proofs and additional experiments can be found in the *supplementary materials*.

the environment and $\gamma \in (0, 1)$ is a discount factor. In this paper, the policy is a differentiable function regarding parameter θ. We follow the standard formalism of state-action value function $Q(s, a)$, state value function $V(s)$ and advantage function $A(s, a)$ in [Sutton and Barto, 2018]. We also adopt the definition of γ-discounted state visitation distribution as $\rho_\theta(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s)$ [Ho *et al.*, 2016]. Correspondingly, γ-discounted state-action visitation distribution [Ho *et al.*, 2016], also known as occupancy measure [Ho and Ermon, 2016], is defined as $\rho_\theta(s, a) = \rho_\theta(s) \times \pi_\theta(a|s)$.

2.2 Trust Region Policy Optimization

TRPO [Schulman *et al.*, 2015a] is an iterative trust region method that effectively optimizes policy by maximizing the per-iteration policy improvement. The optimization problem proposed in TRPO can be formalized as follows:

$$\max_{\theta} \mathop{\mathbb{E}}_{s,a\sim\rho_{\tilde{\theta}}(s,a)} \left[\frac{\pi_\theta(a|s)}{\pi_{\tilde{\theta}}(a|s)} A_{\tilde{\theta}}(s, a) \right] \tag{1}$$

$$s.t. \mathop{\mathbb{E}}_{s\sim\rho_{\tilde{\theta}}(s)} \left[D_{KL}(\pi_{\tilde{\theta}}(a|s)||\pi_\theta(a|s)) \right] \leq \epsilon \tag{2}$$

in which $\rho_{\tilde{\theta}}(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s)$. θ denotes the parameter of the new policy while $\tilde{\theta}$ is that of the old one.

2.3 Hindsight Policy Gradient

HPG [Rauber *et al.*, 2019] combines the idea of hindsight [Andrychowicz *et al.*, 2017] with policy gradient methods. Though goal-conditioned reinforcement learning has been explored for a long time and actively investigated in recent works [Peters and Schaal, 2008; Schaul *et al.*, 2015; Veeriah *et al.*, 2018], HPG firstly extends the idea of hindsight to goal-conditioned policy gradient and shows that the policy gradient can be computed in expectation over all goals. The goal-conditioned policy gradient is derived as follows:

$$\nabla_\theta \eta(\pi_\theta) = \mathop{\mathbb{E}}_{g,\tau} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t, g) A_\theta(s_t, a_t, g) \right] \tag{3}$$

where $\tau \sim p_\theta(\tau|g)$. Then, by applying hindsight formulation, it rewrites goal-conditioned policy gradient with trajectories conditioned on achieved goal g' using importance sampling to solve sparse-reward problems efficiently.

3 Hindsight Trust Region Policy Optimization

In this section, we firstly introduce Quadratic KL divergence Estimation (QKL) method, which efficiently reduces the variance of KL estimation in TRPO and results in higher learning stability. With QKL, we show that TRPO maintains the monotonically-converging property. After that, we derive the hindsight form of TRPO, called Hindsight Trust Region Policy Optimization algorithm, to tackle the severely off-policy hindsight data for better learning with sparse rewards. Specifically, the expected return and the KL divergence constraint are both modified to adapt to hindsight data with importance sampling. Benefiting from QKL, we can precisely estimate KL divergence using hindsight data while keeping the variance below a reasonable level. Intuitively, HTRPO utilizes hindsight data to estimate the objective and the constraint, and iteratively find out the local optimal policy to ensure the approximately monotonous policy improvements.

3.1 TRPO with Quadratic KL Divergence

In TRPO, the KL divergence expectation under $\rho_{\tilde{\theta}}(s)$ is estimated by averaging values of KL divergence conditioned on collected states. However, this method is no longer valid if KL divergence cannot be analytically computed (e.g. Gaussian Mixture Model) or the state distribution changes (e.g. using hindsight data instead of the collected ones). To solve this problem, we firstly transform the KL divergence to an expectation under occupancy measure $\rho_{\tilde{\theta}}(s, a) = \rho_{\tilde{\theta}}(s) \times \pi_{\tilde{\theta}}(a|s)$. It can be estimated using the collected state-action pairs (s, a), no longer depending on the analytical form of KL divergence. Also, such formulation is convenient for correcting changed distribution over state and action by importance sampling, which will be discussed in section 3.2. However, it will increase the estimation variance, causing instability of training. Therefore, by making use of another f-divergence, we propose QKL to approximate KL divergence for variance reduction, and both theoretically and practically, we prove the effectiveness of such an approximation.

Given two policies $\pi_{\tilde{\theta}}(a|s)$ and $\pi_\theta(a|s)$, the KL-divergence over state s can be converted to a logarithmic form:

$$D_{KL}(\pi_{\tilde{\theta}}(a|s)||\pi_\theta(a|s)) = \mathop{\mathbb{E}}_{a\sim\pi_{\tilde{\theta}}(a|s)} \left[\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s) \right]$$

However, simply expanding the KL-divergence into logarithmic form still leaves several problems unhandled. Firstly, such formulation causes excessively high estimation variance. Secondly, such estimation of KL-divergence is of possible negativity. To overcome these two drawbacks, we propose Quadratic KL Divergence Estimation in Proposition 1 and prove that such approximation will reduce the estimation variance in Proposition 2 (detailed proof can be found in Appendix A.1 and A.2):

Proposition 1. *(Quadratic KL Divergence Estimation). For policy $\pi_{\tilde{\theta}}(a|s)$ and $\pi_\theta(a|s)$, and for $\eta = \pi_\theta(a|s) - \pi_{\tilde{\theta}}(a|s)$,*

$$\begin{aligned} &\mathop{\mathbb{E}}_{a} \left[\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s) \right] \\ &\quad = \mathop{\mathbb{E}}_{a} \left[\frac{1}{2} (\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s))^2 \right] + \mathop{\mathbb{E}}_{a} \left[O(\eta^3) \right] \end{aligned} \tag{4}$$

where $a \sim \pi_{\tilde{\theta}}(a|s)$.

Proposition 1 demonstrates that when θ and $\tilde{\theta}$ is of limited difference, the expectation of $\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s)$ can be sufficiently estimated by the expectation of its square. In fact, $\mathbb{E}_{a\sim\pi_{\tilde{\theta}}(a|s)} \left[\frac{1}{2} (\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s))^2 \right]$ is an f-divergence, where $f(x) = \frac{1}{2} x (\log x)^2$, which we call D_{QKL} in this paper. Noticeably, though $f(x)$ is a convex function only when $x \in (\frac{1}{e}, \infty)$, and it indeed does not correspond to an f-divergence, in our practice, $\frac{\pi_{\tilde{\theta}}(a|s)}{\pi_\theta(a|s)} > \frac{1}{e}$ holds, hence we can define a convex function on R^+: $f(x) = \frac{1}{2} x (\log x)^2$ when $x \in (\frac{1}{e}, \infty)$ and $-x + \frac{2}{e}$ when $x \in (0, \frac{1}{e}]$, with an unused piece defined over $(0, \frac{1}{e}]$.

Proposition 2. *(Variance of Constraint Function). For policy $\pi_{\tilde{\theta}}(a|s)$ and $\pi_\theta(a|s)$, let* Var *denote the variance of a variable. For any action $a \in \mathcal{A}$ and any state $s \in \mathcal{S}$, when*

$\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s) \in [-0.5, 0.5]$, *then*

$$\begin{aligned} &\operatorname*{Var}_{a\sim\pi_{\tilde{\theta}}(a|s)}\left[\frac{(\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s))^2}{2}\right] \\ &\le \operatorname*{Var}_{a\sim\pi_{\tilde{\theta}}(a|s)}\left[\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s)\right]. \qquad (5) \end{aligned}$$

Proposition 2 illustrates that there is a decrease from the variance of $\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s)$ to the variance of its square. In fact, the closer it is between $\tilde{\theta}$ and θ, the more the variance decreases. Next, we will show that with the introduction of QKL, TRPO still maintains similar convergence property.

Proposition 3. *(Policy Improvement Guarantee) Given two policies π_θ and $\pi_{\tilde{\theta}}$, Let*

$$s^* = \arg\max_s D_{TV}(\pi_{\tilde{\theta}}(a|s), \pi_\theta(a|s))$$

If $\frac{D_{KL}(\pi_{\tilde{\theta}}(a|s^*), \pi_\theta(a|s^*))}{D_{QKL}(\pi_{\tilde{\theta}}(a|s^*), \pi_\theta(a|s^*))} \le \frac{2}{\ln 2}$, *then*

$$\eta(\pi_\theta) \ge L_{\pi_{\tilde{\theta}}}(\pi_\theta) - C D_{QKL}^{max}(\pi_{\tilde{\theta}}(a|s), \pi_\theta(a|s)) \qquad (6)$$

where

$$L_{\pi_{\tilde{\theta}}}(\pi_\theta) = \eta(\pi_{\tilde{\theta}}) + E_{s\sim\pi_{\tilde{\theta}}(s), a\sim\pi_\theta(a|s)}[A_{\pi_{\tilde{\theta}}}(s,a)] \qquad (7)$$

and $\eta(\pi_\theta) = E\left[\sum \gamma^t r_t\right]$ *is the expected return,* $C = \frac{4\beta\gamma}{(1-\gamma)^2}$, $\beta = max_{s,a}|A_{\pi_{\tilde{\theta}}}(s,a)|$, $D_{TV}(p,q) = \frac{1}{2}\sum_i |p_i - q_i|$.

The proof and detailed analysis are given in Appendix B. Intuitively, Proposition 3 means that when two policies are not far from each other, the convergence property of TRPO also holds for the QKL constraint. As a result, with Proposition 3, we can derive a new but similar monotonically-converging algorithm as in TRPO, given in Appendix B.2. By taking a series of approximation as shown in Appendix B.2, the following policy optimization problem is derived, called QKL-TRPO.

$$\max_\theta \mathop{\mathbb{E}}_{s,a\sim\rho_{\tilde{\theta}}(s,a)}\left[\frac{\pi_\theta(a|s)}{\pi_{\tilde{\theta}}(a|s)} A_{\tilde{\theta}}(s,a)\right] \qquad (1)$$

$$s.t. \mathop{\mathbb{E}}_{s,a}\left[\frac{1}{2}(\log \pi_{\tilde{\theta}}(a|s) - \log \pi_\theta(a|s))^2\right] \le \epsilon \qquad (8)$$

It is noteworthy that QKL-TRPO can be applied to policies which do not correspond to an analytic KL divergence (e.g. GMM policies). We also provide a simple analysis of QKL-TRPO compared with the original TRPO in Appendix G.1, which shows that QKL-TRPO is comparable with TRPO in a series of MuJoCo benchmarks.

3.2 Hindsight Formulation of QKL-TRPO

In this section, we derive the hindsight form of the QKL-TRPO, called Hindsight Trust Region Policy Optimization (HTRPO), to efficiently tackle severely off-policy hindsight experience and sparse-reward RL problems.

Starting from eq.1, it can be written in the following variant form:

$$L_{\tilde{\theta}}(\theta) = \mathop{\mathbb{E}}_{\tau\sim p_{\tilde{\theta}}(\tau)}\left[\sum_{t=0}^{\infty} \gamma^t \frac{\pi_\theta(a_t|s_t)}{\pi_{\tilde{\theta}}(a_t|s_t)} A_{\tilde{\theta}}(s_t, a_t)\right] \qquad (9)$$

The derivation process of this variant form is shown explicitly in Appendix C.1 and in [Schulman *et al.*, 2015a]. Given the expression above, similar to eq.3, we consider the goal-conditioned objective function:

$$L_{\tilde{\theta}}(\theta) = \mathop{\mathbb{E}}_{g,\tau}\left[\sum_{t=0}^{\infty} \gamma^t \frac{\pi_\theta(a_t|s_t,g)}{\pi_{\tilde{\theta}}(a_t|s_t,g)} A_{\tilde{\theta}}(s_t, a_t, g)\right] \qquad (10)$$

where $\tau \sim p_{\tilde{\theta}}(\tau|g)$. Though it seems that eq.10 makes it possible for off-policy learning, it can be used as the objective only when policy π_θ is close to the old policy $\pi_{\tilde{\theta}}$, i.e. within the trust region. Using severely off-policy data like hindsight experience will make the learning process diverge. Therefore, importance sampling is integrated to correct the difference of the trajectory distribution caused by changing the goal. Based on eq.10, the following Proposition gives out the hindsight objective function conditioned on some goal g' with the distribution correction derived from importance sampling.

Proposition 4. *(Hindsight Expected Return). For the original goal g and hindsight goal g', the object function of HTRPO $L_{\tilde{\theta}}(\theta)$ is given by:*

$$L_{\tilde{\theta}}(\theta) = \mathop{\mathbb{E}}_{g',\tau}\left[\sum_{t=0}^{\infty} \prod_{k=0}^{t} \frac{\pi_{\tilde{\theta}}(a_k|s_k,g')}{\pi_{\tilde{\theta}}(a_k|s_k,g)} \gamma^t \frac{\pi_\theta(a_t|s_t,g')}{\pi_{\tilde{\theta}}(a_t|s_t,g')} A_{\tilde{\theta}}(s_t, a_t, g')\right] \qquad (11)$$

in which $\tau \sim p_\theta(\tau|g)$ *and* $\tau = s_0, a_0, s_1, a_1, ..., s_t, a_t$.

Appendix C.2 presents an explicit proof of how the hindsight-form objective function derives from eq.10. In our practice, we introduce a baseline $V_\theta(s)$ for computing the advantage A_θ. Though A_θ here can be estimated by combining per-decision return [Precup *et al.*, 2000], due to its high variance, we adopt one-step TD method instead to get A_θ, i.e., $A_\theta(s,a) = r(s,a) + \gamma V_\theta(s') - V_\theta(s)$. Intuitively, eq.11 provides a way to compute the expected return in terms of the advantage with new-goal-conditioned hindsight experiences which are generated from interactions directed by old goals.

Next, we demonstrate that hindsight can also be introduced to the constraint function. The proof follows the methodology similar to that in Proposition 4, and is deducted explicitly in Appendix C.3.

Proposition 5. *(HTRPO Constraint Function). For the original goal g and hindsight goal g', the constraint between policy $\pi_{\tilde{\theta}}(a|s)$ and policy $\pi_\theta(a|s)$ is given by:*

$$\mathop{\mathbb{E}}_{g'}\left[\mathop{\mathbb{E}}_{\tau\sim p_\theta(\tau|g)}\left[\sum_{t=0}^{\infty} \prod_{k=0}^{t} \frac{\pi_{\tilde{\theta}}(a_k|s_k,g')}{\pi_{\tilde{\theta}}(a_k|s_k,g)} \gamma^t K_t\right]\right] \le \epsilon' \qquad (12)$$

in which $\epsilon' = \frac{\epsilon}{1-\gamma}$, *and* $K_t = \frac{1}{2}(\log \pi_{\tilde{\theta}}(a_t|s_t,g') - \log \pi_\theta(a_t|s_t,g'))^2$.

Proposition 5 implies the practicality of using hindsight data under condition g' to estimate the KL expectation. From all illustration above, we give out the final form of the optimization problem for HTRPO:

$$\max_\theta \mathop{\mathbb{E}}_{g'}\left[\mathop{\mathbb{E}}_{\tau\sim p_\theta(\tau|g)}\left[\sum_{t=0}^{\infty} \prod_{k=0}^{t} \frac{\pi_{\tilde{\theta}}(a_k|s_k,g')}{\pi_{\tilde{\theta}}(a_k|s_k,g)} \gamma^t R_t\right]\right] \qquad (13)$$

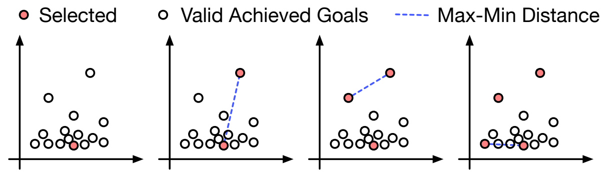


Figure 1: Procedure of Hindsight Goal Filtering

$$s.t.\ \mathop{\mathbb{E}}_{g'}\left[\mathop{\mathbb{E}}_{\tau\sim p_\theta(\tau|g)}\left[\sum_{t=0}^{\infty}\prod_{k=0}^{t}\frac{\pi_{\tilde{\theta}}(a_k|s_k,g')}{\pi_{\tilde{\theta}}(a_k|s_k,g)}\gamma^t K_t\right]\right]\leq\epsilon' \quad (14)$$

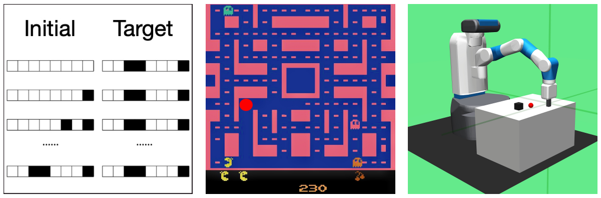
where $R_t = \frac{\pi_\theta(a_t|s_t,g')}{\pi_{\tilde{\theta}}(a_t|s_t,g')}A_{\tilde{\theta}}(s_t,a_t,g')$ and $K_t = \frac{1}{2}(\log\pi_{\tilde{\theta}}(a_t|s_t,g') - \log\pi_\theta(a_t|s_t,g'))^2$. The solving process for HTRPO optimization problem is explicitly demonstrated in Appendix D.

4 Hindsight Goal Filtering

In hindsight learning, the agent generalizes to reaching the original goal through learning to reach the hindsight goal first. Therefore, the selection of hindsight goals imposes a direct impact on the performance. If the hindsight goals are far from the original ones, the learned policy may not generalize well to the original goals. For example, in Fetch PickAndPlace, the initialized random policy barely grasps the target successfully, which results in the hindsight goals majorly distributing on the table. Given the original goals up in the air, such a discrepancy can cause a lower learning efficiency.

In this section, we introduce a heuristic method called Hindsight Goal Filtering(HGF). Intuitively, HGF is trying to filter the most useful goals from the achieved ones instead of random selection. Specifically, based on our analysis (eq. 13), the performance improves if we reduce the distribution discrepancy between original goals g and hindsight goals g'. Ideally, if the distribution of g' matches that of g, the agent will reach g after learning to reach g'. Therefore, we restrict the selected hindsight goals to distribute in the original goal space whenever possible to cover the area of original goals.

The main idea is shown in Figure 1 and the algorithm is summarized in Appendix E.1. The input of HGF includes 2 parts: the achieved goal set $\mathcal{G}_a$ and the original goal set $\mathcal{G}_o$. At the beginning, especially for some complex tasks, $\mathcal{G}_a$ can only have small or even no overlap with $\mathcal{G}_o$. Under this situation, we encourage the agent to learn to reach the original goal region by selecting the nearest achieved goals as the hindsight goals. Once some achieved goals fall in the original goal region, they are considered valid achieved goals, and a subset of this intersection will be sampled to cover the region as fully as possible. This subset is selected following the procedure in Figure 1. Note that the distance metric should be determined by the collected original goal distribution. In our experiments, we use the density-weighted Euclidean distance. Specifically, we initialize the hindsight goal set $\mathcal{G}$ with a randomly sampled achieved goal. To make the goal distribute dispersedly, we use Max-Min Distance as the measurement, which indicates the minimal distance between the new goal and the selected ones. By maximizing the minimal distance, it ensures an overall large distance between the new goal and the rest. HGF is related to Curriculum-guided HER (CHER)[Fang *et al.*, 2019] to some extent. However, CHER is suitable for transition-based RL, and cannot be applied to episode-based policy gradient algorithms directly.



(a) Bit Flipping (b) Ms. Pacman (c) Fetch

Figure 2: Demonstration of experiment environments

The complete algorithm of HGF and HTRPO is presented in Appendix E.

5 Experiments

Our experiments aims to answer the following questions:

1. How does HTRPO compared to other methods when performed over diversified tasks? (Section 5.2)
2. What are the main contributors to HTRPO? (Section 5.3)
3. How do key parameters affect the performance? (Section 5.4)

For 1), we show that HTRPO consistently outperforms both HPG and TRPO on success rate and sample efficiency in a wide variety of tasks, and achieves state-of-the-art performance in sparse-reward stochastic policy gradient methods. We also provide an in-depth comparison with HER in this part. For 2), we ablate the main components of HTRPO. The ablation study shows that QKL effectively reduces the variance and significantly improves the performance in all tasks. HGF plays a crucial role in improved performance for the more challenging tasks (e.g. Fetch PickAndPlace). For 3), we vary the scale of KL estimation constraint and the numbers of hindsight goals and choose the best parameter settings.

5.1 Benchmark Settings

We implement HTRPO on a variety of sparse reward tasks. Firstly, we test HTRPO in simple benchmarks established in previous work [Andrychowicz *et al.*, 2017] including 4-to-100-Bit Flipping tasks. Secondly, We verify HTRPO's performance in Atari games like Ms. Pac-Man [Bellemare *et al.*, 2013] with complex raw image input to demonstrate its generalization to convolutional neural network policies. Finally, we test HTRPO in simulated robot control tasks like Reach, Push, Slide and PickAndPlace in Fetch [Plappert *et al.*, 2018] robot environment. As mentioned in [Plappert *et al.*, 2018], it still remains unexplored that to what extent the policy gradient methods trained with hindsight data can solve continuous control tasks. Since HTRPO is a natural candidate that can be applied to both discrete and continuous tasks, other than discrete Fetch environments introduced in [Rauber *et al.*, 2019], we also implement HTRPO in continuous environments including Fetch Reach, Fetch Push, Fetch Slide, Fetch PickAndPlace. A glimpse of these environments is demonstrated in Figure 2, and the inclusive introductions are

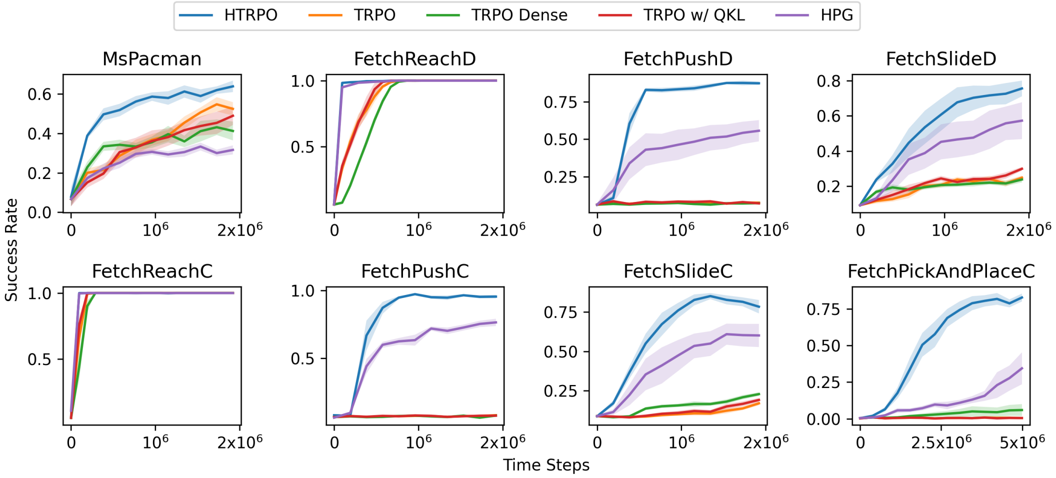


Figure 3: Success rate for benchmark environments. **Top row**: performance of discrete environments. **Bottom row**: performance of continuous environments. The full lines represent the average evaluation over 10 trails and the shaded regions represent the corresponding standard deviation.

included in Appendix F.1. Detailed settings of hyperparameters are listed in Appendix F.2. All experiments are conducted on a platform with NVIDIA GeForce GTX 1080Ti.

We compare HTRPO with HPG [Rauber *et al.*, 2019] and TRPO [Schulman *et al.*, 2015a], which are chosen as the baseline algorithms. The reward setting used in our paper is purely sparse reward, i.e., when the task has not been finished, the agent receives 0 reward in each time step, and once the task is finished, the agent will receive a high positive reward. Besides, TRPO is also implemented with dense rewards and the new KL estimation method proposed in Section 3.1. For a fair comparison, we also combine HPG with Hindsight Goal Filtering in our experiments. To demonstrate the performance level of HTRPO more comprehensively, we also compare HTRPO with the well-known HER algorithm. In all experiments, we directly use the accumulated time steps the agent takes while interacting with the environments throughout episodes and batches, and do not count the hindsight steps which are generated using hindsight goals.

5.2 Comparative Analysis

We evaluate HTRPO's performance from success rate and sample efficiency, and test its generality to different tasks including image-based Atari games, and simulated robot control tasks. Results show HTRPO's consistent effectiveness and strong generality to different kinds of tasks and policies.

Compare with Baselines

The success rate curves for the trained policy are demonstrated in Figure 3. We can conclude that HTRPO consistently outperforms all baselines, including different versions of TRPO and HPG, in most benchmarks, including image-based Atari games (Ms. Pac-Man) and a variety of simulated robot control tasks with different control modes. It demonstrates that HTRPO generalizes well in different kinds of tasks and policies with high-dimensional inputs. Besides, the sample efficiency of HTRPO also exceeds that of HPG, for it reaches a higher average return within less time in most environments.

Compare with HER

We implement HER with DQN [Mnih *et al.*, 2015] for discrete environments and DDPG [Lillicrap *et al.*, 2015] for continuous environments based on OpenAI baselines[2]. We found that HER cannot work well with the purely sparse reward, i.e., the reward is available only when reaching the goal. Thus, we also follow the reward setting in [Andrychowicz *et al.*, 2017] to conduct HER experiments for reference (HER$_{-1}$).

Toy example. To begin with, we test HTRPO on 4-to-100-Bit Flipping task [Andrychowicz *et al.*, 2017] as well as HER (Figure 4). The maximum training steps are $2 \cdot 10^6$. In all Bit Flipping tasks, HTRPO can converge to nearly 100% success rate with much fewer time steps while HER is much data-inefficient as the number of Bits increases.

Benchmarks. Table 1 shows the comparison over the benchmark environments. We can conclude that: 1) HER can not work quite well with the purely sparse reward setting, and HTRPO outperforms HER in 6 out of 7 benchmarks significantly. 2) For discrete robot control tasks, HTRPO can learn a good policy while HER$_{-1}$+DQN cannot work well. For continuous environments, HER$_{-1}$ slightly outperforms HTRPO. In summary, HTRPO can be applied both in discrete and continuous tasks without any modification and achieve commendable performance compared to HER.

5.3 Ablation Studies

There are mainly 2 components in HTRPO, QKL and HGF, that impose an effect on the performance. Besides, we will also investigate the impact of Weighted Importance Sampling (WIS), which is conducive to variance reduction. To study the effect of reward settings, we implement HTRPO with dense rewards. Selected results are shown in Figure 5 and the full ablation study is available in Appendix G.2. We can conclude that: 1) QKL plays a crucial role for the high performance

[2] https://github.com/openai/baselines

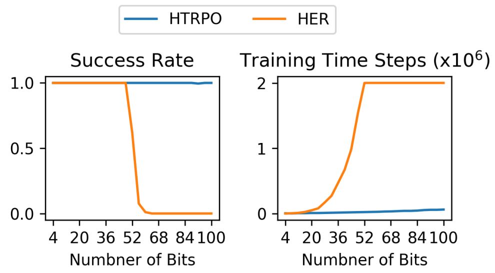


Figure 4: Performance of Bit Flipping.

Environment	HER	HER$_{-1}$	HTRPO
Ms. Pacman	72 ± 3	**74 ± 4**	64 ± 6
Fetch Reach D	53 ± 41	**100 ± 0**	**100 ± 0**
Fetch Push D	8±1	7±2	**88 ± 2**
Fetch Slide D	11±3	10±5	**76 ± 9**
Fetch Reach C	18±3	**100 ± 0**	**100 ± 0**
Fetch Push C	7±2	**100 ± 0**	98 ± 1
Fetch Slide C	1±1	**93 ± 7**	85 ± 4

Table 1: Success rate comparison between HTRPO and HER (%). HER$_{-1}$ means using the original -1-and-0 reward setting instead of the purely sparse reward that HTRPO used, i.e., only when the agent achieves the goal can it receive a high reward.

of HTRPO by significantly reducing the estimation variance of KL divergence; 2) HGF can enhance the performance of HTRPO to a higher level; 3) WIS is important since it can reduce the variance of importance sampling significantly; 4) Dense-reward setting harms the performance, which has also been verified in [Plappert *et al.*, 2018].

5.4 Hyperparameter Selection

We take Continuous Fetch Push as an example to study the impact of different KL estimation constraint scales and different numbers of hindsight goals.

Different KL estimation constraint scales. KL estimation constraint, i.e. max KL step specifies the trust region, the range within which the agent searches for the next-step optimal policy. In the sense of controlling the scale to which the agent updates the policy per step, this parameter presents similar functionality as learning step size. If set too low, say 5e-6 shown in Figure 6, it would inevitably slow down the converging speed. If set too high, the potentially large divergence between the new and old policy may violate the premise for some core parts of HTRPO theory derivation including Proposition 1 and HTRPO solving process.

Different number of hindsight goals. From the results in Figure 7, it is straightforward that more hindsight goals lead to faster converging speed. This phenomenon accords with the mechanism of how hindsight methodology deals with sparse reward scenarios, i.e. it augments the sample pool with substantial hindsight data rather than leaving it with few valid original trajectories. It's intuitive that the more hindsight data there are, the higher sample efficiency HTRPO achieves. However, limited by the hardware resources, we need to trade off the sampled goal number.

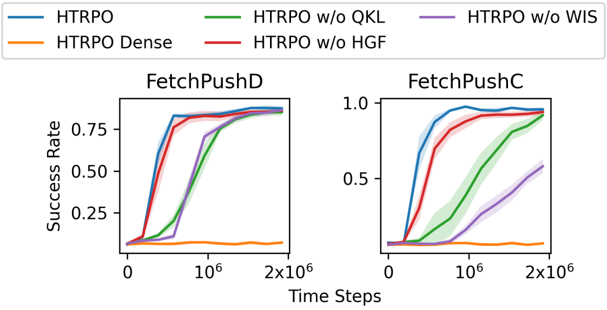


Figure 5: Ablation Experiments.

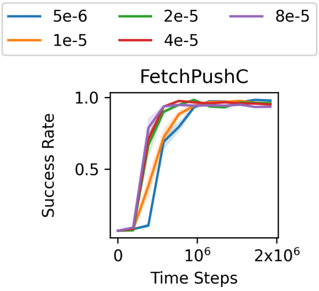


Figure 6: Max KL steps.

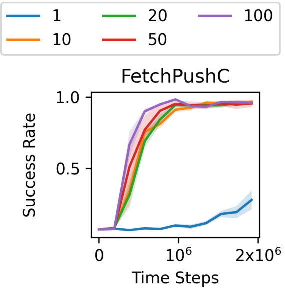


Figure 7: Goal numbers.

6 Conclusion

We proposed Hindsight Trust Region Policy Optimization(HTRPO), a new RL algorithm that extends the highly successful TRPO algorithm with hindsight to tackle the challenge of sparse rewards. We show that with the help of the proposed Quadratic KL divergence Estimation (QKL), HTRPO significantly reduces the variance of KL estimation and improves the performance and learning stability. Moreover, we design a Hindsight Goal Filtering mechanism to narrow the discrepancy between hindsight and original goal space, leading to better performance. Results on diversified benchmarks demonstrate the effectiveness of HTRPO.

Since HTRPO is a natural candidate for both discrete and continuous tasks and the QKL constraint gets rid of the demand for analytical form, it is promising to optimize policies with non-Gaussian (e.g. GMM) or mixed (discrete+continuous) action space. It also provides the possibility to tackle high-dimensional real-world problems and train robot control policies without arduous reward shaping. Besides, HGF can be integrated into hindsight-goal exploration methods naturally [Ren *et al.*, 2019; Pitis *et al.*, 2020], which should lead to a higher performance.

Acknowledgements

This work was supported in part by NSFC under grant No.91748208, No.62088102, No.61973246, Shaanxi Project under grant No.2018ZDCXLGY0607, and the program of the Ministry of Education. D. Hsu is supported by the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG2-RP-2020-016).

References

[Andrychowicz *et al.*, 2017] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.

[Bellemare *et al.*, 2013] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013.

[Deisenroth *et al.*, 2013] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

[Fang *et al.*, 2019] Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 12623–12634, 2019.

[Grzes, 2017] Marek Grzes. Reward shaping in episodic reinforcement learning. In *International Joint Conference on Autonomous Agents and Multi-agent Systems*, 2017.

[Ho and Ermon, 2016] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[Ho *et al.*, 2016] Jonathan Ho, Jayesh Gupta, and Stefano Ermon. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning*, pages 2760–2769, 2016.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Ng *et al.*, 1999] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[Peters and Schaal, 2008] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

[Pitis *et al.*, 2020] Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International Conference on Machine Learning*, pages 7750–7761. PMLR, 2020.

[Plappert *et al.*, 2018] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[Precup *et al.*, 2000] Doina Precup, Richard S Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In *ICML'00 Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[Rauber *et al.*, 2019] Paulo Rauber, Avinash Ummadisingu, Filipe Mutz, and Jürgen Schmidhuber. Hindsight policy gradients. In *International Conference on Learning Representations*, 2019.

[Ren *et al.*, 2019] Zhizhou Ren, Kefan Dong, Yuan Zhou, Qiang Liu, and Jian Peng. Exploration via hindsight goal generation. In *Advances in Neural Information Processing Systems*, pages 13485–13496, 2019.

[Schaul *et al.*, 2015] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320, 2015.

[Schulman *et al.*, 2015a] 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.

[Schulman *et al.*, 2015b] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[Sutton and Barto, 2018] Ricahrd S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning series. MIT Press, 2018.

[Veeriah *et al.*, 2018] Vivek Veeriah, Junhyuk Oh, and Satinder Singh. Many-goals reinforcement learning. *arXiv preprint arXiv:1806.09605*, 2018.