

# Reinforcement Learning for Path Planning of Robotic Arms

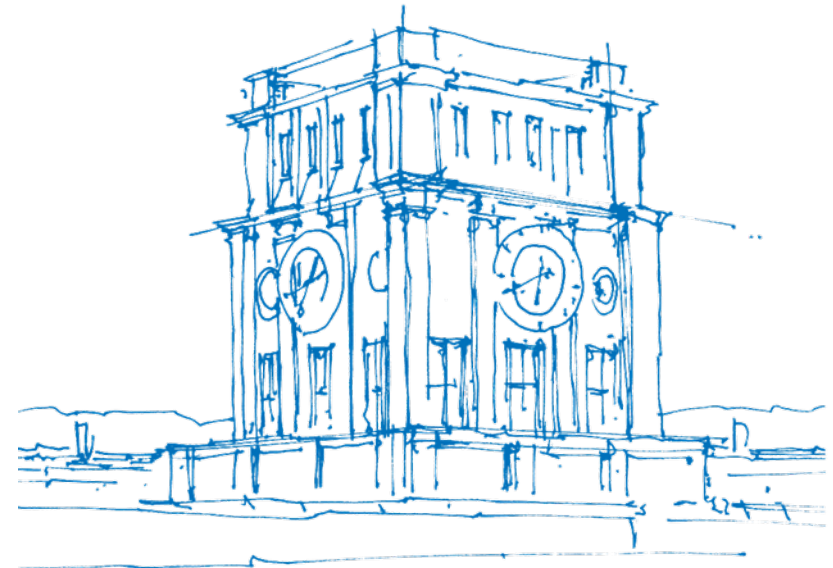
Anton Mai

Technische Universität München

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

München, 27. März 2015



*TUM Uhrenturm*

# Reinforcement Learning for Path Planning of Robotic Arms

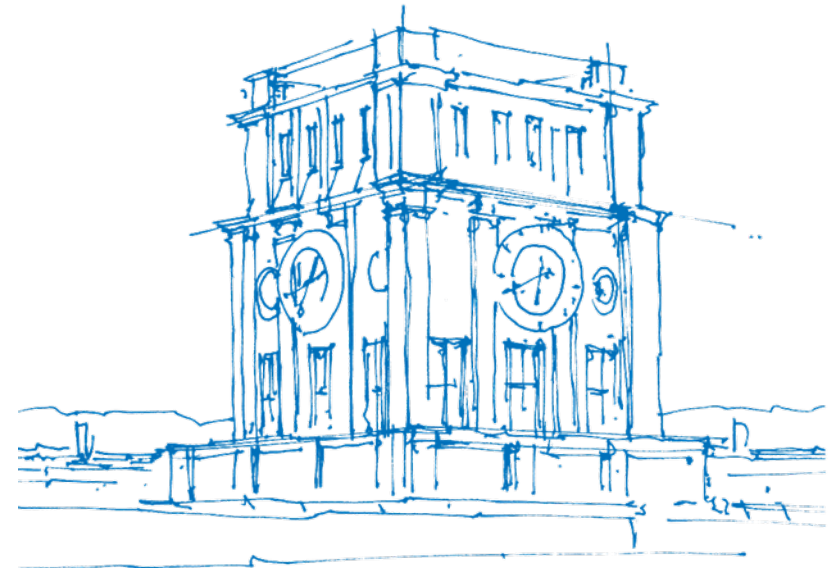
Anton Mai

Technische Universität München

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

München, 27. März 2015



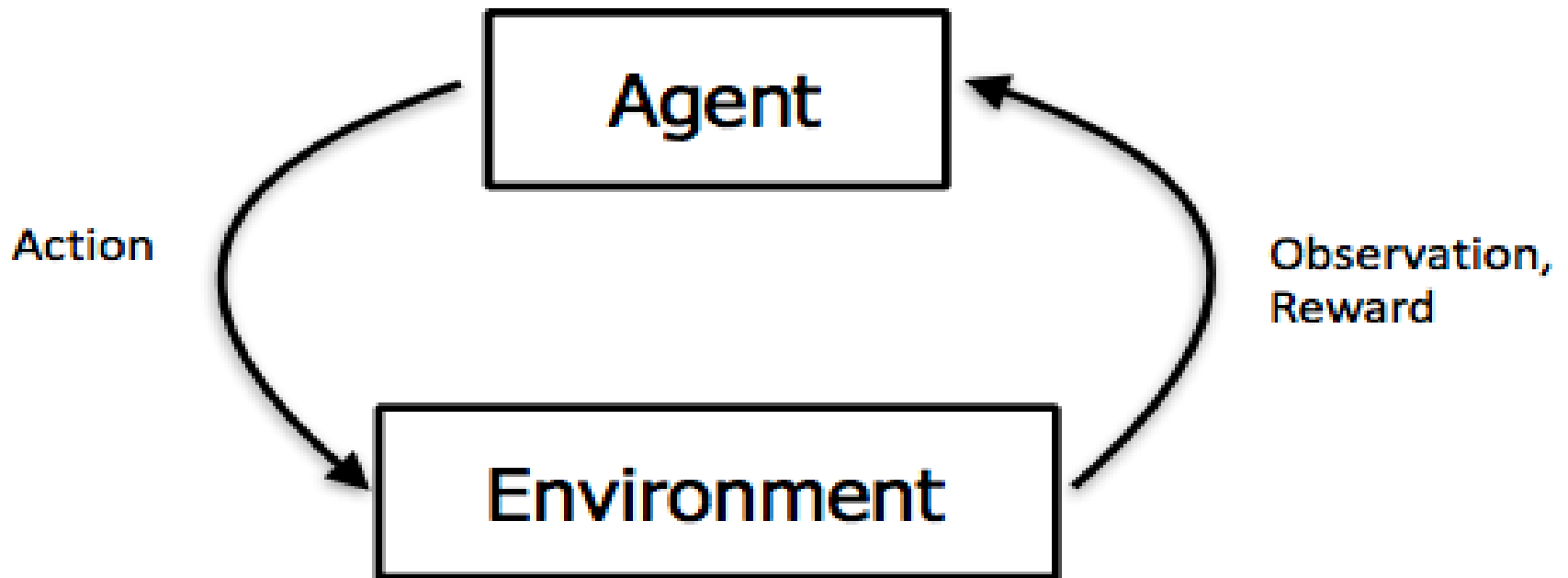
*TUM Uhrenturm*

# Outline

- Motivation
- Reinforcement Learning
- Hindsight Experience Replay
- Related Works
- Environment 1: FetchSlideball
- Environment 2: FetchToss



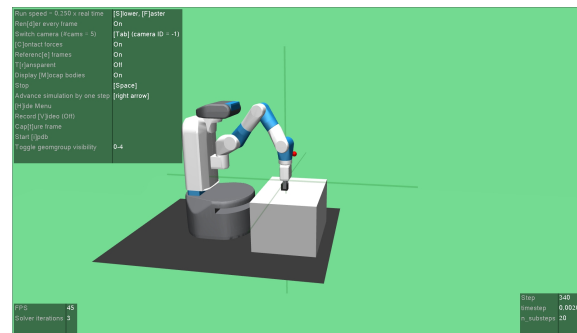
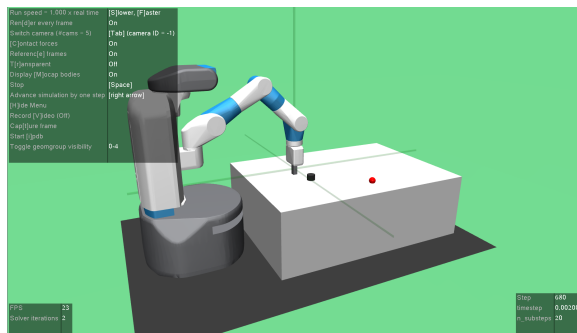
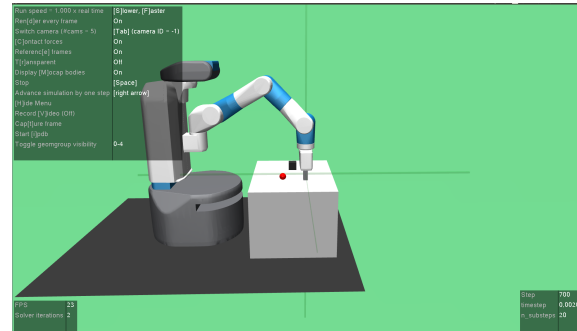
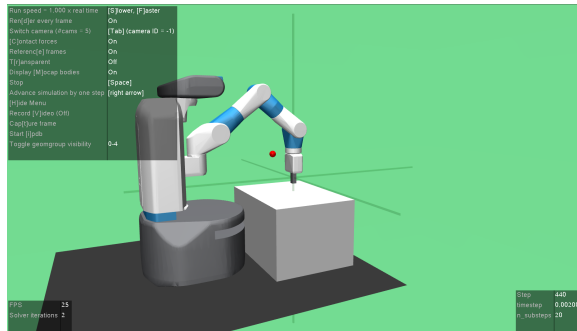
# Reinforcement Learning



# Hindsight Experience Replay

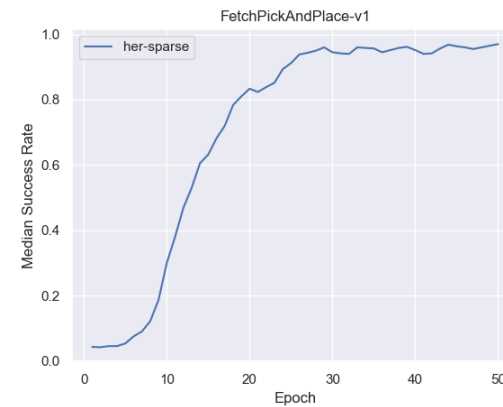
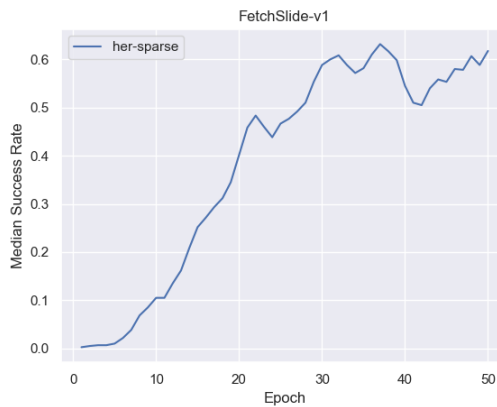
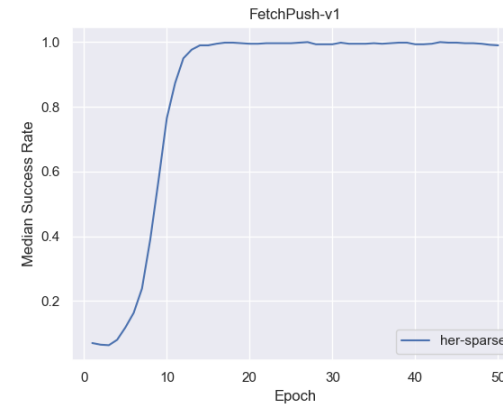
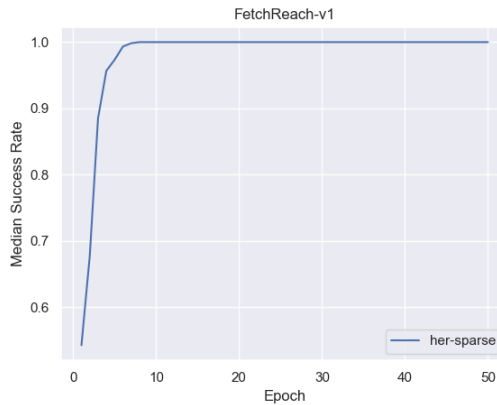
# Some more HER ideas if I use them

# Benchmarks



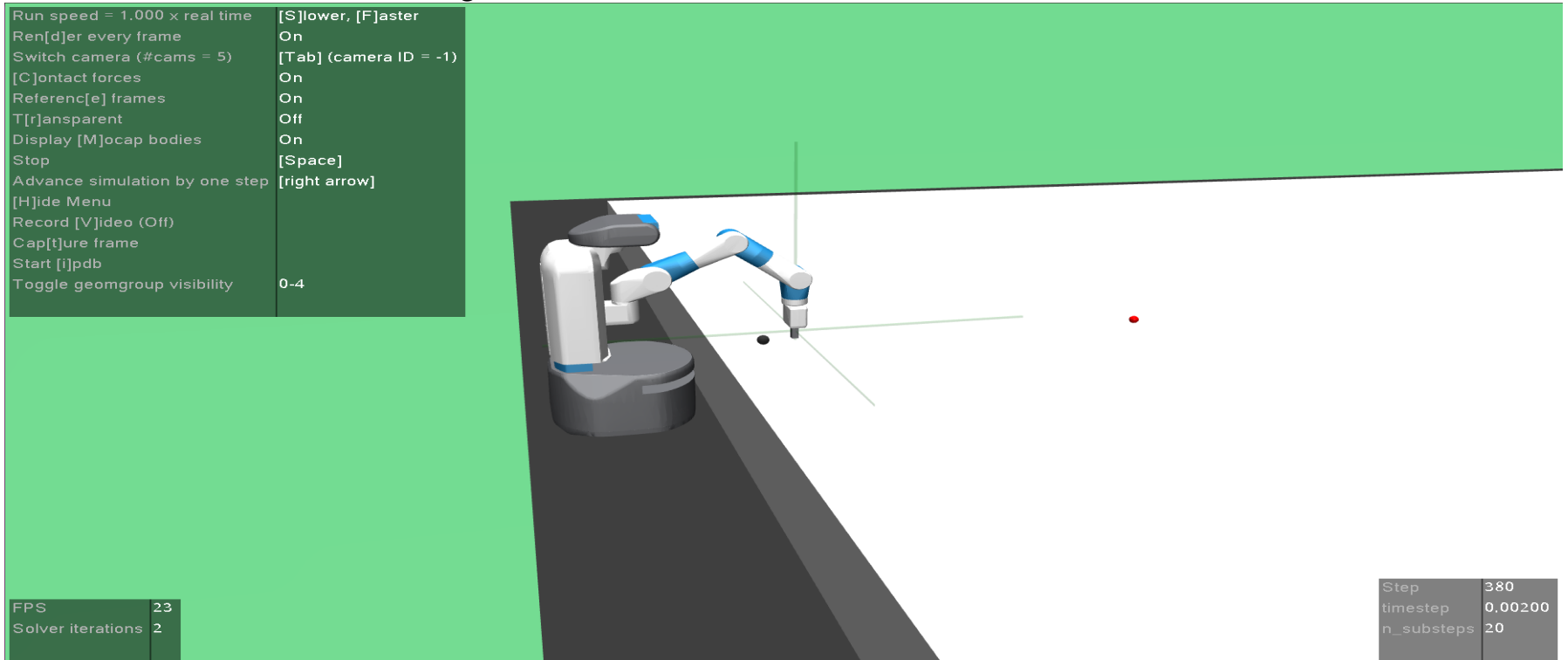


# Benchmarks



# FetchSlideball

## Environment Picture Extending FetchSlide

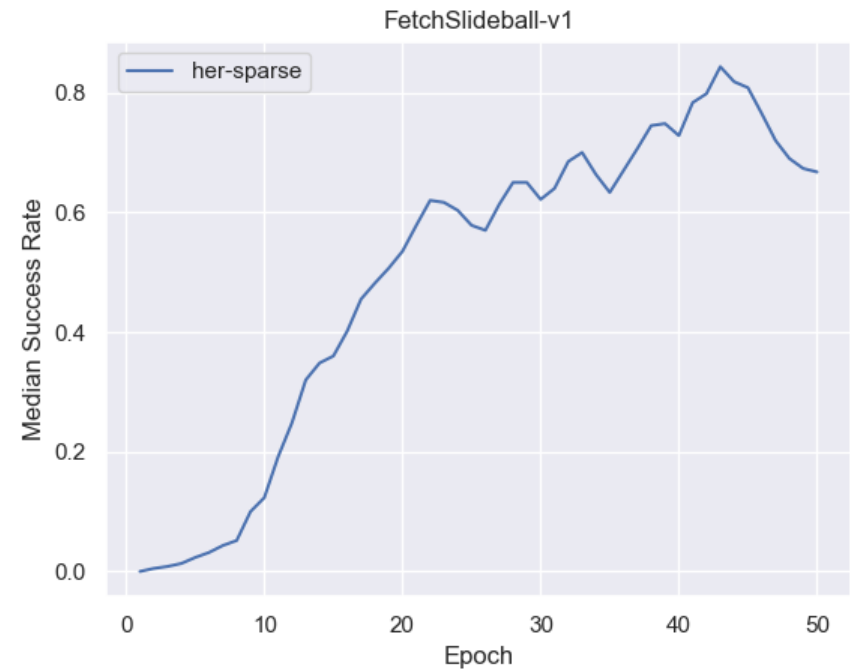
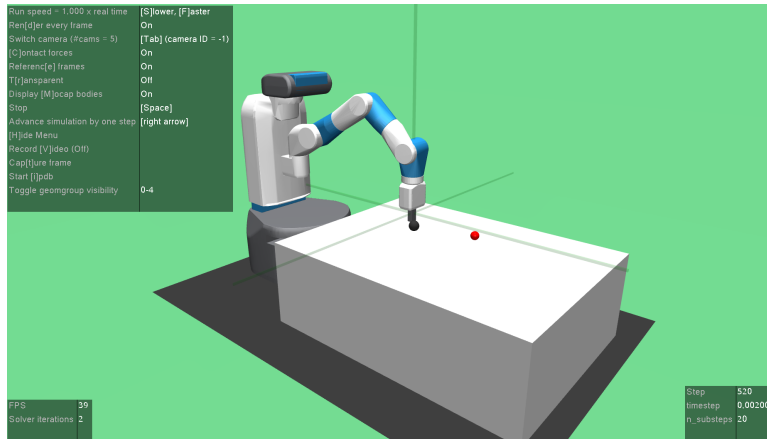


Goal: bigger distance

Anton Mai (TUM) | RL for Robotic Arms

# FetchSlideball Version 1

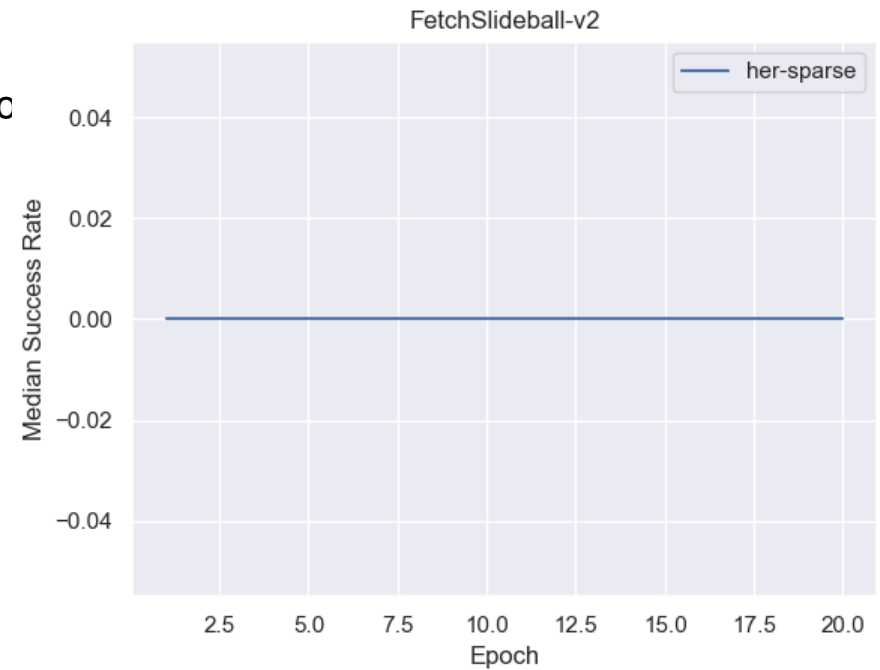
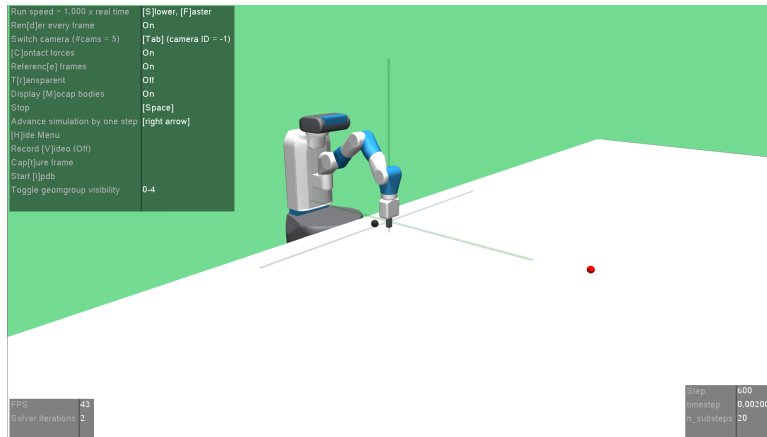
Same Distance as FetchSlide-v1, but with a ball



# FetchSlideball Version 2

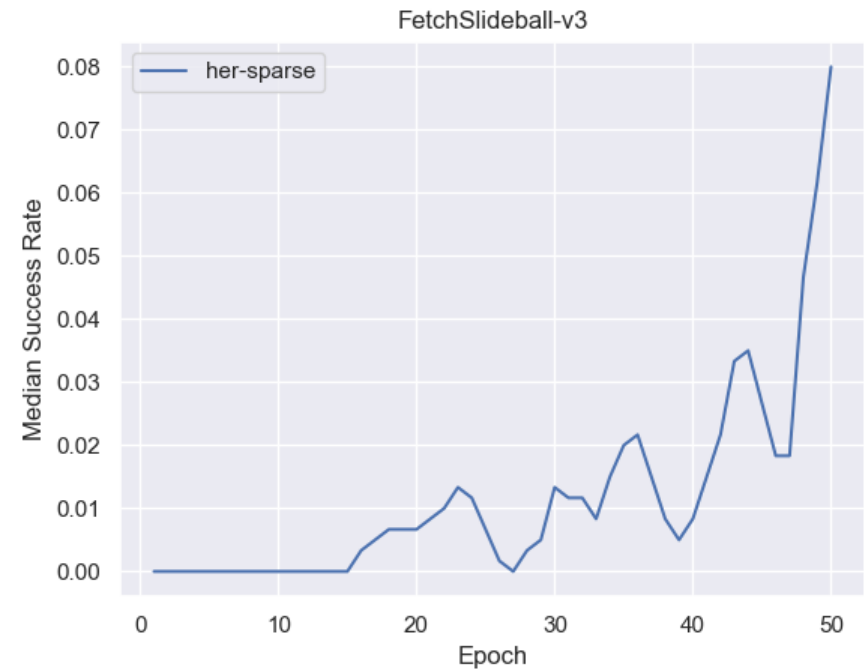
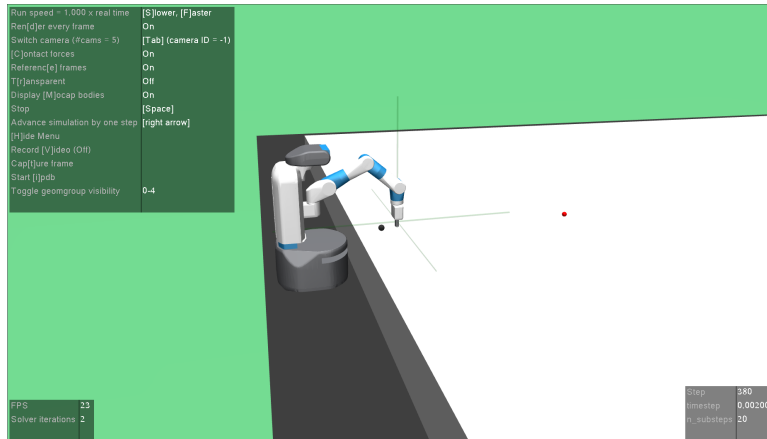
Doubled the distance

Problem: Arm not strong enough too push it far end



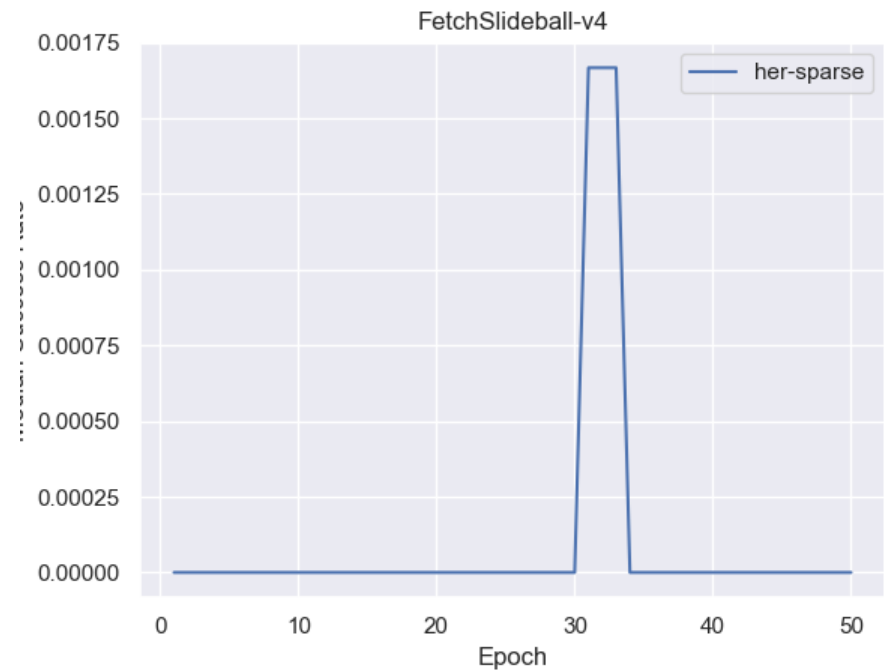
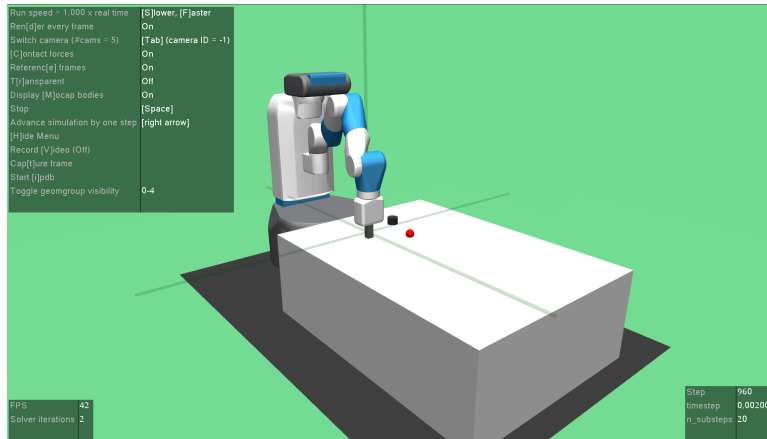
# FetchSlideball Version 3

10% friction



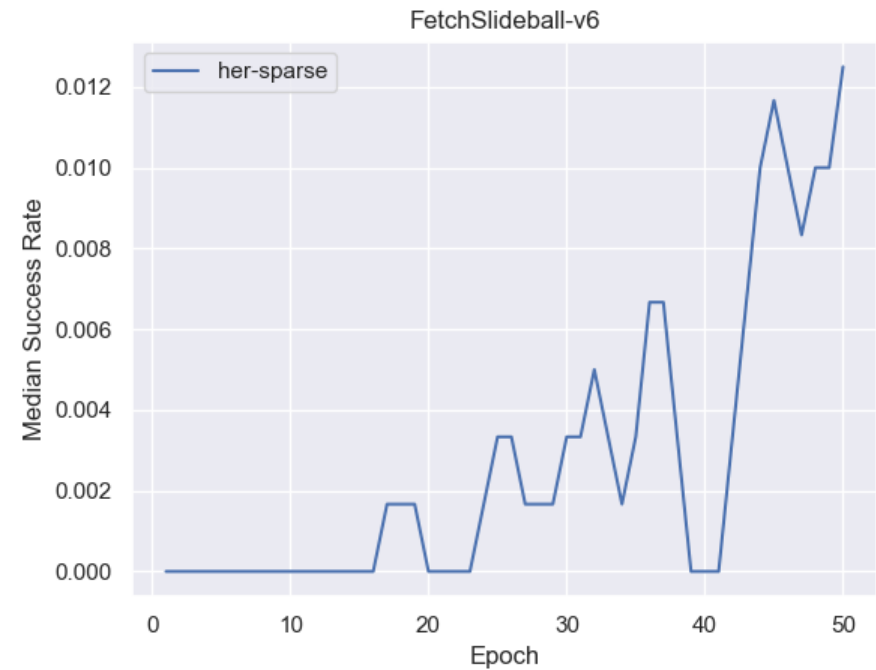
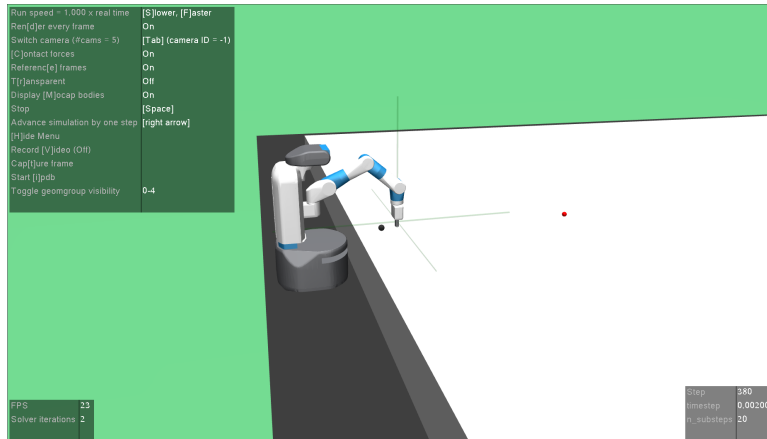
# FetchSlideball Version 4

50% friction



# FetchSlideball Version 6

25% friction

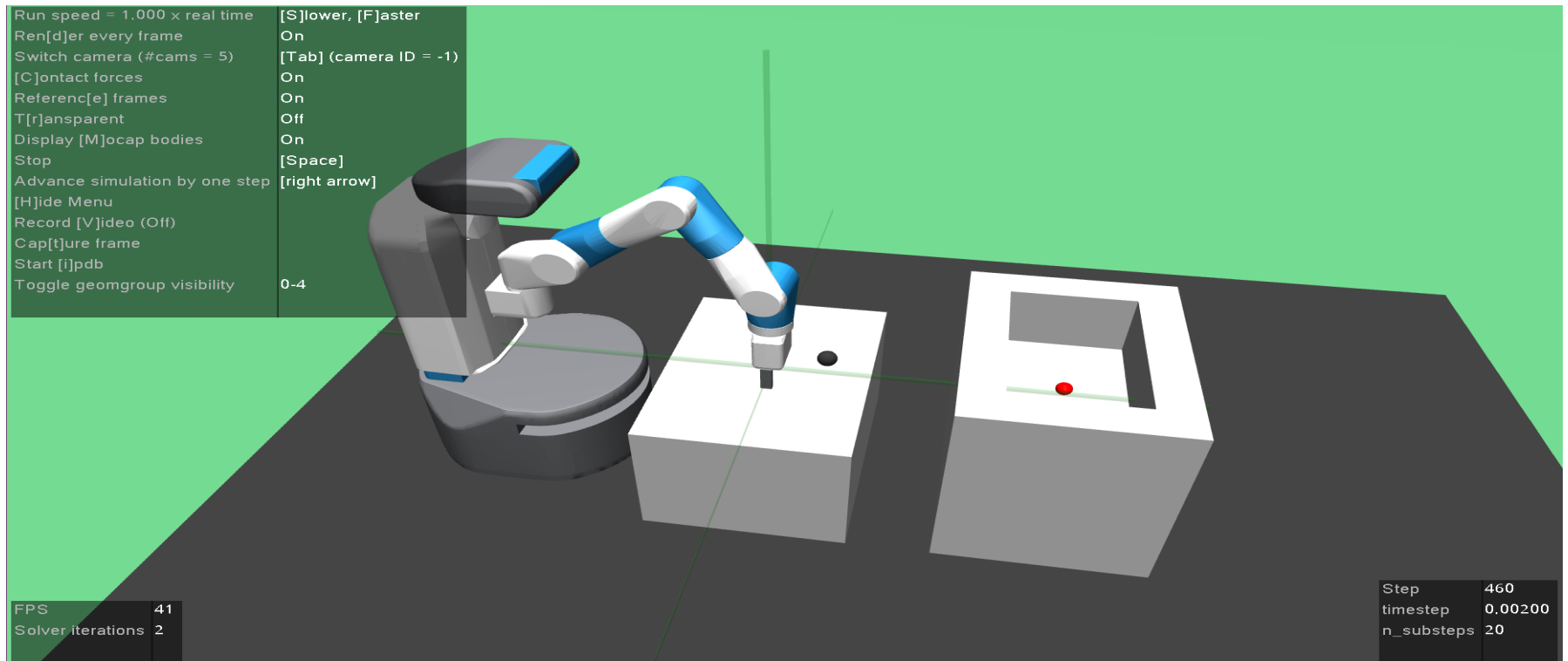


# FetchSlideball

Show results, all in one page. Explain how I did these things

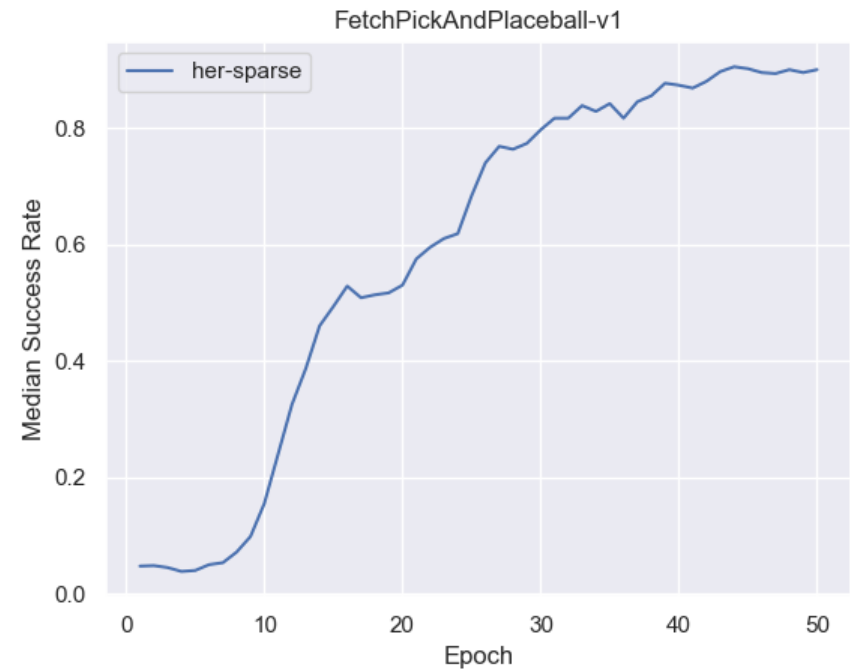
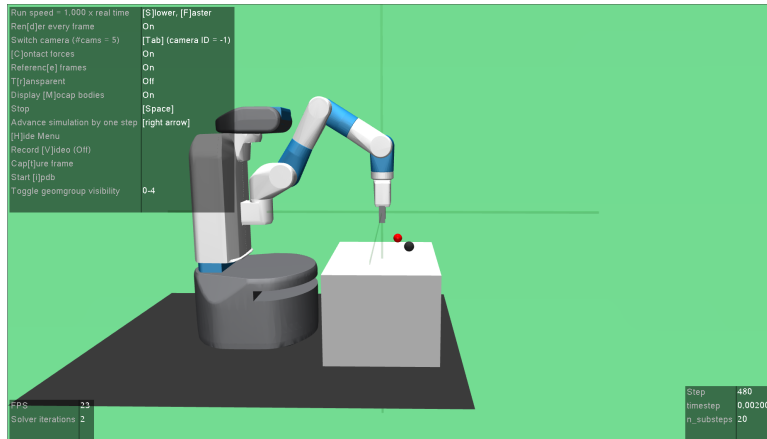


# FetchToss



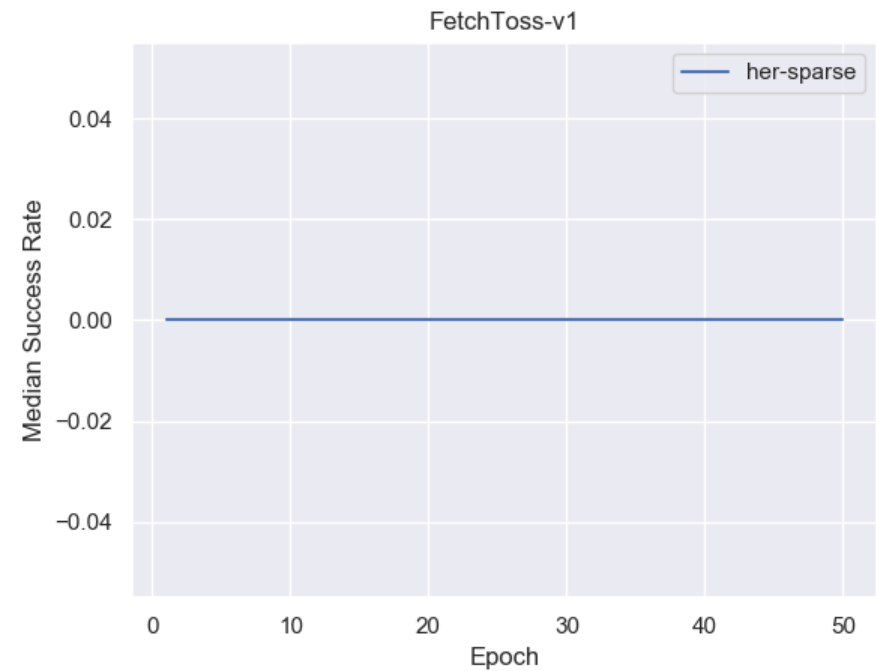
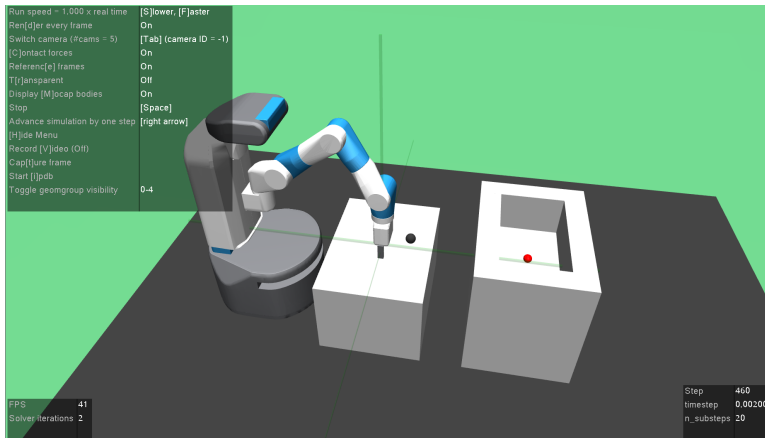
# FetchPickAndPlaceball

Testing PickAndPlace with a ball for comparison



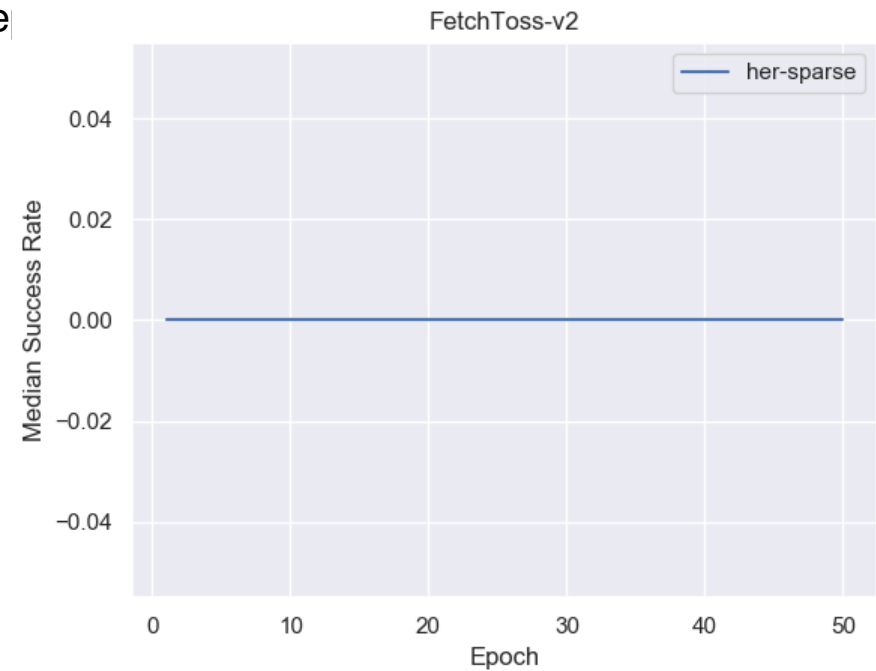
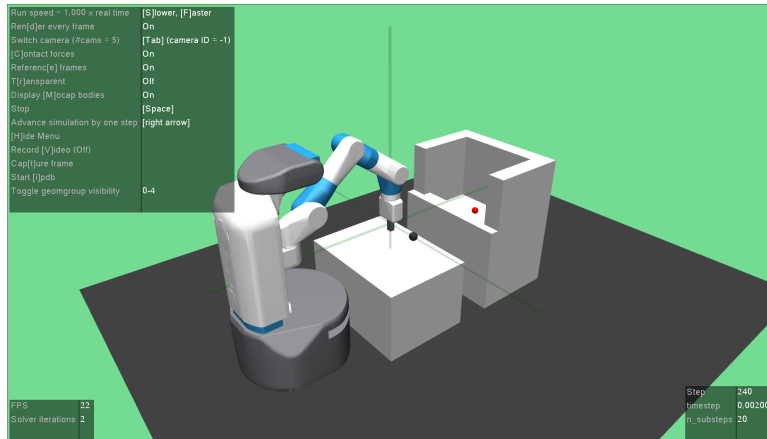
# FetchToss Version 1

Goal is only in the box



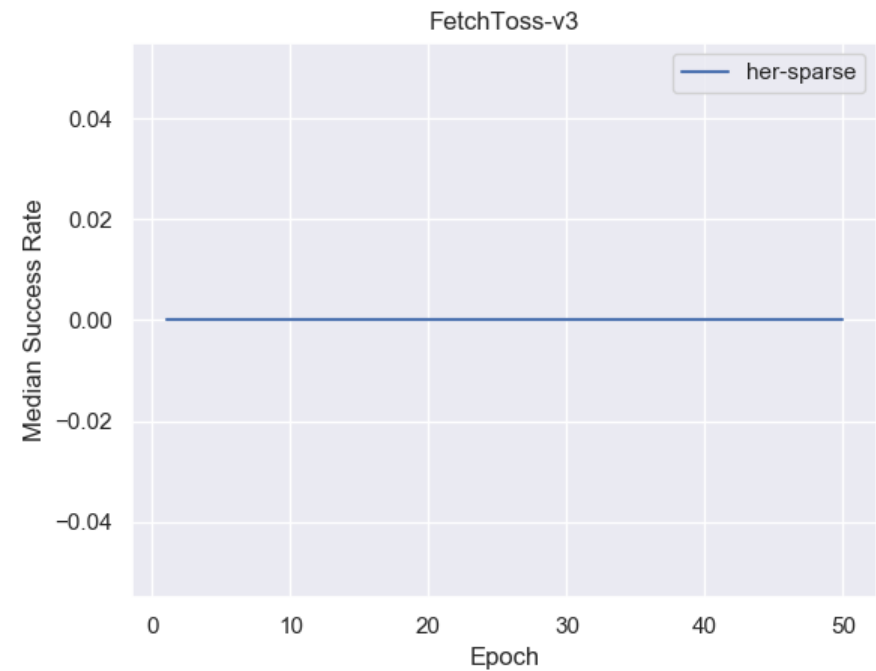
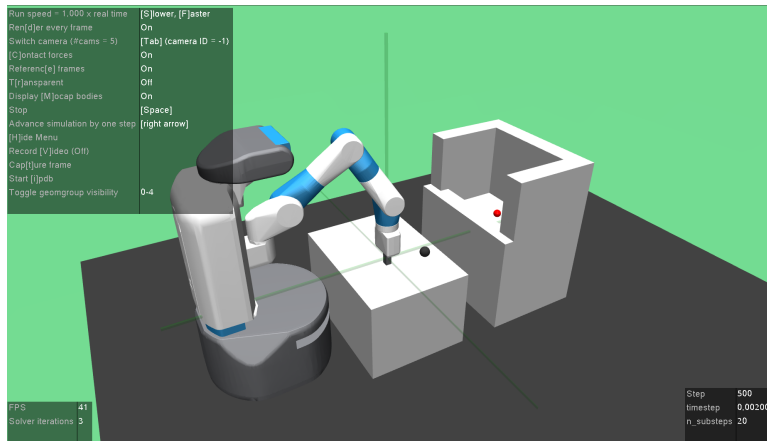
# FetchToss Version 2

Better box, some friction added, double training steps



# FetchToss Version 3

1% weight on the ball

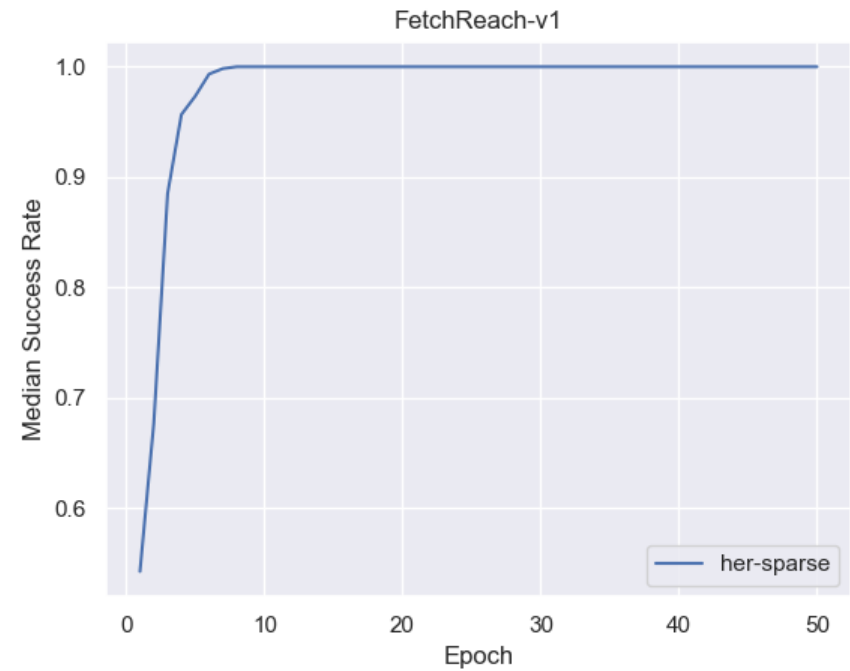
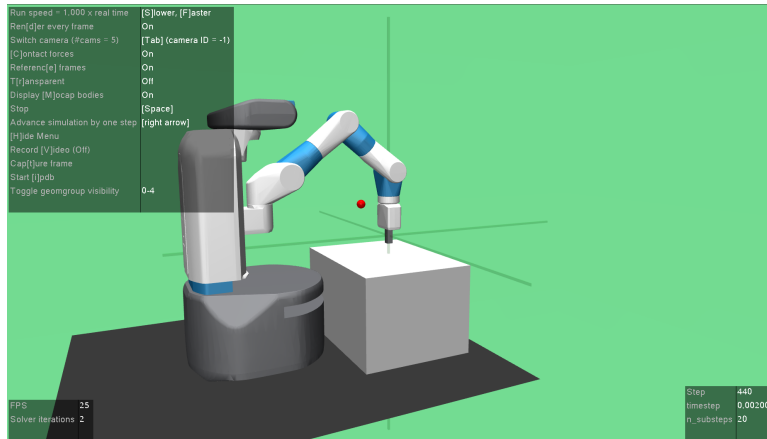


# Presentation Sources

<https://blog.goodaudience.com/what-is-inverse-reinforcement-learning-e333228af146>

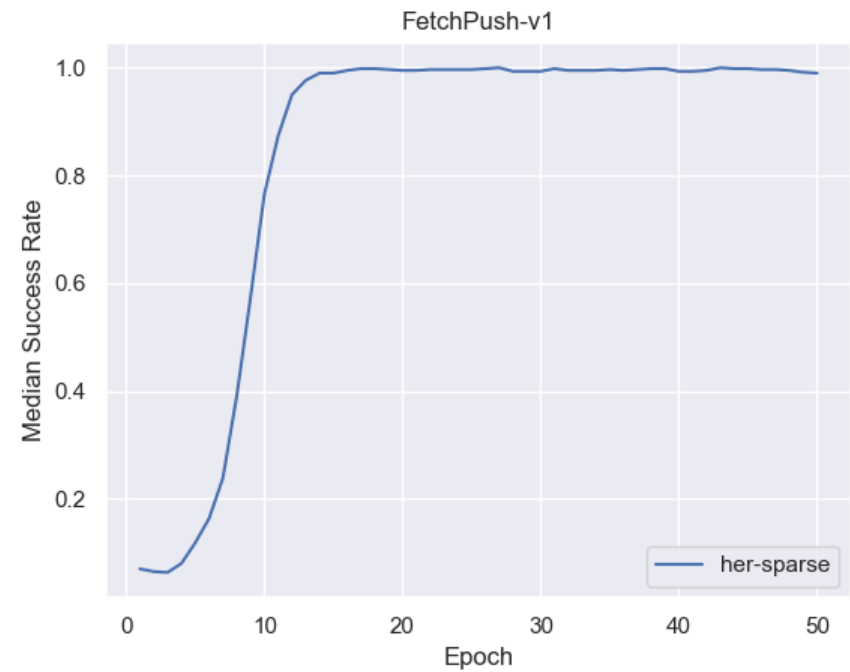
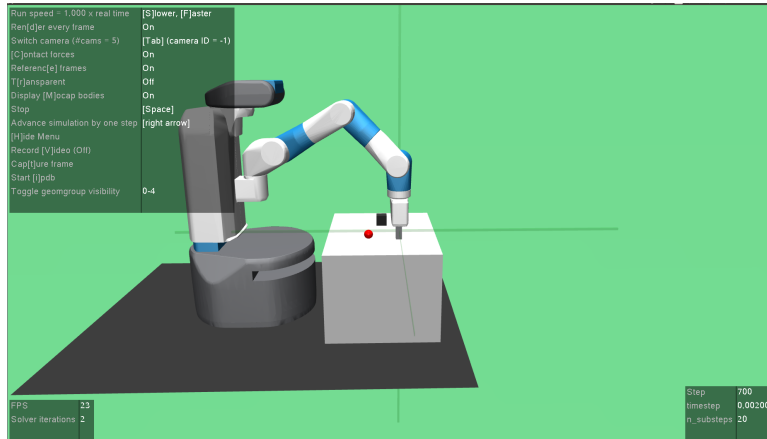
# Backup Slides

# FetchReach-v1

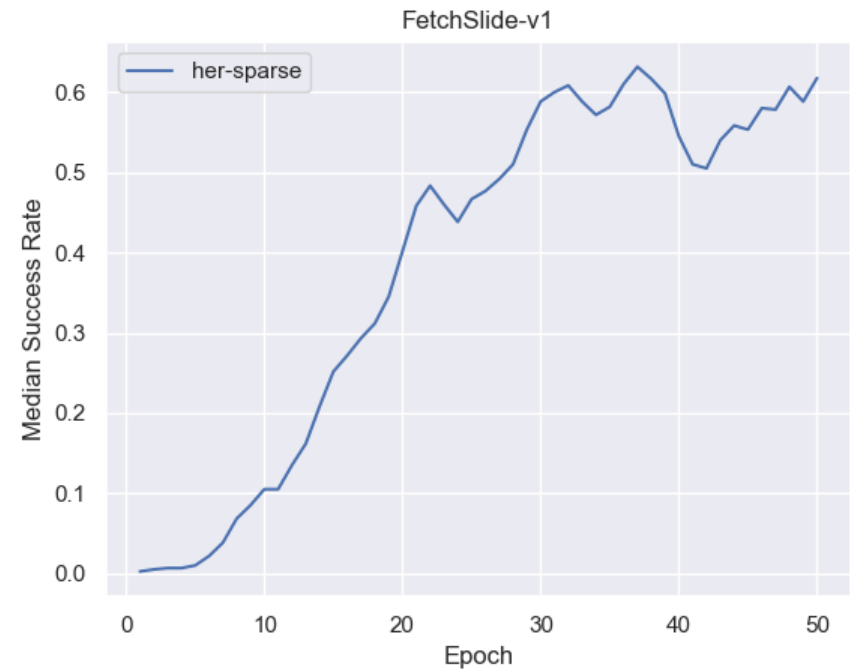
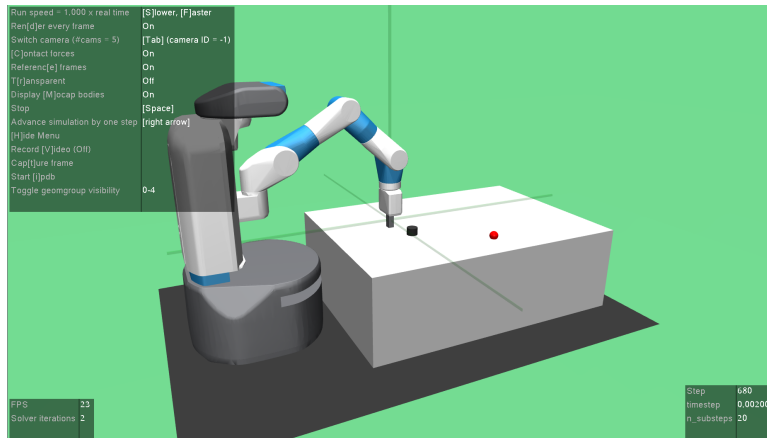




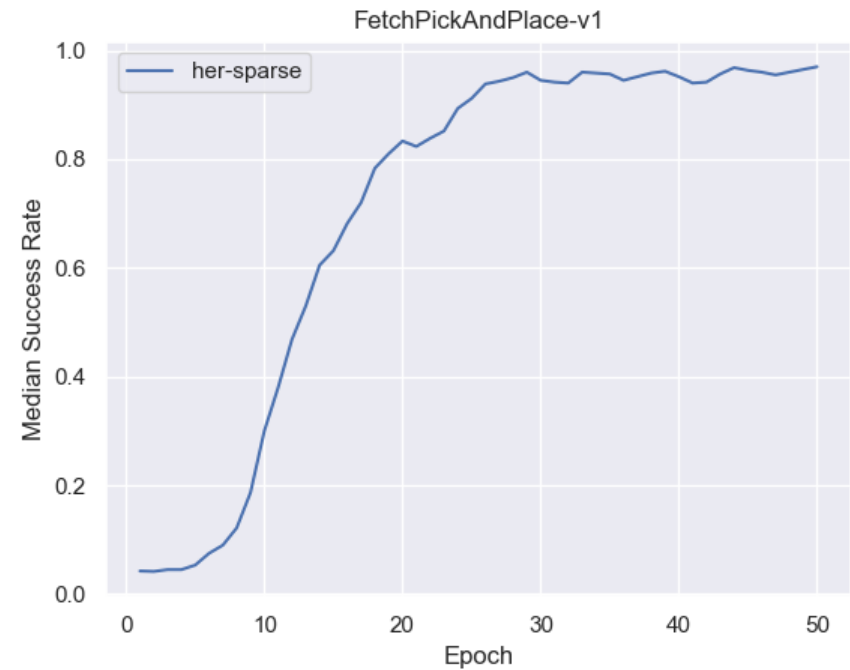
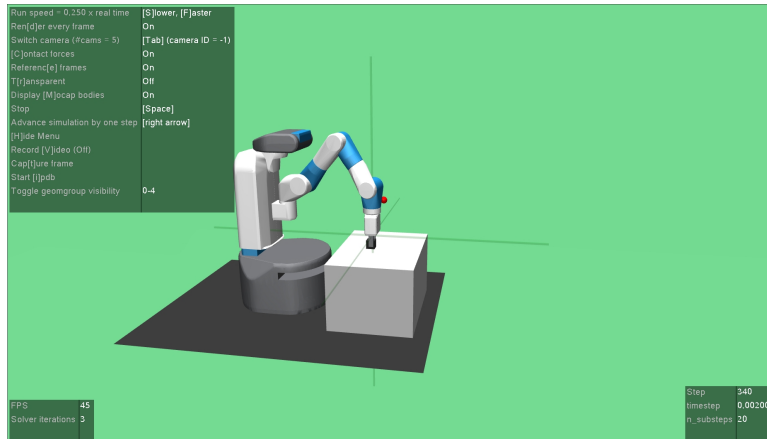
# FetchPush-v1



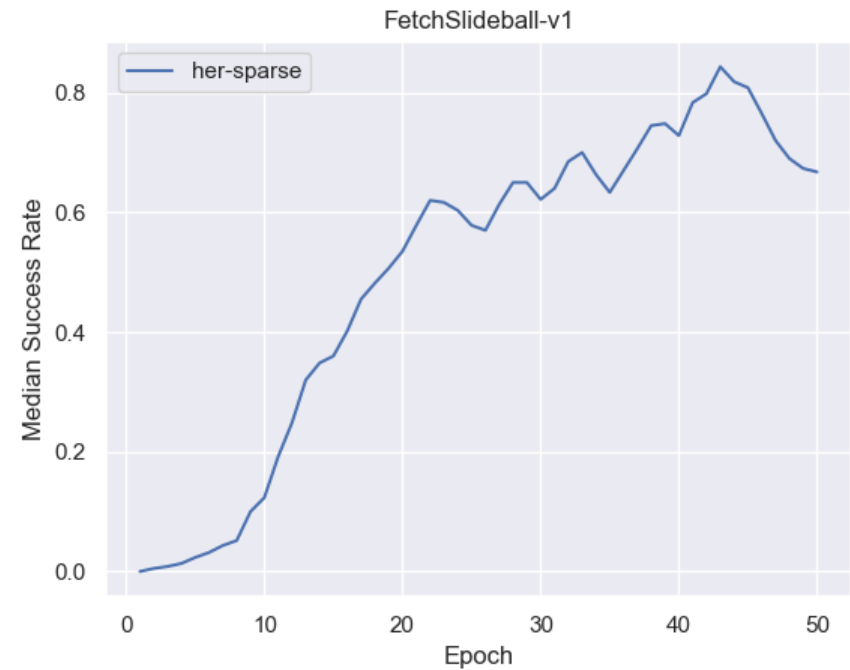
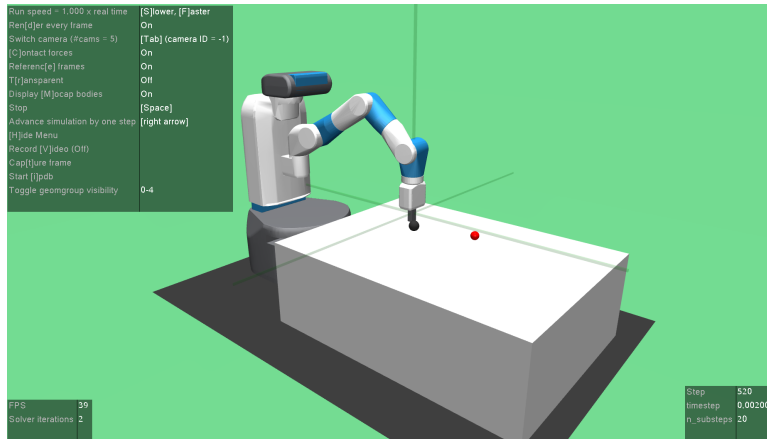
# FetchSlide-v1



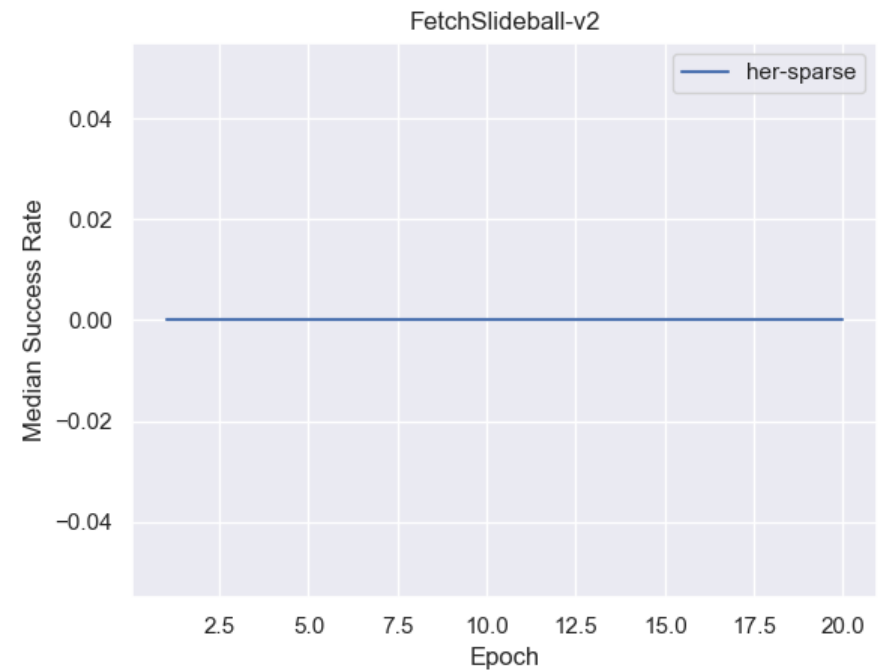
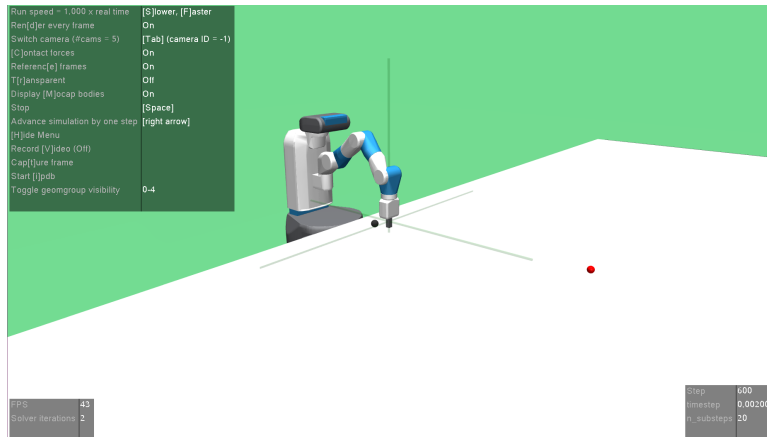
# FetchPickAndPlace-v1



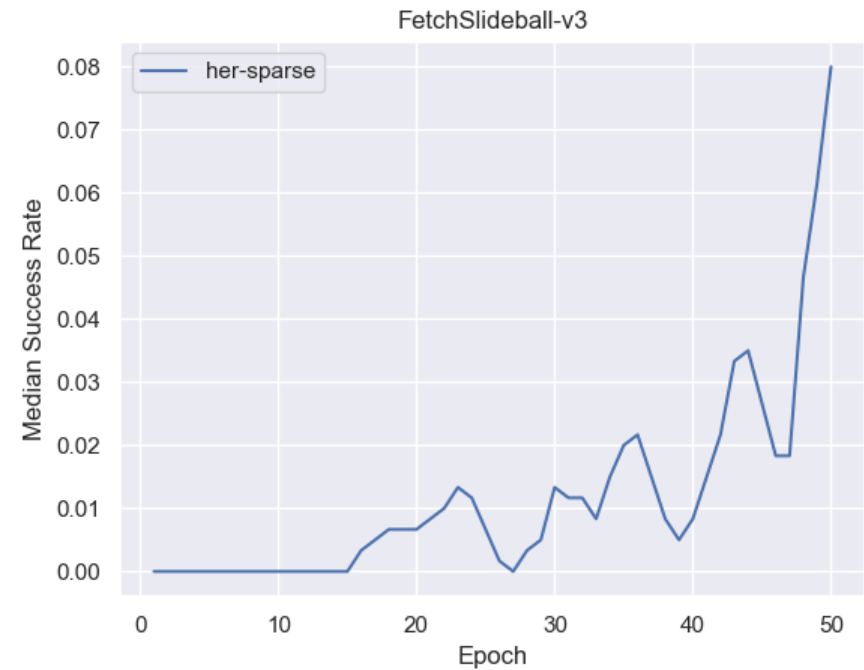
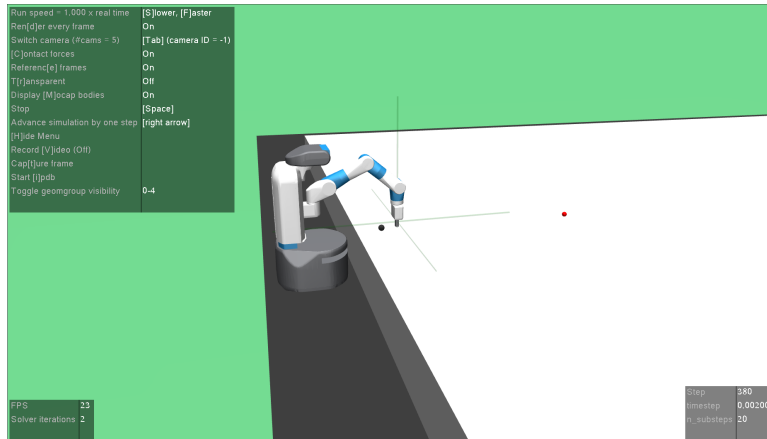
# FetchSlideball-v1



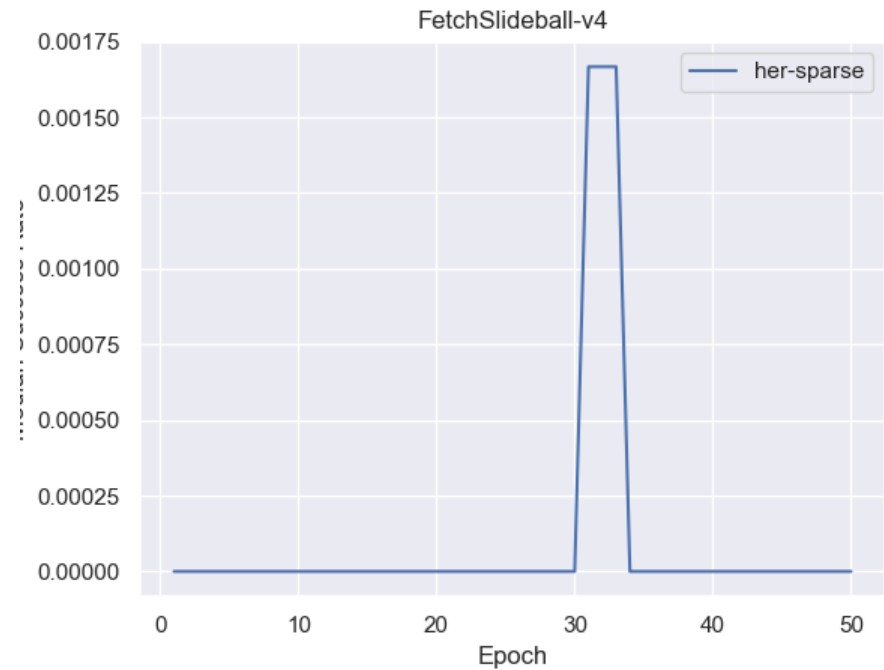
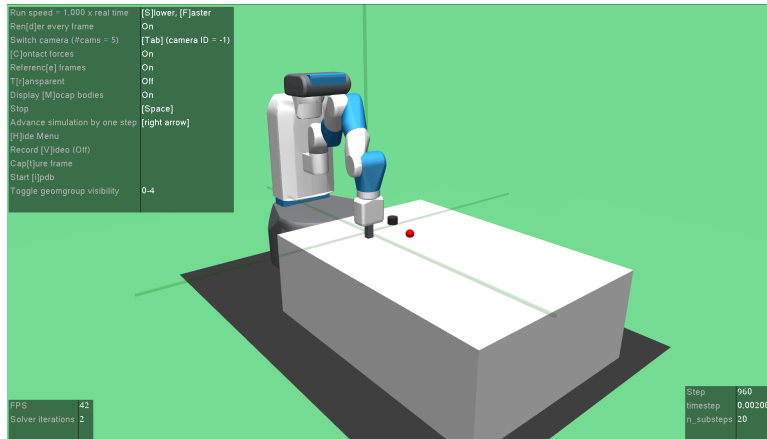
# FetchSlideball-v2



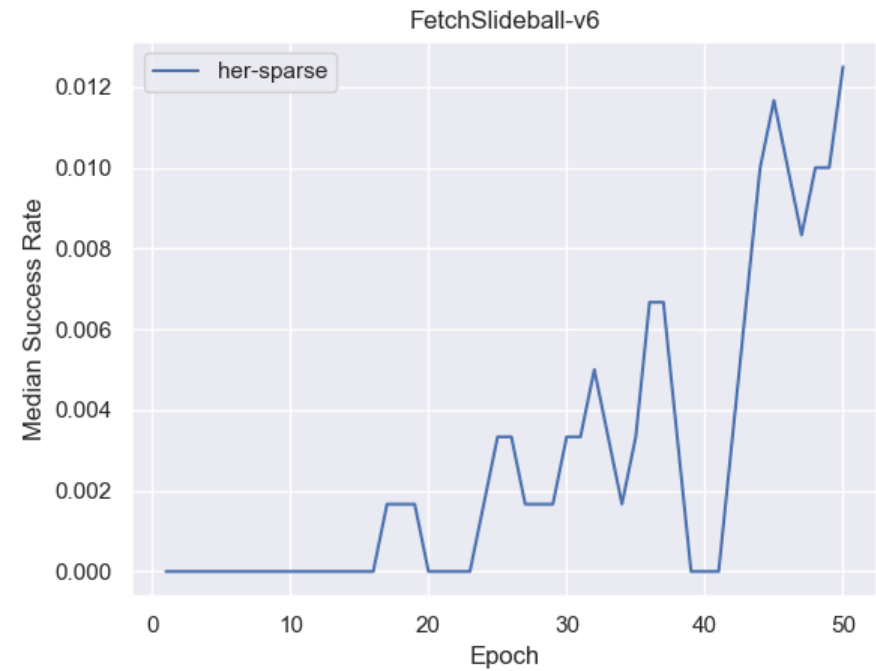
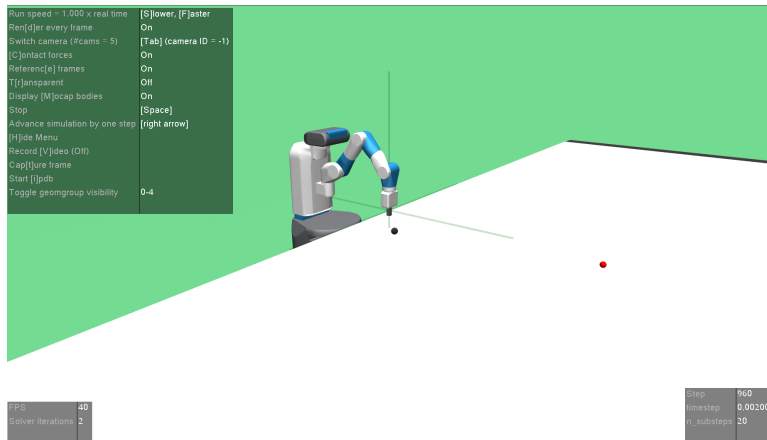
# FetchSlideball-v3



# FetchSlideball-v4

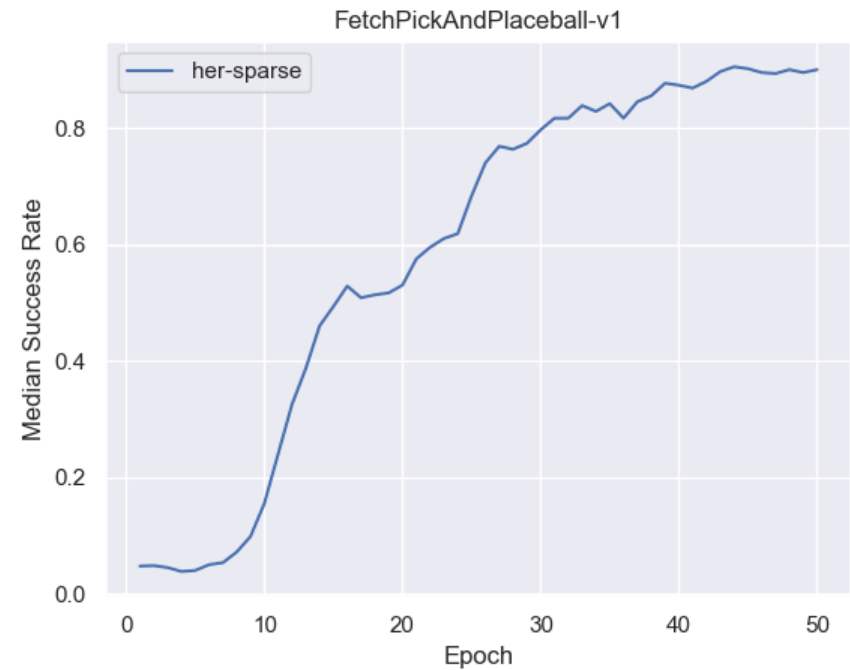
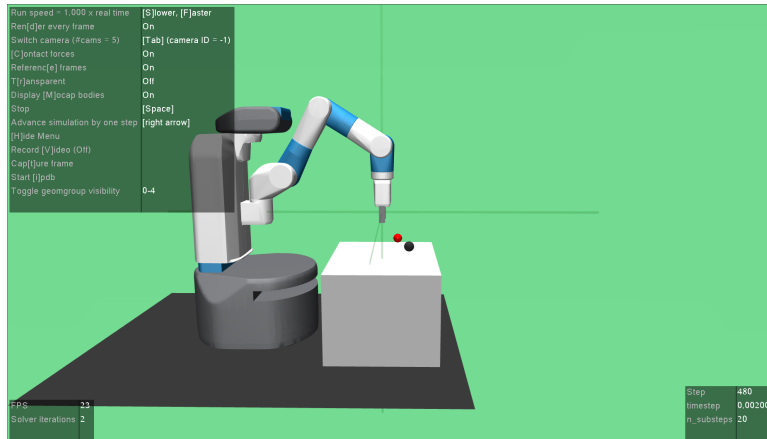


# FetchSlideball-v6

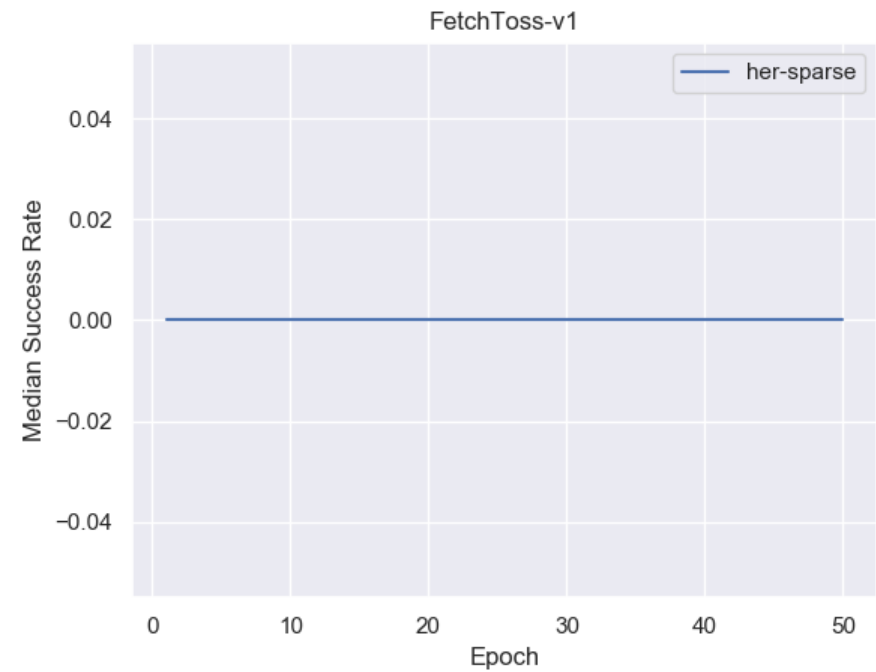
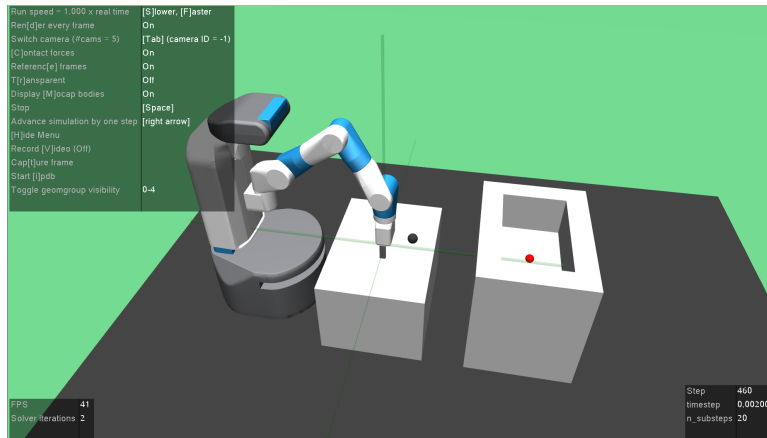




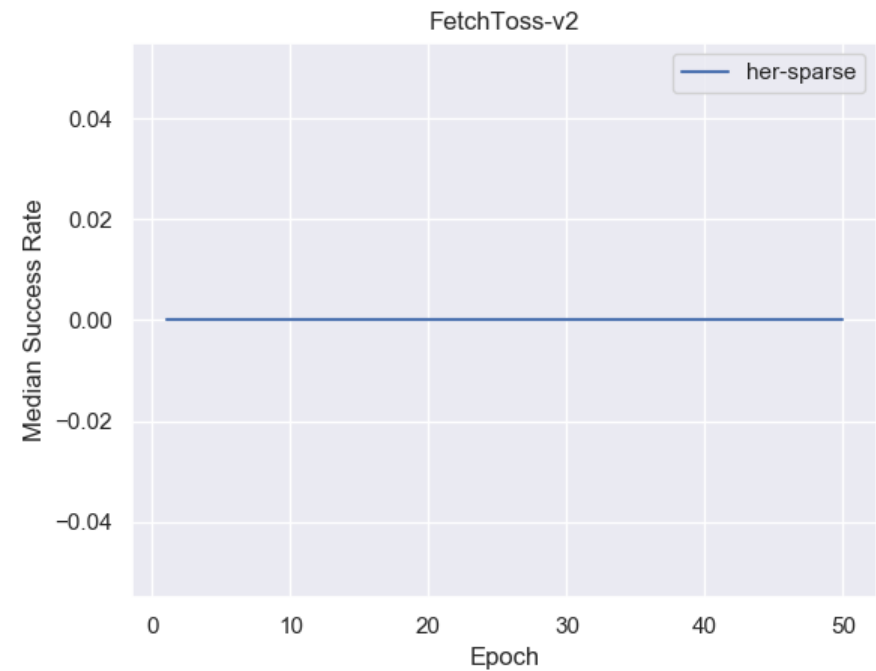
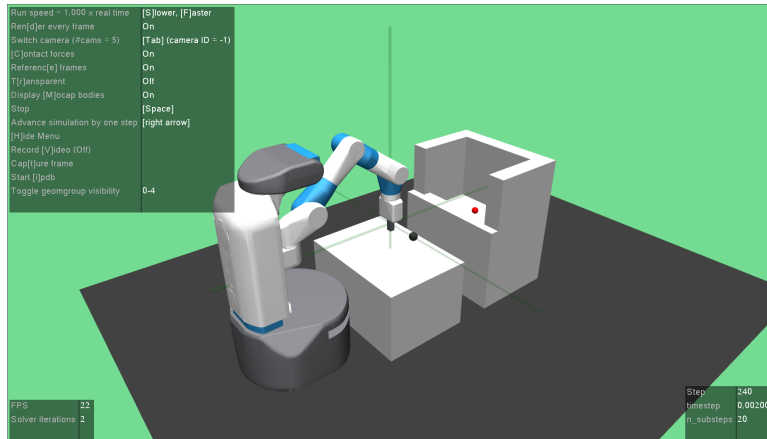
# FetchPickAndPlaceball-v1



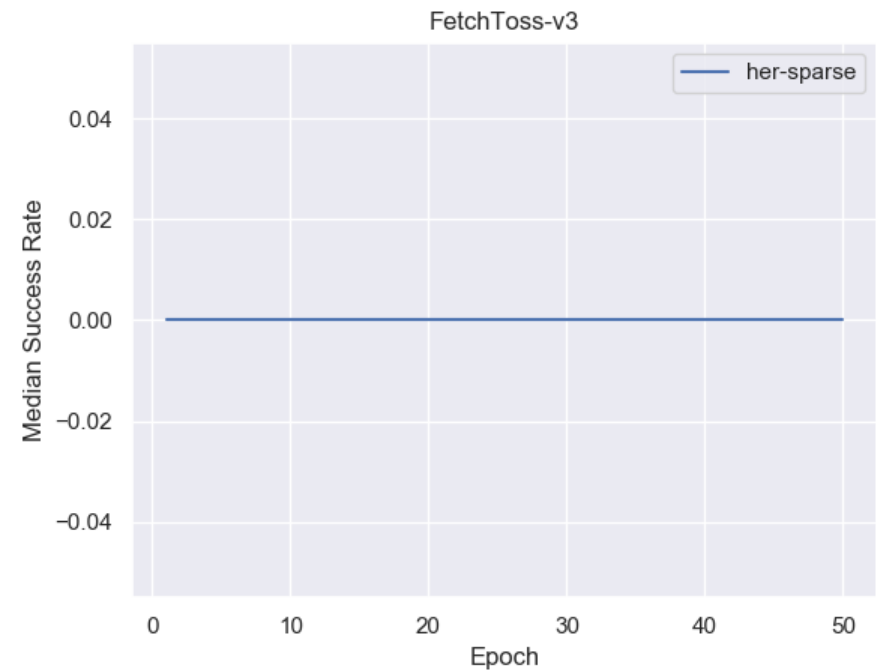
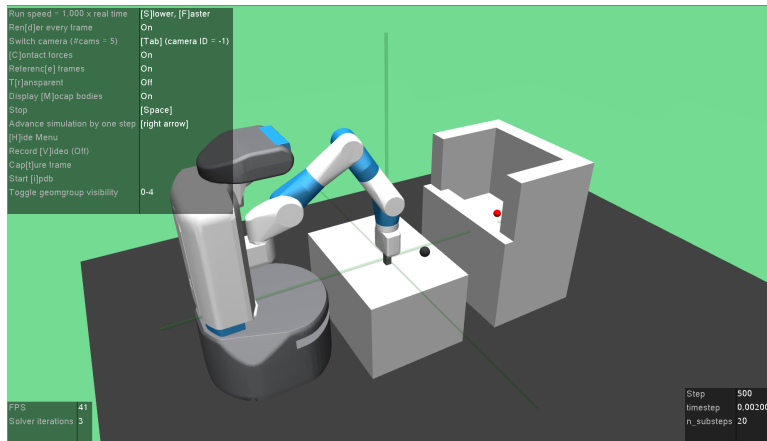
# FetchToss-v1



# FetchToss-v2



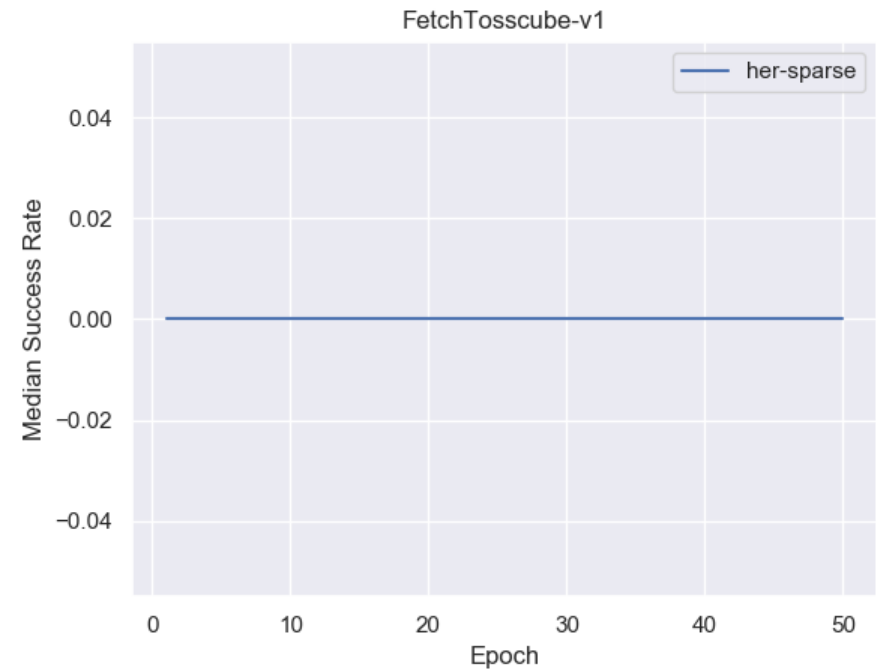
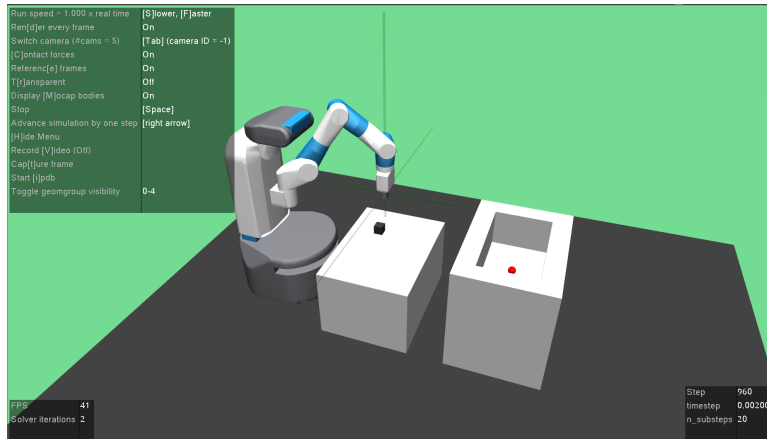
# FetchToss-v3



# FetchToss-v0 (Tossing is possible)

# FetchTosscube-v1

Tossing a cube (with v3 parameters)



Anton Mai

Technische Universität München

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

München, 27. März 2015



# Präsentationsmuster

kann auch als Kapiteltrenner verwendet werden