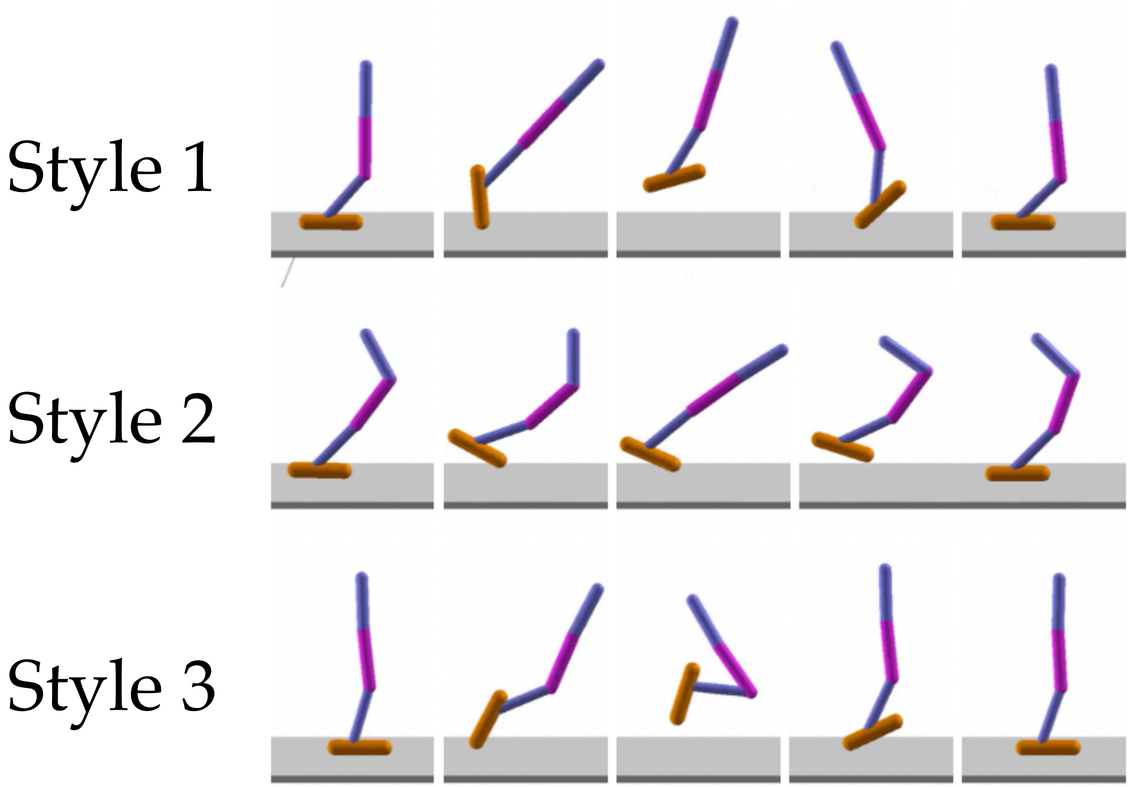
Learning Novel Policies For Tasks

Yunbo Zhang, Wenhao Yu, Greg Turk

Georgia Tech School of Interactive Computing
College of Computing

Motivation

- Want more than one solution (i.e. novel solutions) to a problem.
  - E.g. Different Locomotion styles for legged robots.



Key Aspects

- Novelty measurement function
  - Measures the novelty of a trajectory compared with trajectories from other policies

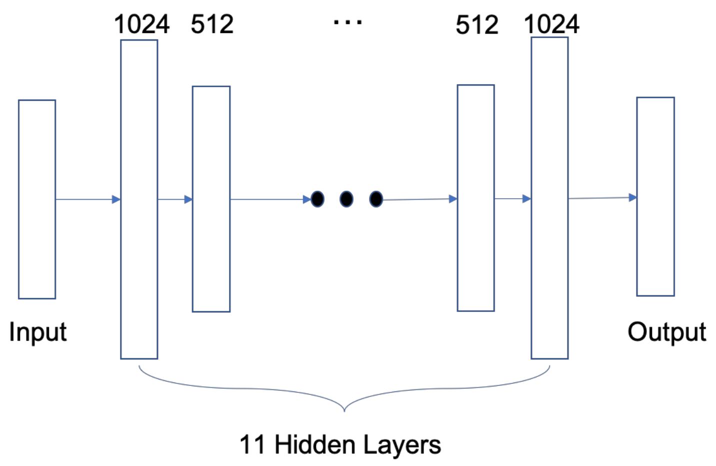
- Policy Gradient Update
  - Make sure final gradient compromises between task and novelty
  - Task-Novelty Bisector (TNB)

Method Overview

- Define a separate novelty reward function apart from task reward.
- Train a policy using Task-Novelty Bisector (TNB) to balance the optimization of task and novelty.
- Update novelty measurement function.
- Repeat

Novelty Measurement

- Use autoencoder reconstruction error of state sequences to compute novelty.
- One autoencoder for each policy.



- For the set of autoencoders $\boldsymbol{D} = \{D_1, \ldots, D_i\}$, the novelty reward function is:

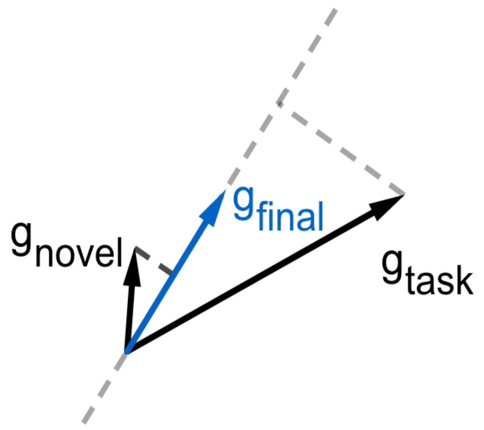
$$r_{novel} = -\exp\left(-w_{novel} \min_{D \in \boldsymbol{D}} \|D_\pi(\boldsymbol{s}) - \boldsymbol{s}\|^2\right)$$

Task-Novelty Bisector (TNB)

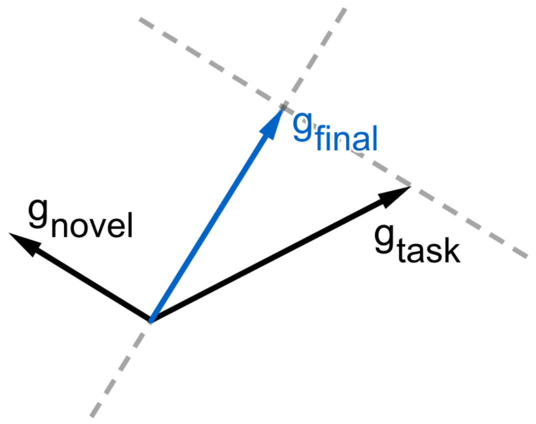
- Compute policy gradients for task reward and novelty reward

$$g_{task} = \frac{\partial J_{task}}{\partial \theta} \qquad g_{novel} = \frac{\partial J_{novel}}{\partial \theta}$$

- Compute the final policy gradient using the following rules:

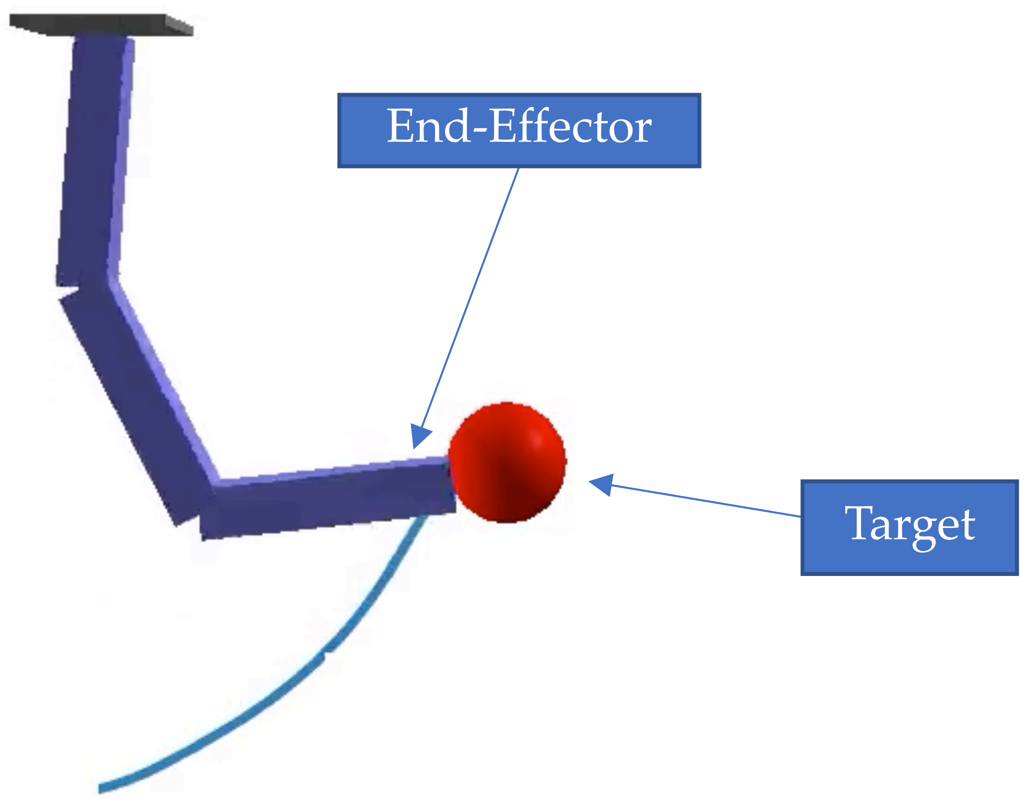


or

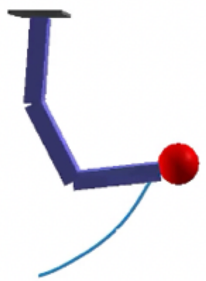


Multiple Solutions

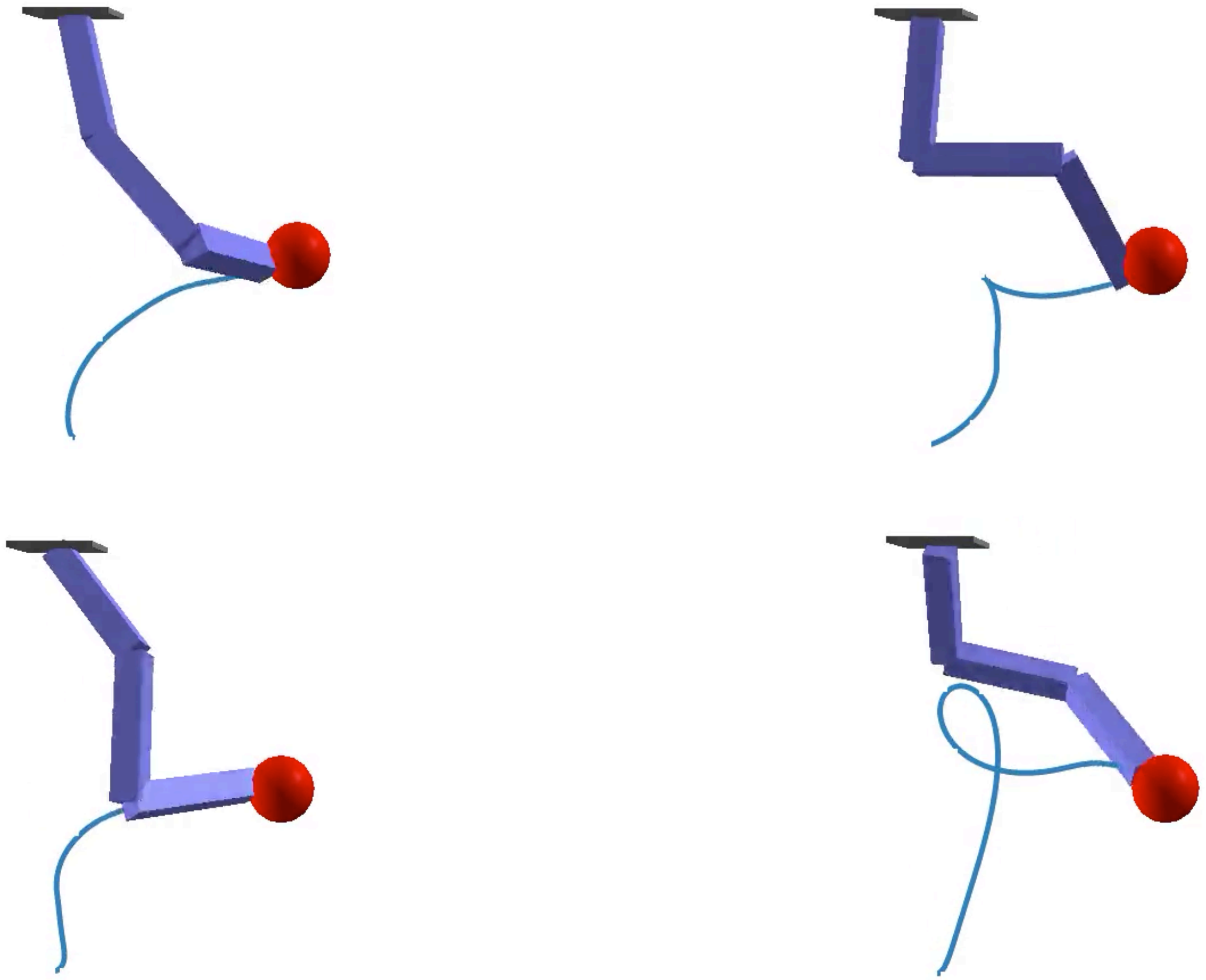
PPO Policy



Multiple Solutions

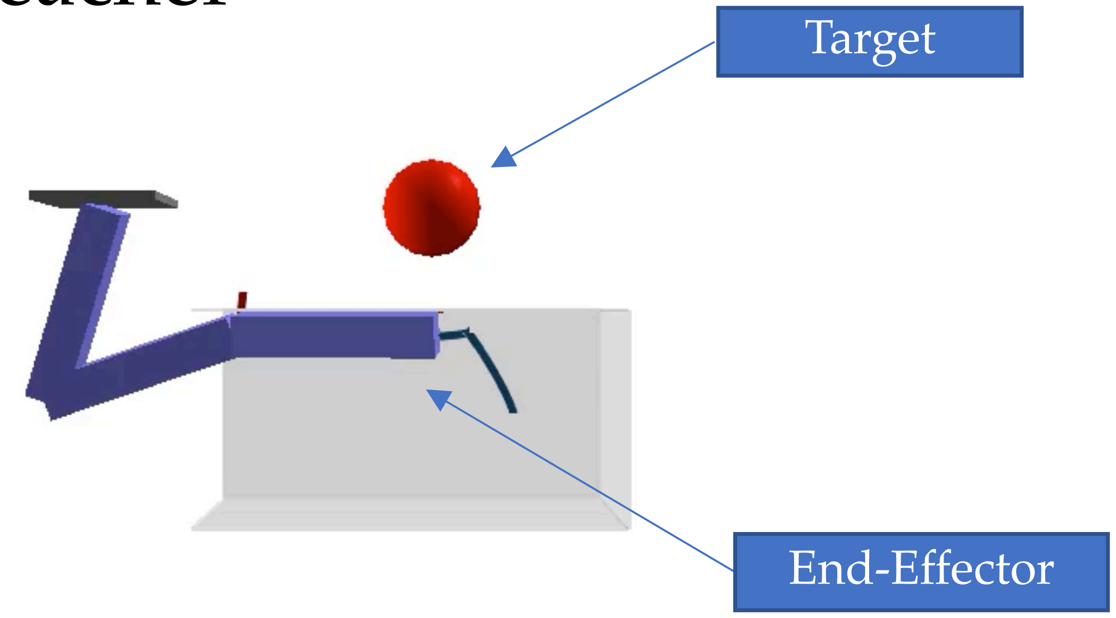


TNB Policies



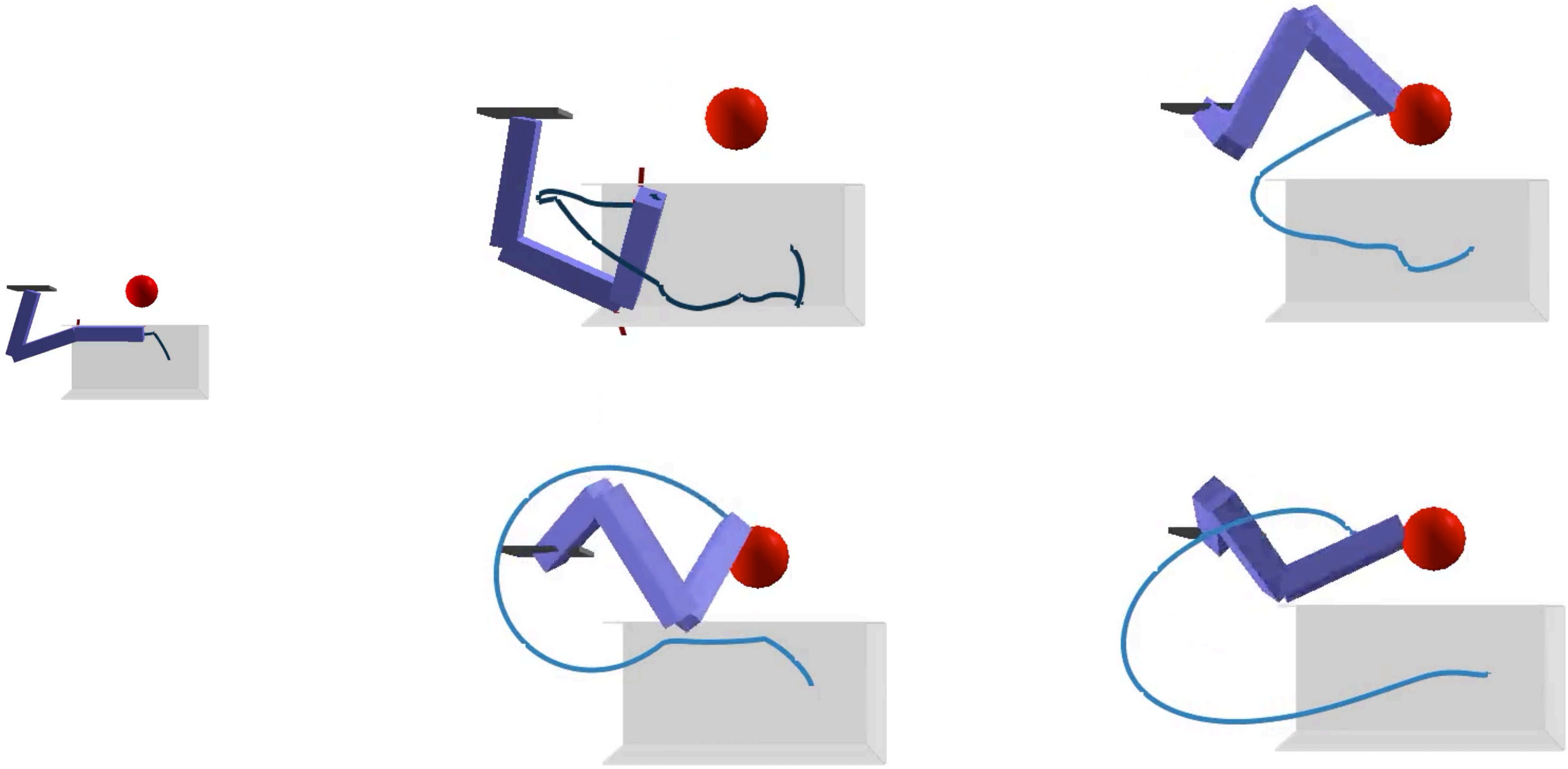
Deceptive Reward Problems

- Our methods could be further extended to solve tasks with deceptive reward signals.
  - E.g. Deceptive Reacher



Deceptive Reward Problems

TNB Policies



Thank You!

Poster: Pacific Ballroom #37