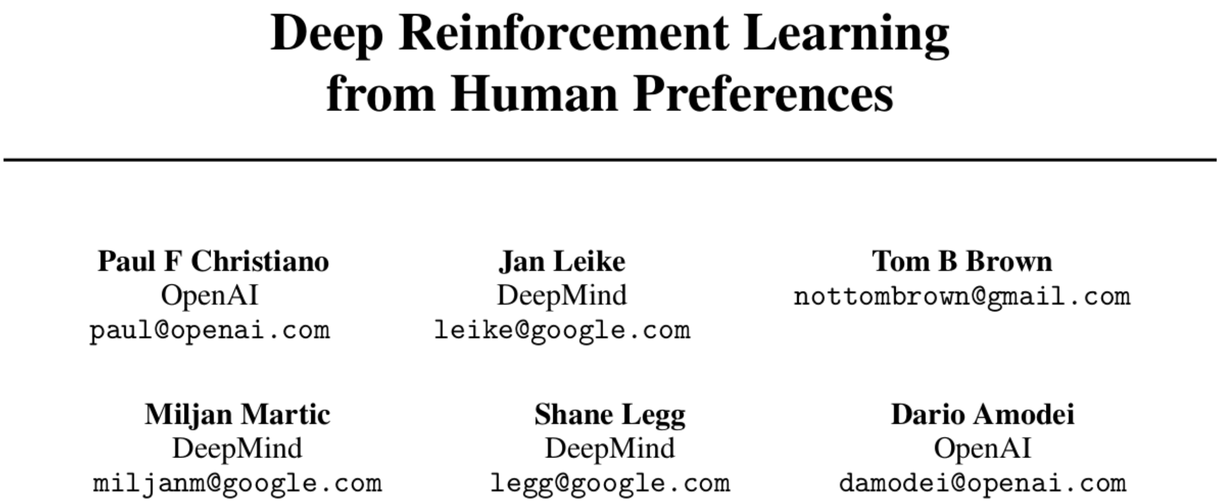
1706.03741 - Deep Reinforcement Learning from Human Preferences



Deep Reinforcement Learning from Human Preferences

Paul F Christiano
OpenAI
paul@openai.com

Jan Leike
DeepMind
leike@google.com

Tom B Brown
nottombrown@gmail.com

Miljan Martic
DeepMind
miljanm@google.com

Shane Legg
DeepMind
legg@google.com

Dario Amodei
OpenAI
damodei@openai.com

- **Yunqiu Xu**
- Other reference
  - OpenAI blog
  - 机器之心
  - Two Minute Papers
  - An implementation
- Leverage **simple human preferences (give feedback)** to help learning → solve complex RL tasks without access to reward function

1. Introduction

- Challenge
  - For many tasks, goals are complex, poorly-defined, hard to specify
  - Imitation learning : sometimes it's difficult to provide demonstration
  - Directly human feedback : require hundreds or thousands of hours of experience
- Our work:

    - Learn reward function from human feedback and then to optimize that reward function
    - Try to decrease the amount of human feedback to make it more practical
    - Goal: handle sequential decision problems without a well-specified reward function
        - 不一定需要提供完整的demonstration, 只要提供偏好就行(e.g. 哪个agent的表现看起来好一些)
        - 不一定需要专家级demonstration → 普通人也可提供feedback
        - 不需要提供过多或者过于复杂的回馈, 即使对于大型问题

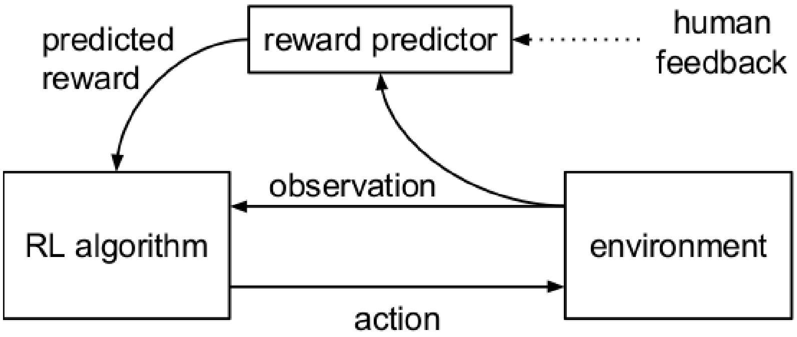


Figure 1: Schematic illustration of our approach: the reward predictor is trained asynchronously from comparisons of trajectory segments, and the agent maximizes predicted reward.

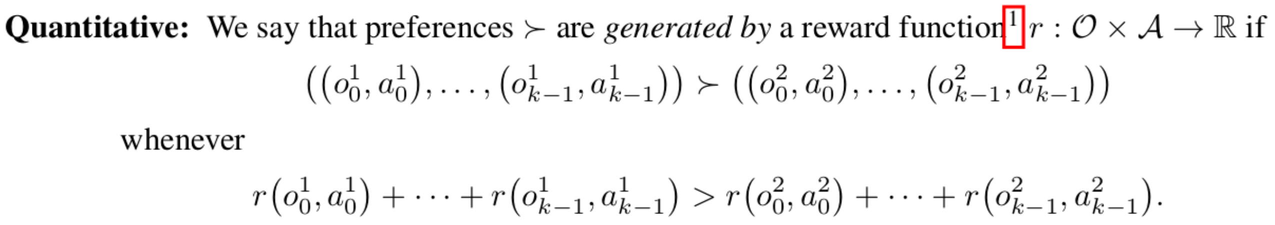
- 同以往工作相比, 我们的主要贡献在于**极大简化了human feedback**, 例如对于工作Akrour et al, 2014. Programming by Feedback, feedback为对整体trajectory的偏好, 而我们的只要对short clip进行判断就好, 更高效

2. Preliminaries and Method

2.1 Problem Setting

- Trajectory segment: a sequence of observations and actions $\sigma = ((o_0, a_0), (o_1, a_1), \ldots, (o_{k-1}, a_{k-1})) \in (O \times A)^k$
- Goal: let the agent produce trajectories which are preferred by human, while making as few queries as possible to human

- By using human preference, we can try to evaluate the algorithm qualitatively, which can help to improve it quantitatively
- 本文也给出了如何定量评估preference的方法:



Quantitative: We say that preferences $\succ$ are *generated by* a reward function[1] $r : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ if

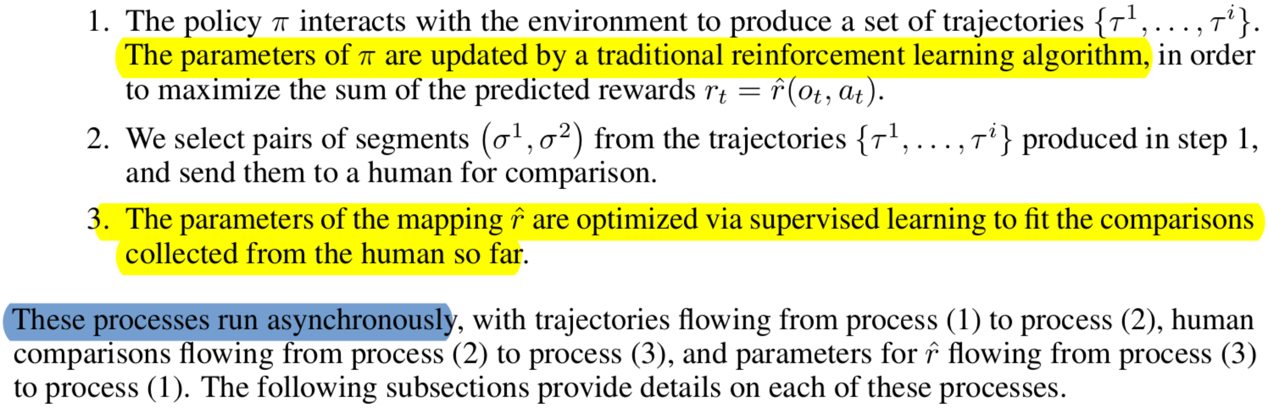
$$\left(\left(o_0^1, a_0^1\right), \ldots, \left(o_{k-1}^1, a_{k-1}^1\right)\right) \succ \left(\left(o_0^2, a_0^2\right), \ldots, \left(o_{k-1}^2, a_{k-1}^2\right)\right)$$

whenever

$$r\left(o_0^1, a_0^1\right) + \cdots + r\left(o_{k-1}^1, a_{k-1}^1\right) > r\left(o_0^2, a_0^2\right) + \cdots + r\left(o_{k-1}^2, a_{k-1}^2\right).$$

2.2 Method

- Policy $\pi : O \to A$
- Estimated reward function $\hat{r} : O \times A \to R$
- Update process:

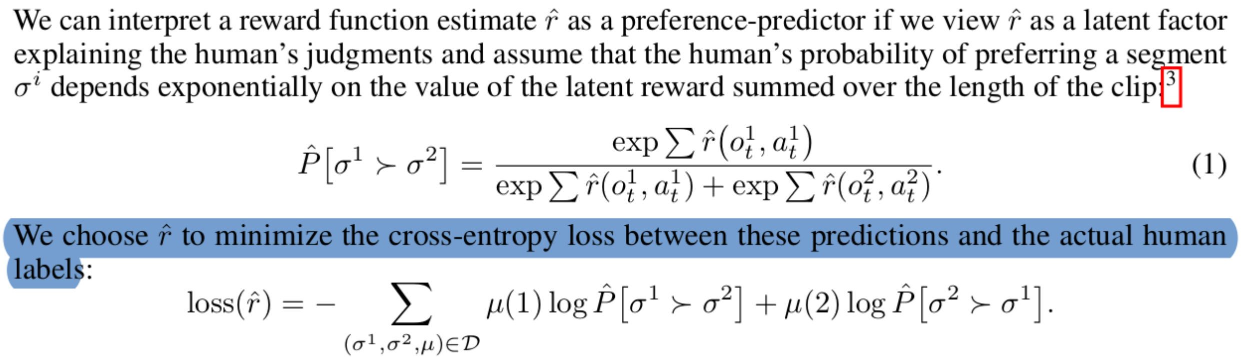


1. The policy π interacts with the environment to produce a set of trajectories $\{\tau^1, \ldots, \tau^i\}$. The parameters of π are updated by a traditional reinforcement learning algorithm, in order to maximize the sum of the predicted rewards $r_t = \hat{r}(o_t, a_t)$.
2. We select pairs of segments $\left(\sigma^1, \sigma^2\right)$ from the trajectories $\{\tau^1, \ldots, \tau^i\}$ produced in step 1, and send them to a human for comparison.
3. The parameters of the mapping $\hat{r}$ are optimized via supervised learning to fit the comparisons collected from the human so far.

These processes run asynchronously, with trajectories flowing from process (1) to process (2), human comparisons flowing from process (2) to process (3), and parameters for $\hat{r}$ flowing from process (3) to process (1). The following subsections provide details on each of these processes.

2.3 Details of the Method

- How to optimize policy: A2C for Atari, TRPO for robot tasks
- The shape of human judgement : A database D of $(\sigma_1, \sigma_2, \mu)$
  - σ_1, σ_2 : two segments to compare
  - μ : distribution over $\{1, 2\}$ indicating which segment is preferred $\to$
    - If one segment is more preferable $\to$ all mass of μ will be put on it
    - If segments are equally preferable $\to$ μ is uniform
    - If segments are incomparable $\to$ this pair will not be included in D
- How to fit reward function with preference result



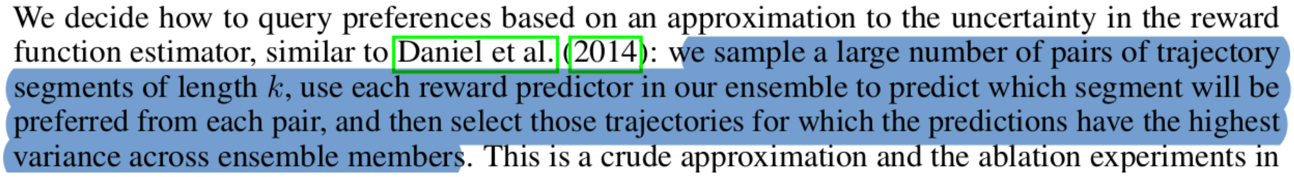
We can interpret a reward function estimate $\hat{r}$ as a preference-predictor if we view $\hat{r}$ as a latent factor explaining the human's judgments and assume that the human's probability of preferring a segment σ^i depends exponentially on the value of the latent reward summed over the length of the clip.[3]

$$\hat{P}[\sigma^1 \succ \sigma^2] = \frac{\exp \sum \hat{r}(o_t^1, a_t^1)}{\exp \sum \hat{r}(o_t^1, a_t^1) + \exp \sum \hat{r}(o_t^2, a_t^2)}. \tag{1}$$

We choose $\hat{r}$ to minimize the cross-entropy loss between these predictions and the actual human labels:

$$\text{loss}(\hat{r}) = -\sum_{(\sigma^1,\sigma^2,\mu)\in\mathcal{D}} \mu(1) \log \hat{P}[\sigma^1 \succ \sigma^2] + \mu(2) \log \hat{P}[\sigma^2 \succ \sigma^1].$$

- How to select queries:



We decide how to query preferences based on an approximation to the uncertainty in the reward function estimator, similar to Daniel et al. (2014): we sample a large number of pairs of trajectory segments of length k, use each reward predictor in our ensemble to predict which segment will be preferred from each pair, and then select those trajectories for which the predictions have the highest variance across ensemble members. This is a crude approximation and the ablation experiments in

3. Experiment

3.1 RL Tasks with unobserved rewards

- Learn the goal by only asking human which of two trajectory segments is better
- Atari result

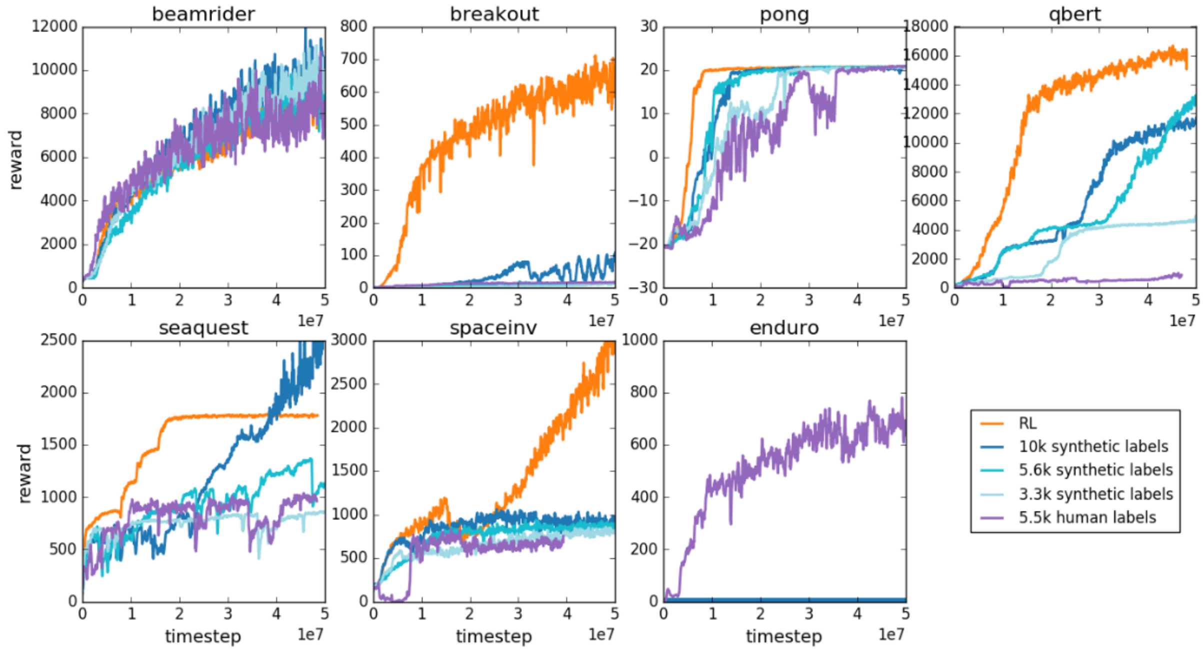


Figure 3: Results on Atari games as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are the average of 3 runs, except for the real human feedback which is a single run, and each point is the average reward over about 150,000 consecutive frames.

- Mujoco result

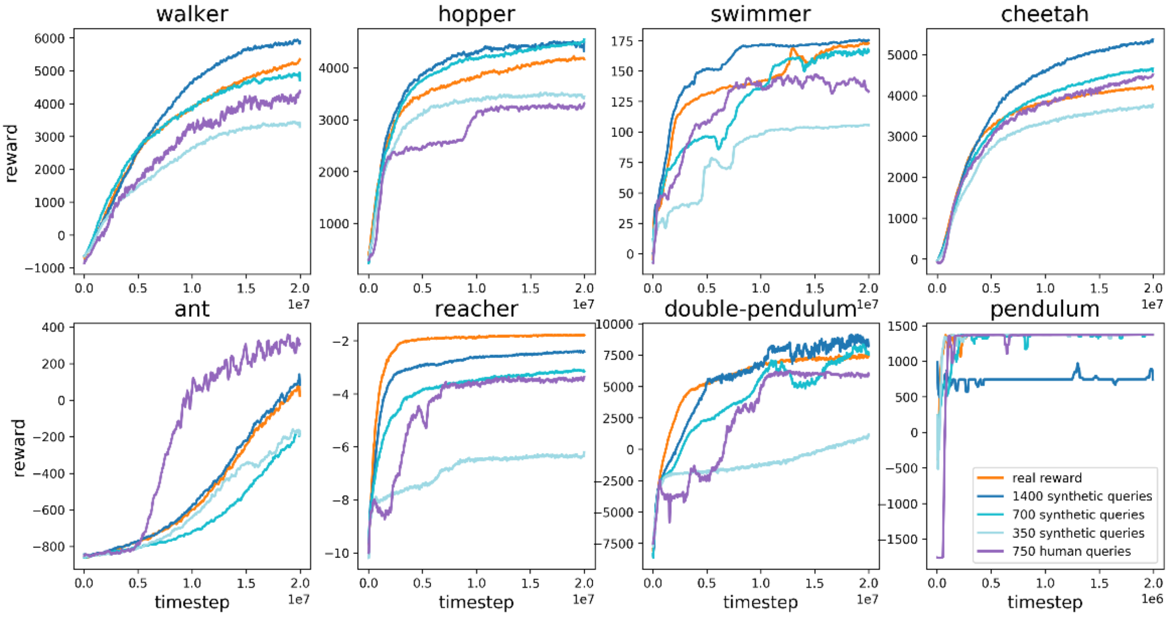


Figure 2: Results on MuJoCo simulated robotics as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are the average of 5 runs, except for the real human feedback, which is a single run, and each point is the average reward over five consecutive batches. For Reacher and Cheetah feedback was provided by an author due to time constraints. For all other tasks, feedback was provided by contractors unfamiliar with the environments and with our algorithm. The irregular progress on Hopper is due to one contractor deviating from the typical labeling schedule.

3.2 Novel Behaviors

- Try to solve some complex behaviors where no reward function is available
    - The Hopper robot performing a sequence of backflips
    - The Half-Cheetah robot moving forward while standing on one leg
    - Keeping alongside other cars in Enduro

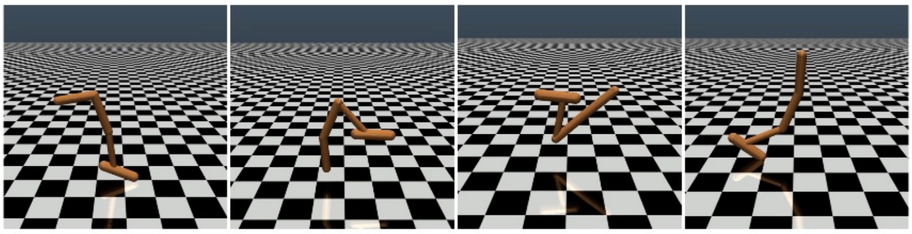


Figure 4: Four frames from a single backflip. The agent is trained to perform a sequence of backflips, landing upright each time. The video is available at https://goo.gl/MhgvIU.

3.3 Ablation Studies

- Atari result

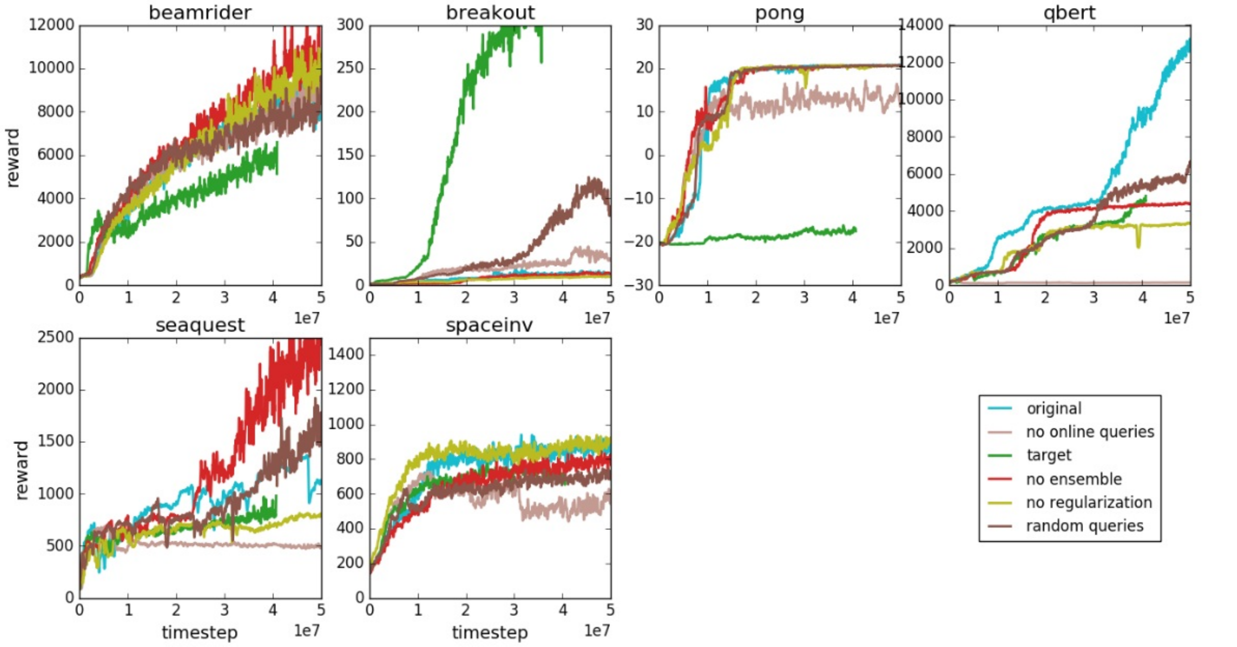


Figure 6: Performance of our algorithm on Atari tasks after removing various components, as described in Section 3.3. All curves are an average of 3 runs using 5,500 synthetic labels (see minor exceptions in Section A.2).

- Mujoco result

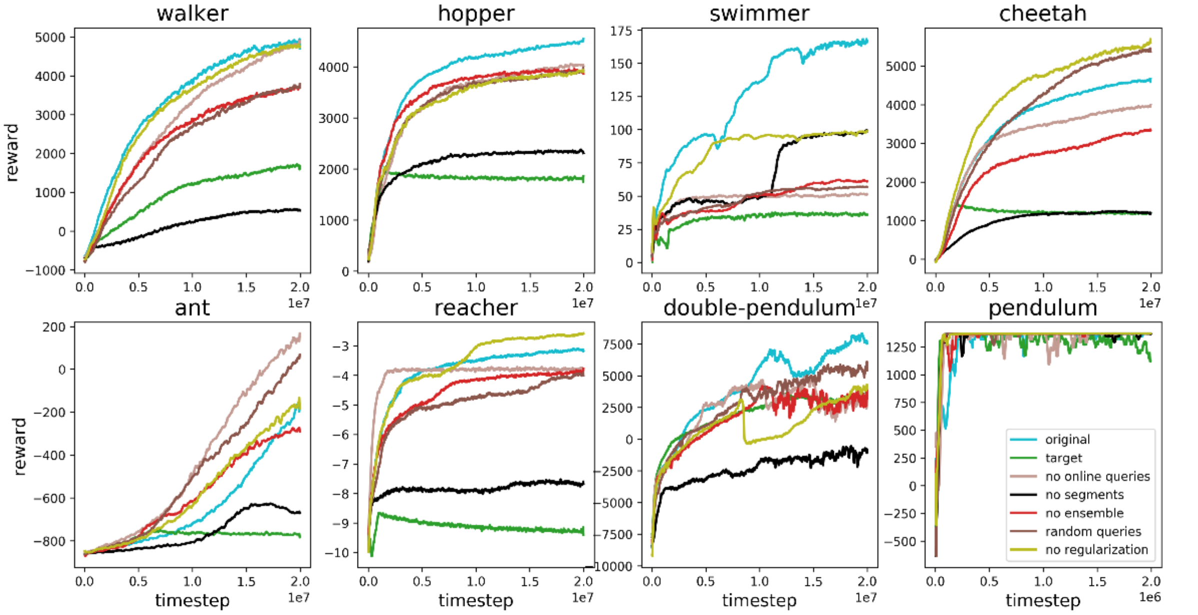


Figure 5: Performance of our algorithm on MuJoCo tasks after removing various components, as described in Section Section 3.3. All graphs are averaged over 5 runs, using 700 synthetic labels each.

4. Summary

- 本文利用human preference来加速学习过程, 并尝试解决一些不容易定义reward function的任务
- 虽然之前已经有关于利用preference或者未知reward function的工作, 本文的创新点在于尽可能设置有效且高效的preference selection, 在提升性能的同时尽可能减少对人类参与的需求
- 和 imitation learning 相比, 本工作不需要提供"专家"的demonstration, 仅仅提供比较偏好即可, 在真实世界中我们往往难以获得足够多的demonstration, 该工作或许能提供一些思路