Balancing Shapes using Human Preferences

Takuma Osaka, Luke Kulm
(to95, lbk73)

1 Motivation

Balancing shapes is a fundamental, yet crucial task for many real-world situations. From building a snowman to automating warehouses, balancing shapes is inevitable. Inspired by the success of OpenAI's "Deep Reinforcement Learning from Human Preferences [1]," where they managed to make a robot learn to backflip in 900 human-feedback bits, this project explores a more practical (and probably a more difficult challenge) of precisely balancing shapes by receiving human feedback on pairs of trajectory clips. If robots can efficiently learn to grasp a "sense of balance", an unimaginable world lies ahead.

2 Problem Formulation

Our goal is to teach a robotic arm to stack shapes on top of each other while balancing them in a stable configuration. We will have a variety of shapes including cubes, spheres, and pyramids to experiment how our robotic arm will perform on various balancing difficulties. Traditional RL will require a well-defined reward function, but our project allows a human to provide binary preferences over two short video clips of the robot trying to accomplish the task. These comparisons will be used to formulate a reward predictor during training and guide our robot. Our main hypothesis is that "an agent trained on human preferences over trajectory segments will lead to faster and more stable balancing behaviors than traditional RL methods."

3 Method Overview

We will implement the learning-from-preferences pipeline as described in [1]. The setup includes:

- A trajectory generator using a simulation environment with a robot arm
- An RL policy trained to maximize predicted rewards (using PPO)
- A human-in-the-loop interface for binary preferences on two videos of selected trajectories
- A reward predictor model trained on these human preferences

This method will run in a loop, where the newly generated reward predictor is used to retrain the RL policy after prefences are given. To fit the reward model to user prefences we will use Bradley-Terry Loss [2] which helps estimate score functions for pairwise prefences. We will implement this method as in [1] as well as implement a traditional hard-coded reward function. The traditional reward function will provide a baseline to compare our method against.

4 Expected Results

We expect that training an agent to balance objects using human preferences will produce more stable and intuitive behaviors than training with hand-crafted rewards. Specifically, we anticipate that preference-based learning will lead to:

- Improved generalization to unseen shapes or surfaces

- Smoother, safer balancing strategies more aligned with subjective human expectations
- More human-like behavior with lower odds of reward hacking

Compared to a traditional RL methods, our agent should be able to balance more unique shapes for longer, and exhibit behaviors that better match subjective human judgment.

5 Plan

We will use the PyBullet simulator with the Franka Panda robot arm to simulate object balancing tasks. PyBullet provides high-quality physics and manipulation environments that are well-suited for fine control tasks like stacking and balancing. To collect human preferences, we will build a simple web interface that shows pairs of short video clips of the robot attempting to balance objects. The preference feedback will be used to train a reward predictor, following the method described by [1]. We will train our reinforcement learning agent using Proximal Policy Optimization (PPO) with the reward model.

Work Division:

- Takuma: Set up the PyBullet environment and implement the RL agent using PPO.
- Luke: Develop the Web interface for collecting human preferences. Create the and integrate the reward predictor with preference collection.
- Both: Run experiments, evaluate results, and co-write the final report and presentation

Timeline:

- April 8: Submit proposal
- April 13: PyBullet environment and baseline PPO agent set up
- April 18: Human preference collection interface complete
- April 22: Initial reward predictor trained on sample preferences
- April 26: Train RL agent using learned reward model
- April 28: Create hard-coded reward function for task
- April 30: Evaluate and compare to baselines
- May 4: Final report and presentation preparation

References

[1] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. 2023.

[2] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. 1952.