Deep Reinforcement Learning from Human Preferences

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, Dario Amodei
NIPS 2017

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

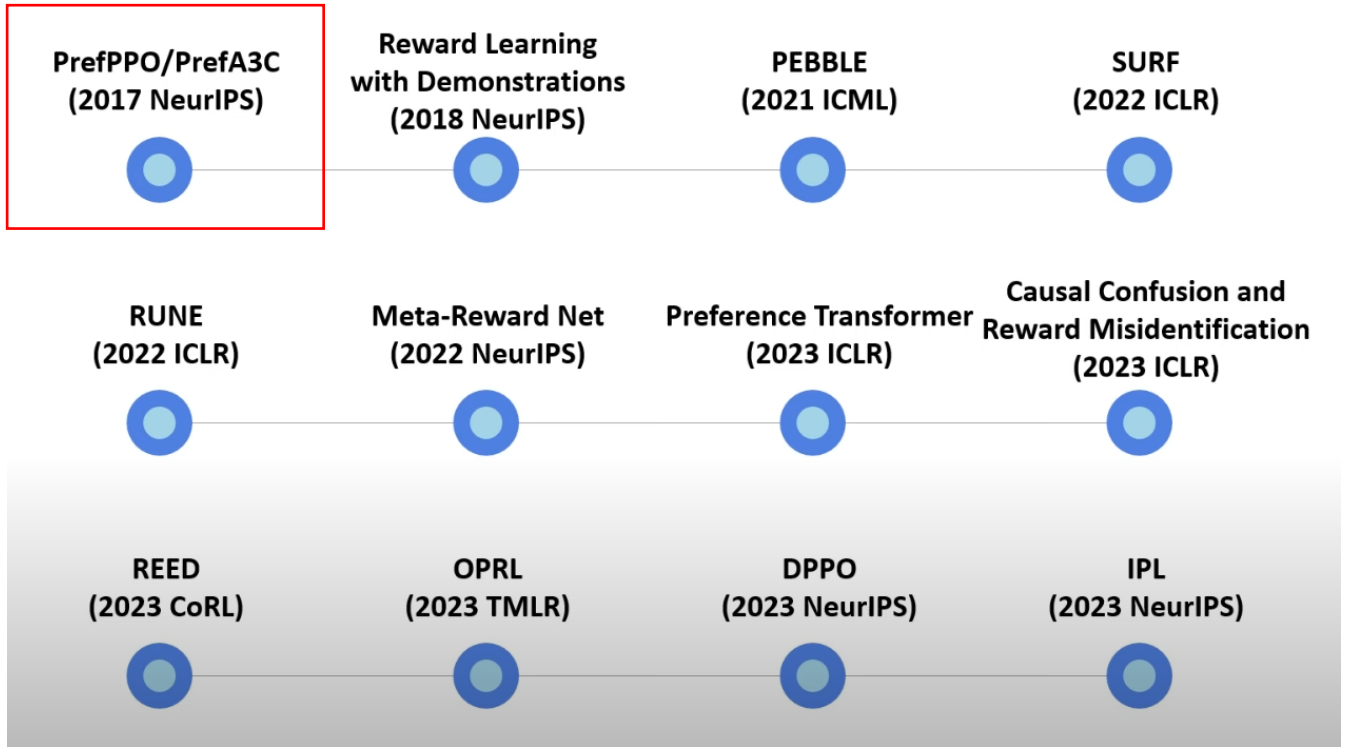
Shane Legg
DeepMind
legg@google.com

Dario Amodei
OpenAI
damodei@openai.com

2024. 10. 10
Learning Agents 강화학습 논문 리뷰 스터디
Minkyoung Kim

해당 논문의 의의

- Preference-based RL을 최초로 제안한 논문
- On-policy 알고리즘인 A3C와 TRPO에 True Reward 없이 선호도 기반으로 학습 가능함을 입증
- InstructGPT를 통한 챗봇, 요약 언어 모델에 RLHF를 적용할 수 있는 계기 마련



Agenda

- Introduction
- Related work
- Preliminaries and Method
- Conclusions
- References

Introduction

- Recent success in scaling RL to large problems has been driven in domains that have a well-specified reward function.
- Unfortunately, many task involve goals that are complex, poorly-defined, or hard to specify.

- Could try to design a simple reward function that approximately captures the intended behavior.
- But there are difficulties between our values(preferences) and the objectives of our RL systems.

- Could extract reward function using inverse RL or could use imitation learning to clone the demonstrated behavior.
- However, theses approaches are not directly applicable to behaviors that are difficult for humans to demonstrates.
  - e.g. controlling a robot with many degrees of freedom but very non-human morphology

Introduction

- An alternative approach is to allow a human to provide feedback on our system's current behavior and to use this feedback to define the task.

- In summary, Desire a solution to sequential decision problems without a well-specified reward function that:
  - Enable us to solve tasks for which we can only *recognize* the desired behavior, but not necessarily demonstrate it.
  - allows agent to be taught by non-expert users.
  - scales to large problems,
  - is economical with user feedback.
    - requiring hundreds or thousands of hours of human experience is expensive for RL

Introduction

- Fits a reward function to the human's preferences
  while simultaneously training a policy to optimize the current predicted reward function.

- Human feedback
  - compare short video clips of agent's behavior,
    rather than to supply an absolute numerical score.

- Experiments in Atari, robotics task in MuJoCo
  - small amount of feedback from a non-export human(15mins ~ 1hour)
    suffices to learn most of the original RL tasks
    even when the reward function is not observable.
  - Contractors responded to average query in 3~5 seconds.

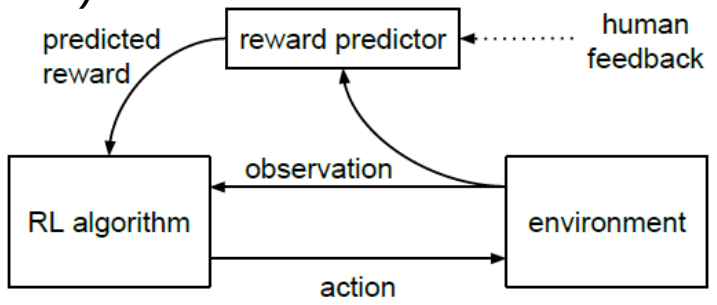


Figure 1: Schematic illustration of our approach: the reward predictor is trained asynchronously from comparisons of trajectory segments, and the agent maximizes predicted reward.

Related work

- Consider physics tasks with dozens of degrees of freedom and Atari tasks with no hand-engineered features: (complexities)
  - past: assume four degrees of freedom and small discrete domains

- Elicit preferences over short clips
  - past: whole trajectories

- use real human feedback gathered from non-expert users
  - past: use synthetic human feedback from Bayesian model

Preliminaries and Method

- Setting and Goal
  - traditional RL, the environment would supply a reward $r_t \in \mathbb{R}$.
  - the agent's goal would be to maximize the discounted sum of rewards

  - Assume that there is a human overseer who can express preferences between *trajectory segments*.
  - Trajectory segments is a sequence of observations and actions;

$$\sigma = ((o_0, a_0), (o_1, a_1), \ldots, (o_{k-1}, a_{k-1})) \in (\mathcal{O} \times \mathcal{A})^k$$

  - $\sigma^1 \succ \sigma^2$: human preferred trajectories segment σ^1 to trajectory segment σ^2

  - the agent's goal is to produce trajectories which are preferred by the human, while making as few queries as possible to the human

Preliminaries and Method

- Setting and Goal
  - evaluate our algorithms' behavior in two ways; Quantitative and Qualitative

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

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

whenever

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

If the human's preferences are generated by a reward function r, then our agent ought to receive a high total reward according to r. So if we know the reward function r, we can evaluate the agent quantitatively. Ideally the agent will achieve reward nearly as high as if it had been using RL to optimize r.

Qualitative: Sometimes we have no reward function by which we can quantitatively evaluate behavior (this is the situation where our approach would be practically useful). In these cases, all we can do is qualitatively evaluate how well the agent satisfies to the human's preferences. In this paper, we will start from a goal expressed in natural language, ask a human to evaluate the agent's behavior based on how well it fulfills that goal, and then present videos of agents attempting to fulfill that goal.

Preliminaries and Method

- Method
  - maintains a policy $\pi : O \longrightarrow A$ and reward function estimate $\hat{r} : O \times A \rightarrow \mathbb{R}$, each parametrized by deep neural networks.
  - These networks are updated by three processes: (These process run asynchronously)

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 (σ^1, σ^2) 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 comparison collected from the human so far.

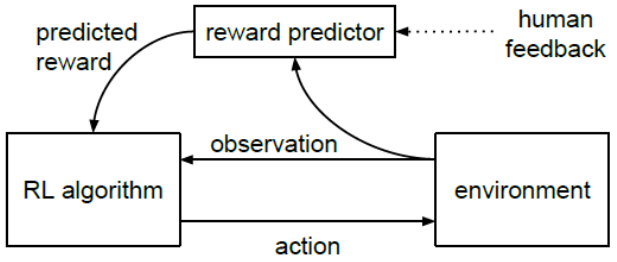
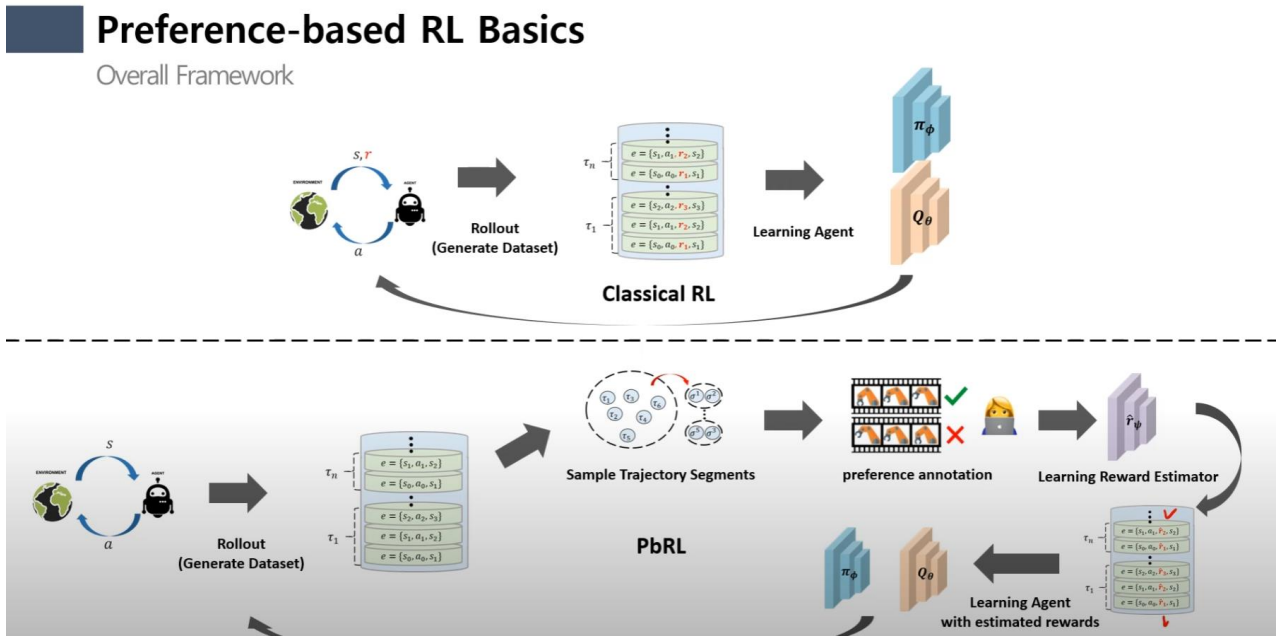


Figure 1: Schematic illustration of our approach: the reward predictor is trained asynchronously from comparisons of trajectory segments, and the agent maximizes predicted reward.

Preliminaries and Method

- Method – Optimizing the Policy
  - After using reward function estimate $\hat{r}$, we left with a traditional RL
  - can use any RL algorithms that is appropriate for the domain.
  - reward function $\hat{r}$ may be non-stationary,
    which leads us to prefer methods which are robust to changes in the reward function.
  - This led us to focus on-policy gradient methods: A2C to Atari, TRPO to robotics task



Preliminaries and Method

- Method – Preference Elicitation
  - the human overseer is given a visualization of two trajectory segments, in the form of short movie clips. In experiments, these clips are between 1 and 2 seconds long.

  - The human judgments are recoded in a database D of triples $(\sigma^1, \sigma^2, \mu)$;
    - σ^1, σ^2: trajectory segments
    - μ : distribution over {1, 2} indicating which segment the user preferred.

  - 3 types of responses; Prefer, Equally good, unable to compare
    - Prefer: μ puts all of its mass on that choice
    - Equally preferable: the μ is uniform
    - incomparable: not included in the database

Preliminaries and Method

- Method – Fitting the Reward Function
  - reward function estimate $\hat{r}$ == preference-predictor
  - Follows the Bradley-Terry Model for estimating score functions from pairwise preferences

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

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

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

Preliminaries and Method

- Method – Selecting Queries
  - How to query preferences?
  - 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 pairs
  - and select those trajectories for which the predictions have the highest variation across ensemble members.

Experimental Results

- Attempt to solve a range of benchmark tasks for deep RL without observing the true reward.
- Instead, the agent learns about the goal of the task only by asking a human which of two trajectory segments is better.
- Solve the task in a reasonable amount of time using as few queries as possible.
  - use real human feedback required between 30 minutes and 5 hours of human time.
- For comparison, run experiments using synthetic oracle data.
  - instead of sending the query to a human, reply immediately by indicating a preference.

- **aim is not to outperform but rather to do nearly as well as RL without access to reward information.**

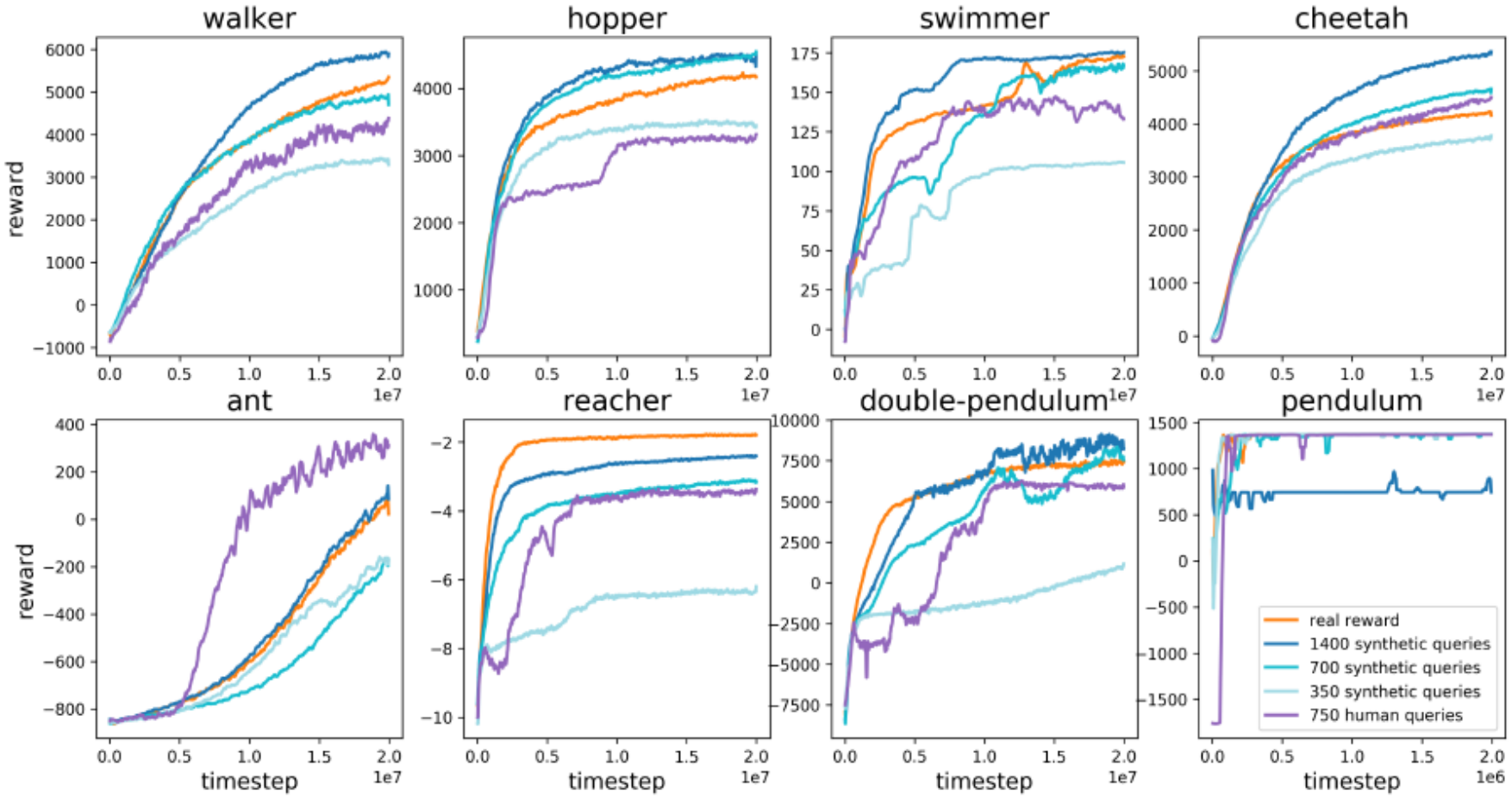


Figure 2: Results on MuJoCo simulated robotics as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are 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.

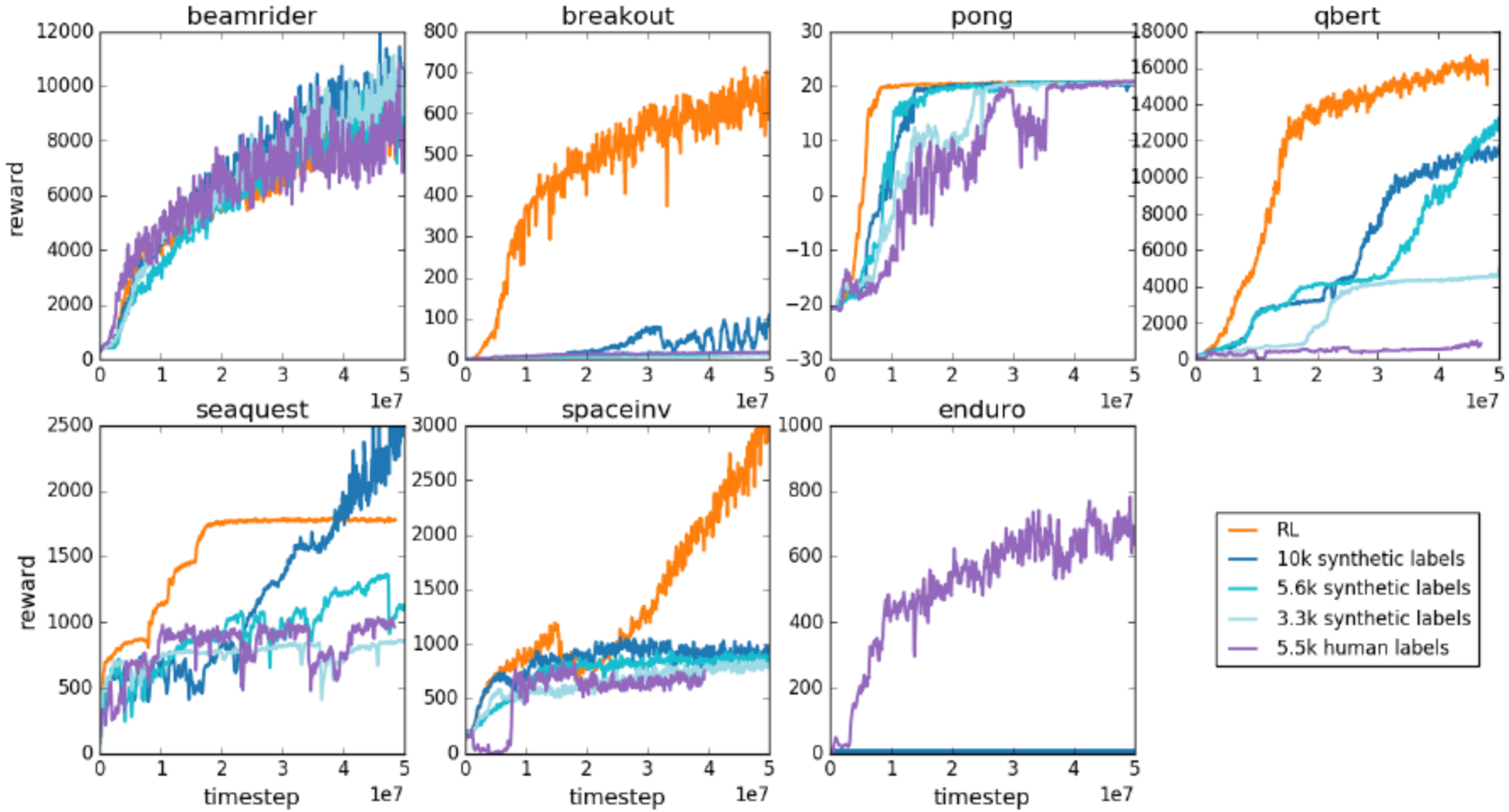
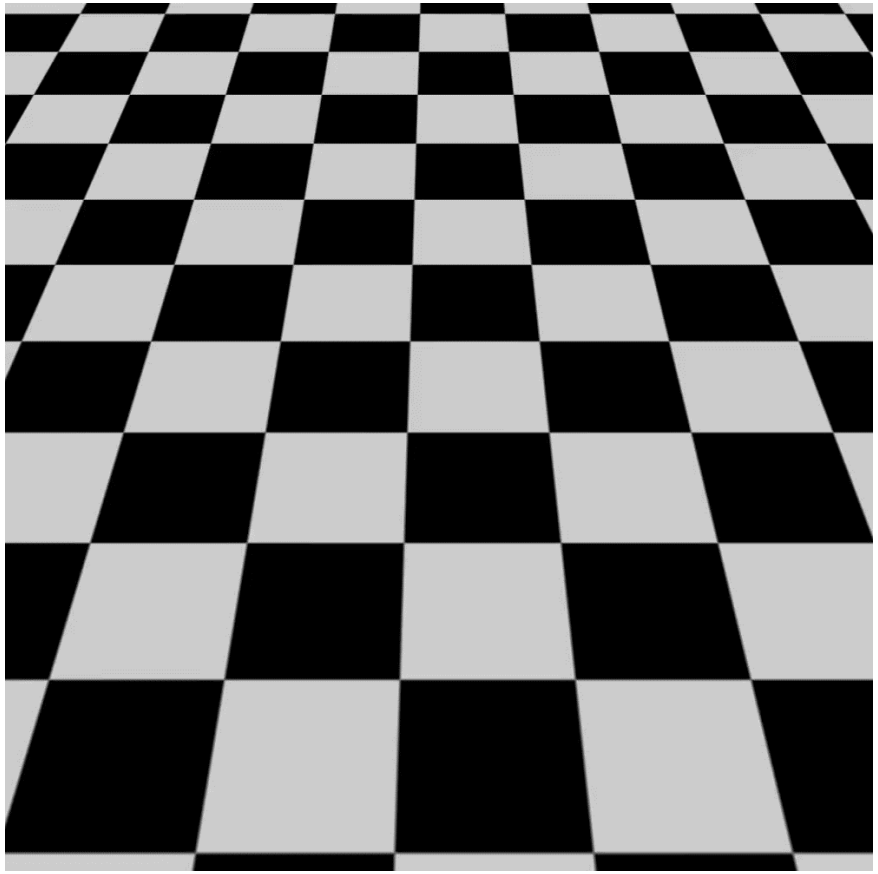
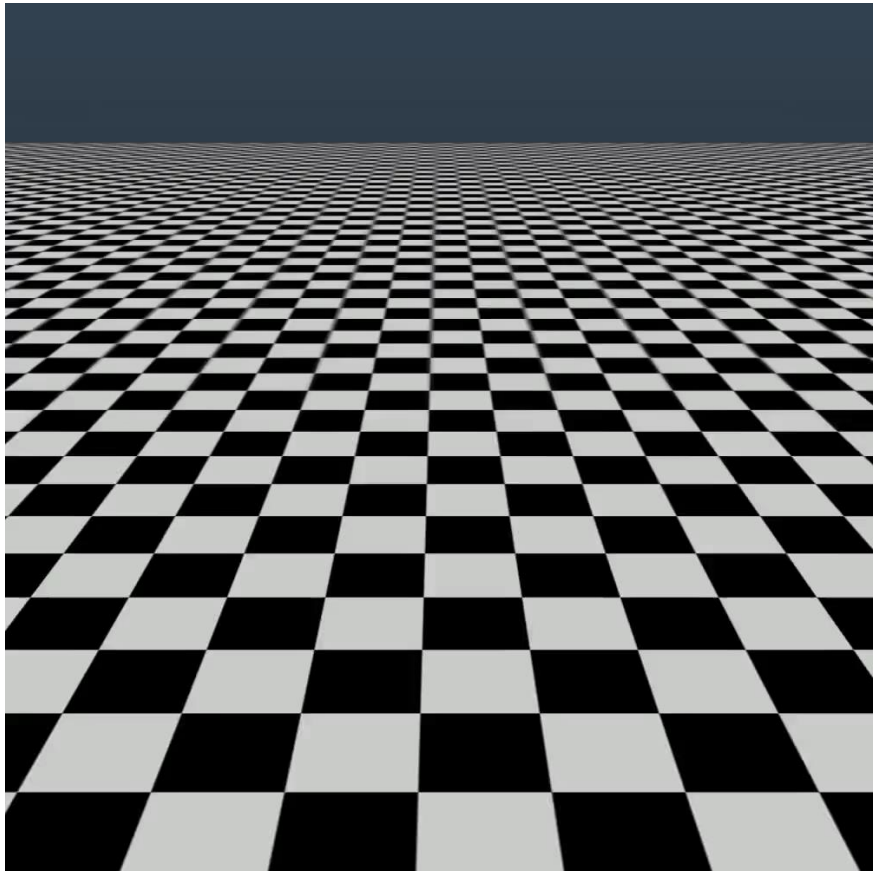


Figure 3: Results on Atari games as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are the average 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.

Experimental Results



Cheetah_Handstand
training using 800 queries in less than a hour



Hopper_Backflips
using 900 queries in less than a hour

Conclusions

- Although there is a large literature on preference elicitation and reinforcement learning from unknown reward functions,
- provide the first evidence that these techniques can be economically scaled up to sate-of-the-art reinforcement learning systems.

References

- [Open DMQA Seminar] RLHF-Preference-based Reinforcement Learning, https://www.youtube.com/watch?v=Vzno0oBbm6w
- [2017] Deep reinforcement learning from human preferences, https://youtu.be/B48rm3jeGmQ?si=heolqyBHdbNBS5WL