Deep Reinforcement Learning

TD3 in OpenAI Gym

by

Martin Baur
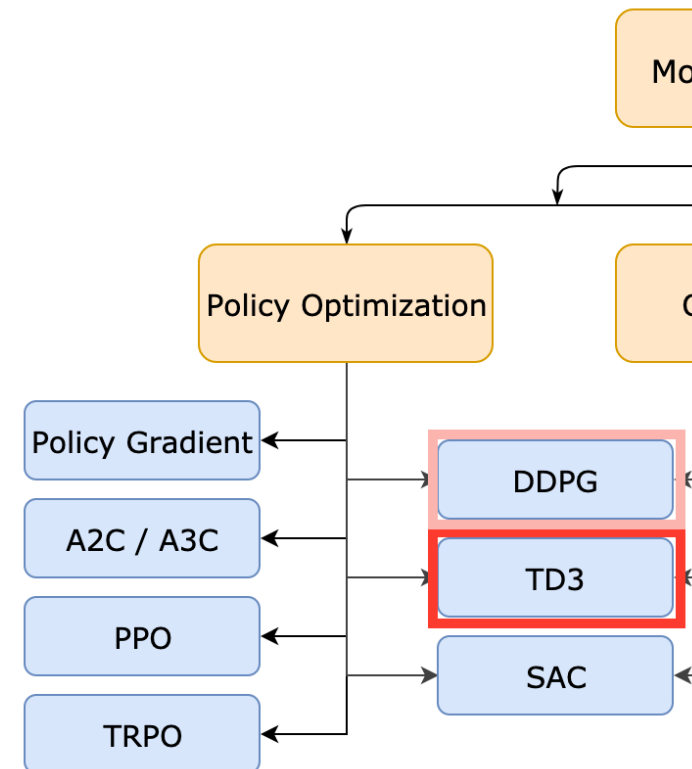
mail: mail@martinbaur.eu

Agenda

- TD3 algorithm

    – What to learn?

    – DDPG vs. Twin Delayed DDPG

    – Three Tricks

- Experiment setup

- Experiment: 2D Swing a Pendulum

- Experiment: 3D Ant

- Literature

TD3

- Original Paper published 22 October 2018

- Twin Delayed Deep Deterministic Policy Gradient

- Successor of DDPG

- Off-policy [1]

TD3 – What to learn: Q-Learning

- Q-table

- Q-Learning

  - How it works:

    - Selection of random state (s) and taking random action (a)

    - Next state reached (s')

    - Getting a reward (r)

    - Updating Q-Value

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} \right)$$

Q-learning update function [3]

  - maximum expected future rewards for each action in each state

TD3 – What to learn: Q-Learning II

- Here: Deep Q-Learning

  – Not iterativ but a neural network

- How it works:

  – Predicts the Q-Values of current state

  – Take action with highest Q-Value

  – get reward

  – Reach s'

  – Q-loss is calculated with target and prediction

  – Goal: reduce loss

Source: [7]

TD3 – What to learn: Policy Learning

- Second neural network: Policy Gradient

- Updates policy weights

- Input: states

- Output: actions with probability (agent chooses)

TD3 - DDPG vs. Twin Delayed DDPG

- TD3 is a direct successor of Deep Deterministic Policy Gradient (DDPG)

- DDPG:

  – Base paper published in 2014 but first described in 2015 [8]

  – Learns Q-function and policy

  – „DDPG is an off-policy algorithm.

  – DDPG can only be used for environments with continuous action spaces."[4]

- Problems with DDPG:

  – overestimates Q-values because of exploitation of errors in Q-function

TD3 - Three Tricks

Solutions for DDPG Issue:

- **Twin:** Double-Q Learning

  – Learns two Q-functions

  – Smaller is used to update the Q-value

  – -> favors underestimation of Q-values

- **Delayed:** Delayed Policy Updates

  – Policy Updates are delayed

  – Usally two Q-function updates and one policy update

  – -> more stable and efficient training

- Target Policy Smoothing

  – Adding noise to action to make Q-function error exploitation harder

  – -> more robust actions are chosen

Experiment Setup

- Algorithm: TD3 of stable-baselines

- Pendulum Env from OpenAI Gym

- Ant Env from PyBullet Gym

- Running on Google Colab

- Logging with Tensorboard

Experiment: Swing a Pendulum

- „ In this version of the problem, the pendulum starts in a random position, and the goal is to swing it up so it stays upright." [6]

- First run

  – Standard Settings

  – 50000 timesteps

- Second run

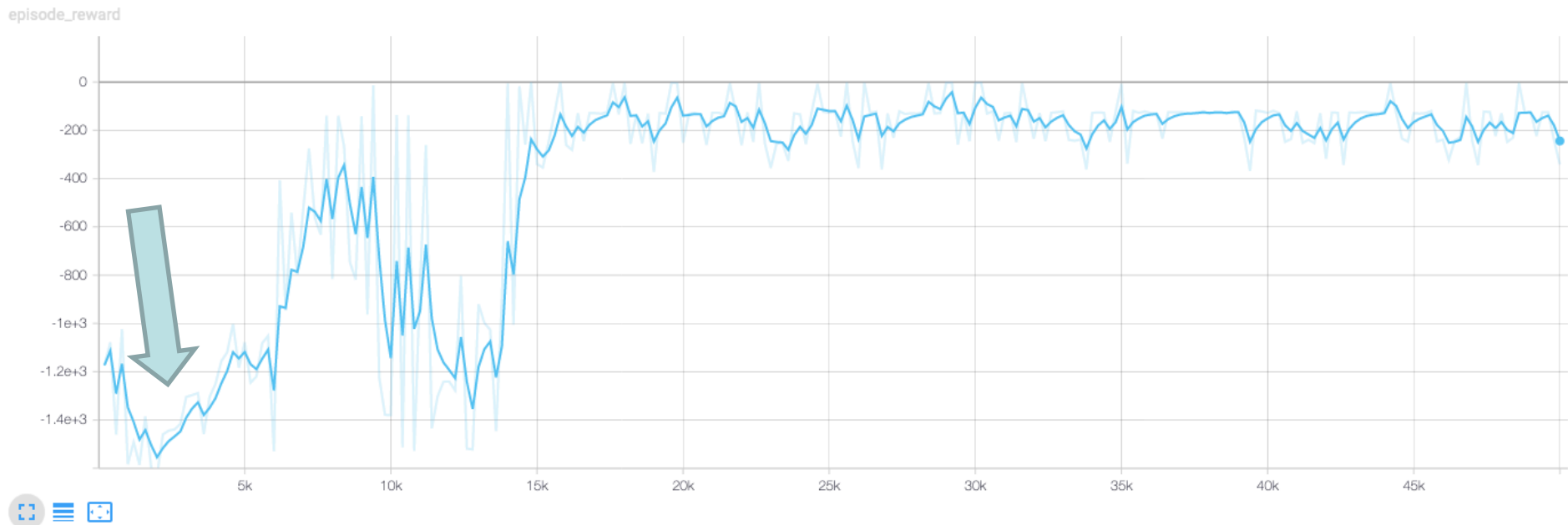 – High Noise

- Third run

  – Higher learning rate

- Fourth run

  – Very high learning rate

Experiment: Swing a Pendulum
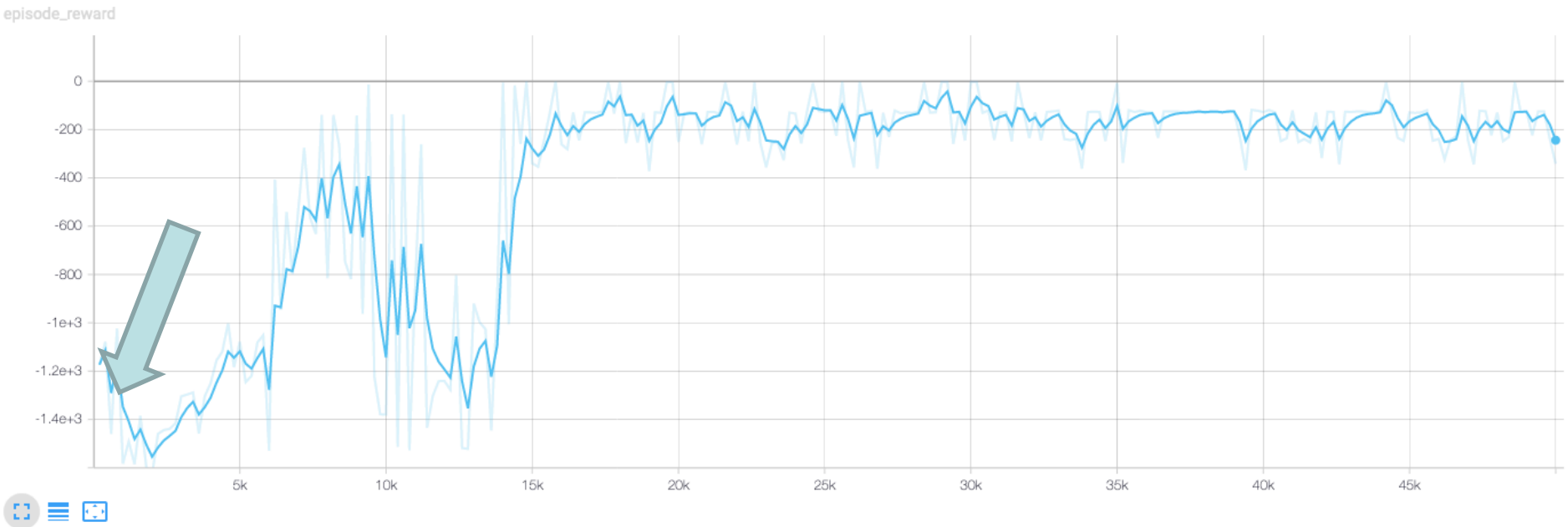
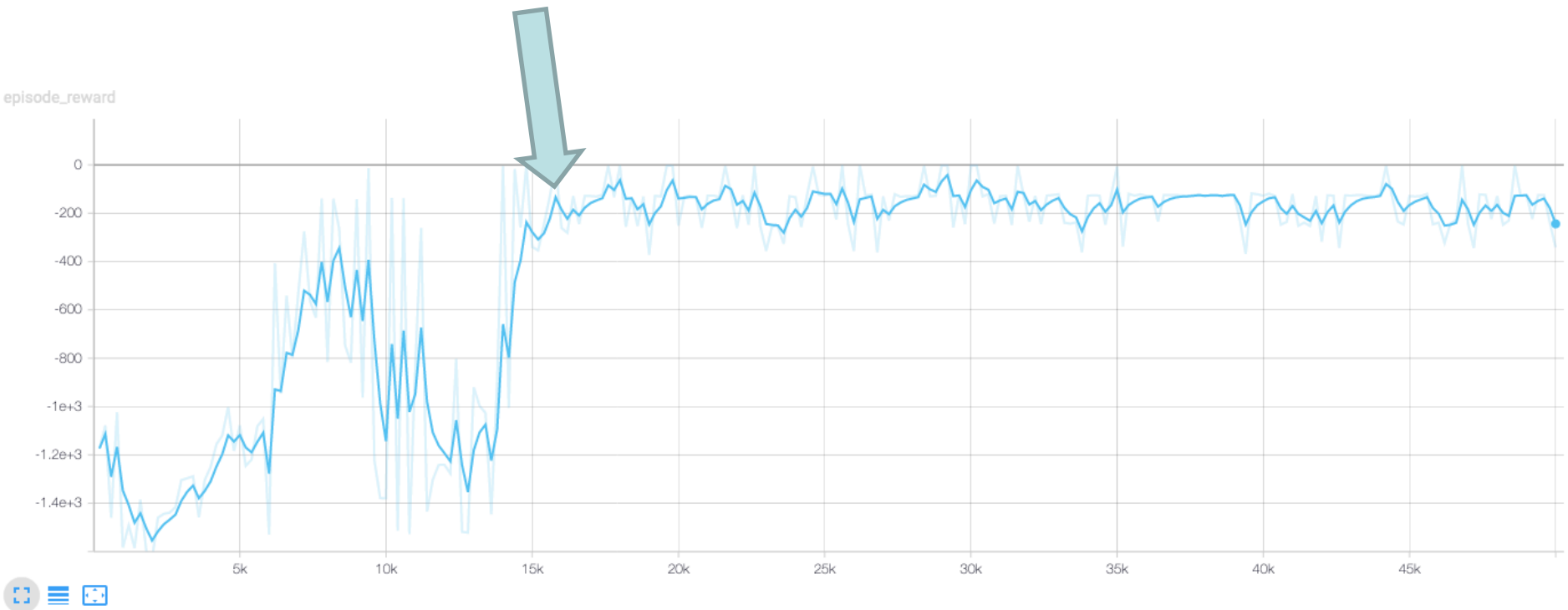# Start Video
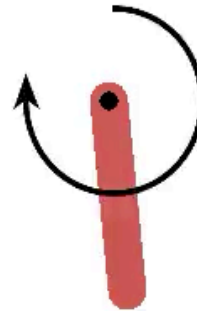
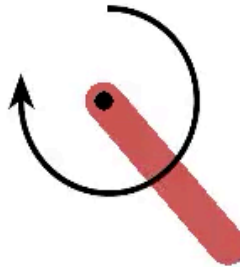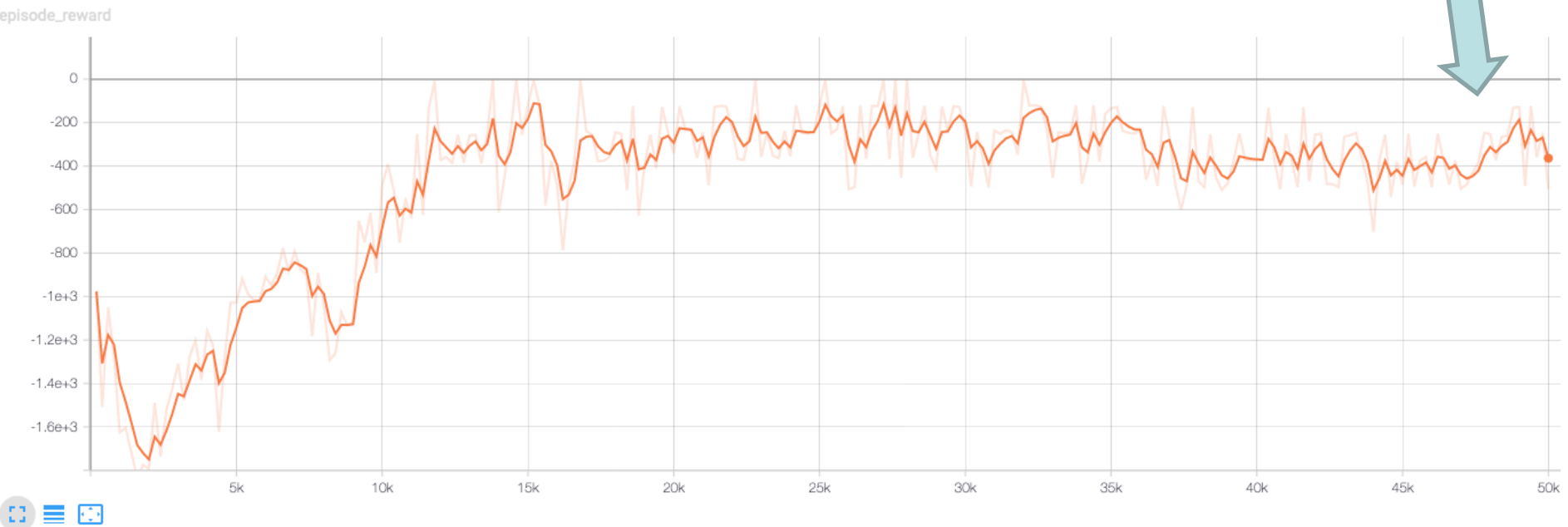# Experiment: Swing a Pendulum
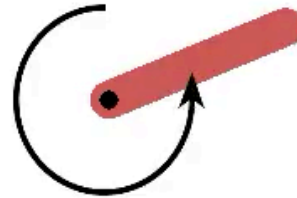
Video

Experiment: Swing a Pendulum
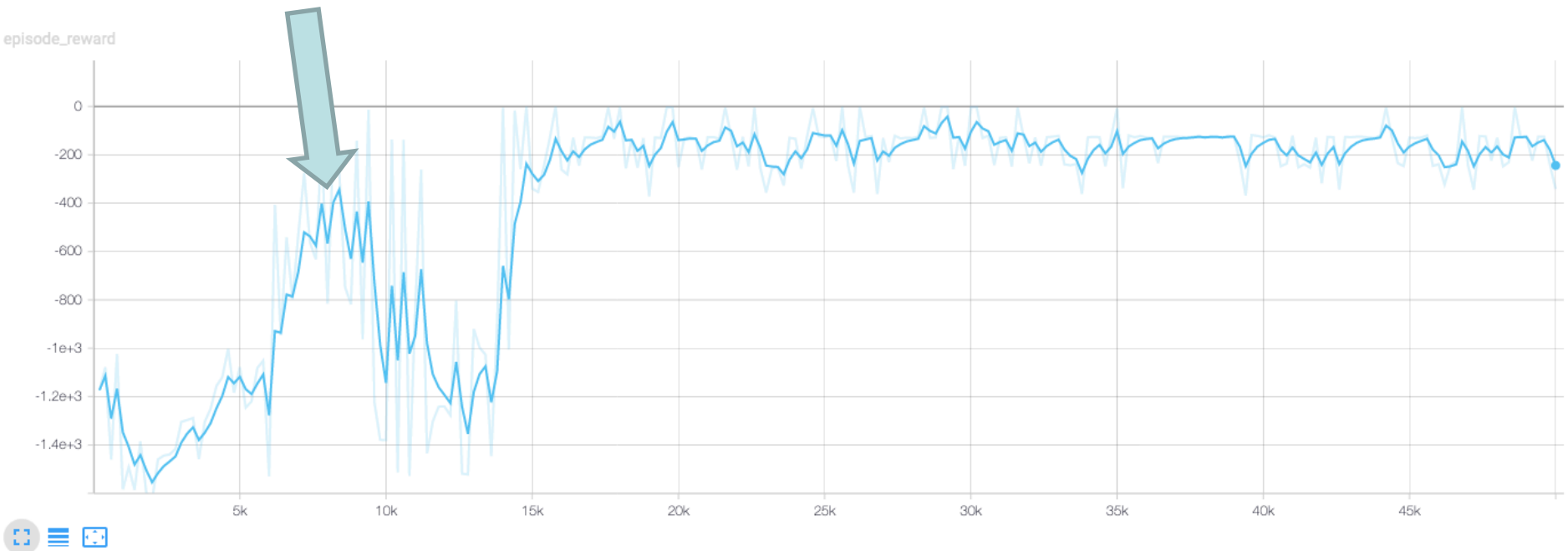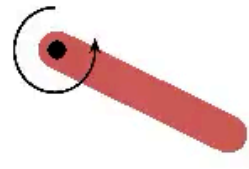
Video

Experiment: Swing a Pendulum

Video

Higher noise

High Noise
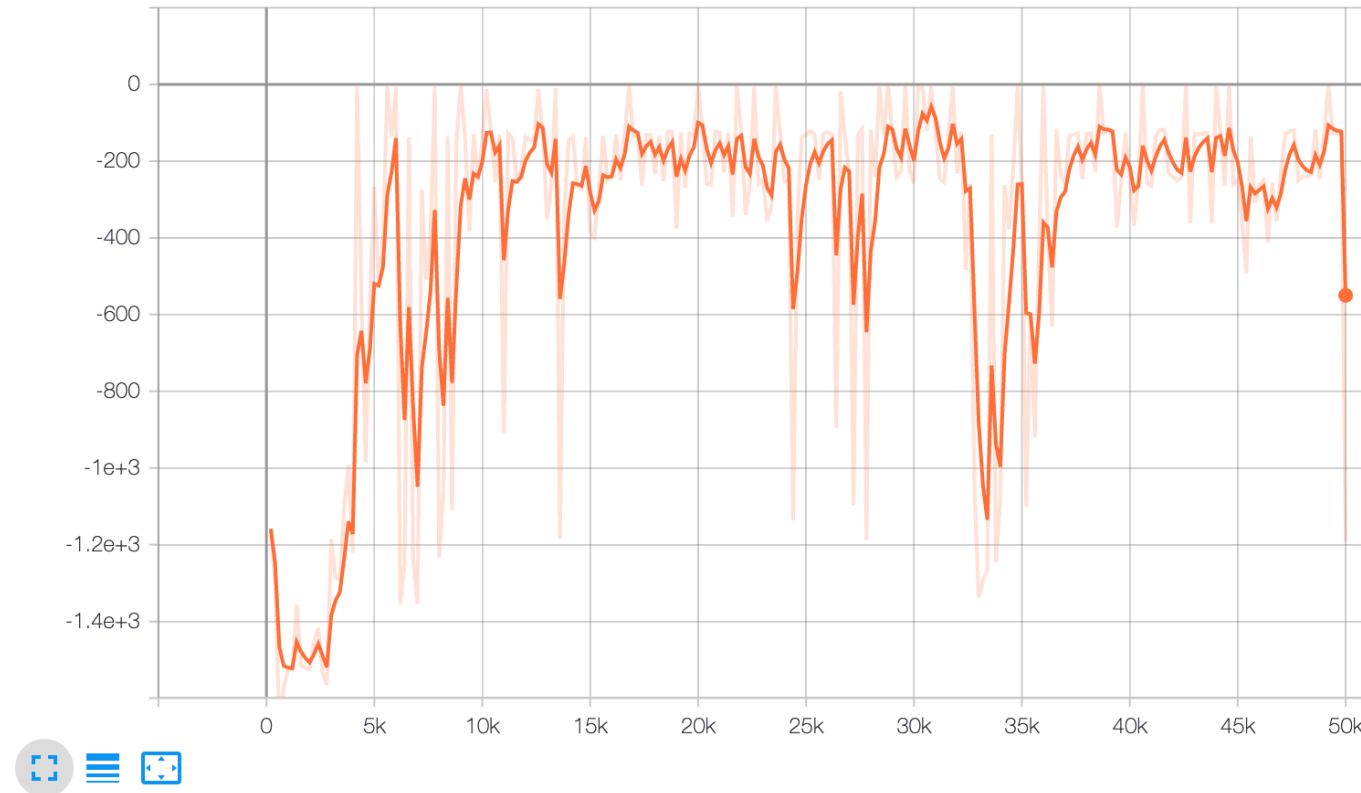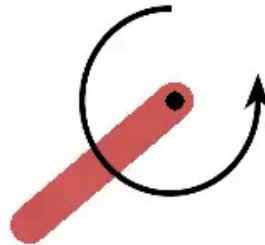
High Noise

Deep Reinforcement Learning: TD3

Higher learning rate (x100)

Higher learning rate

episode_reward

Higher learning rate

Deep Reinforcement Learning: TD3

Higher learning rate (x1000)

Deep Reinforcement Learning: TD3

High learning rate

High Noise

Deep Reinforcement Learning: TD3
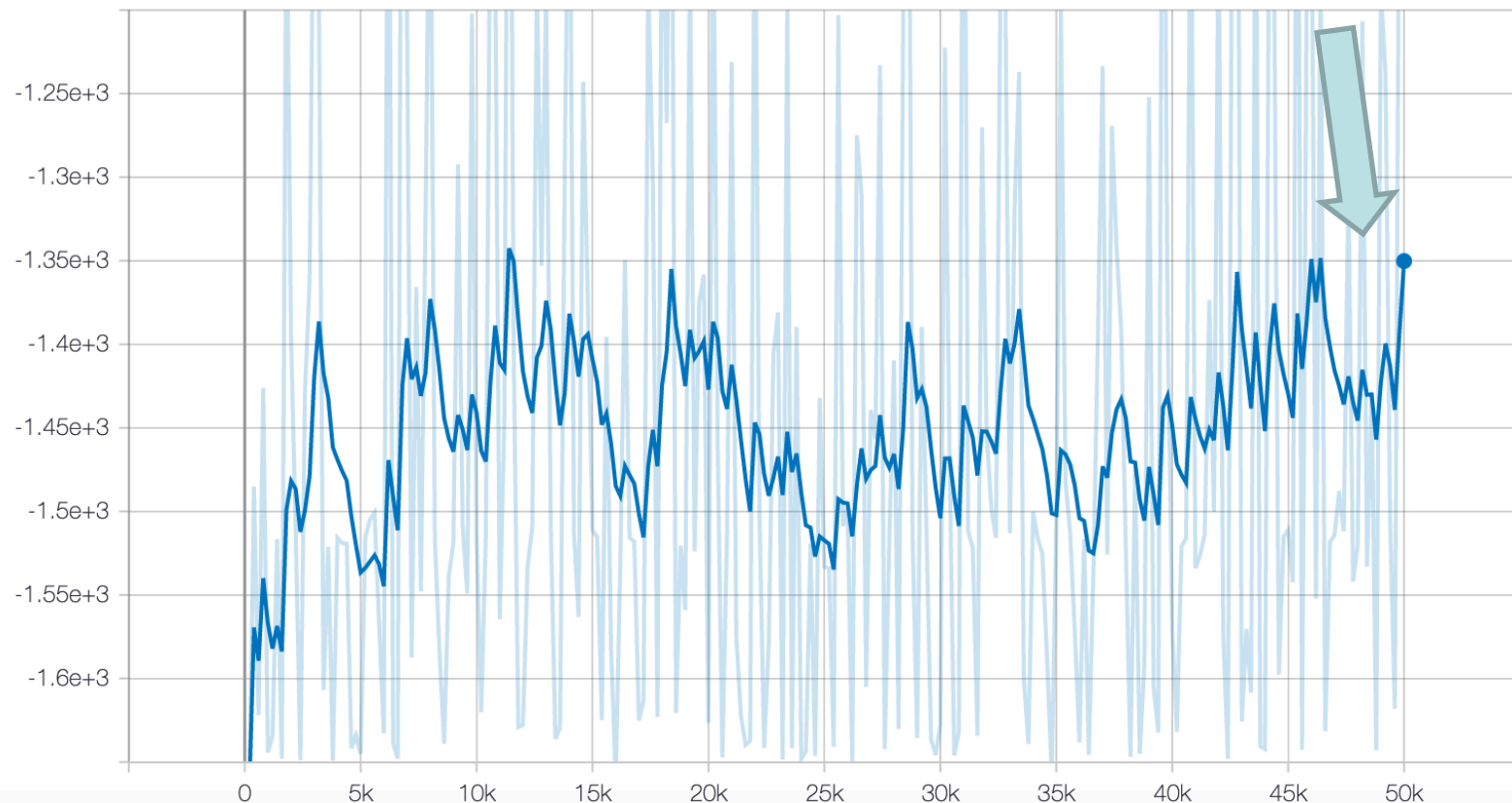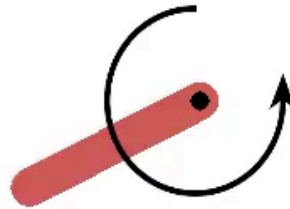
Experiment: Walking Ant

- Walking Ant

- Default Settings with a little more noise
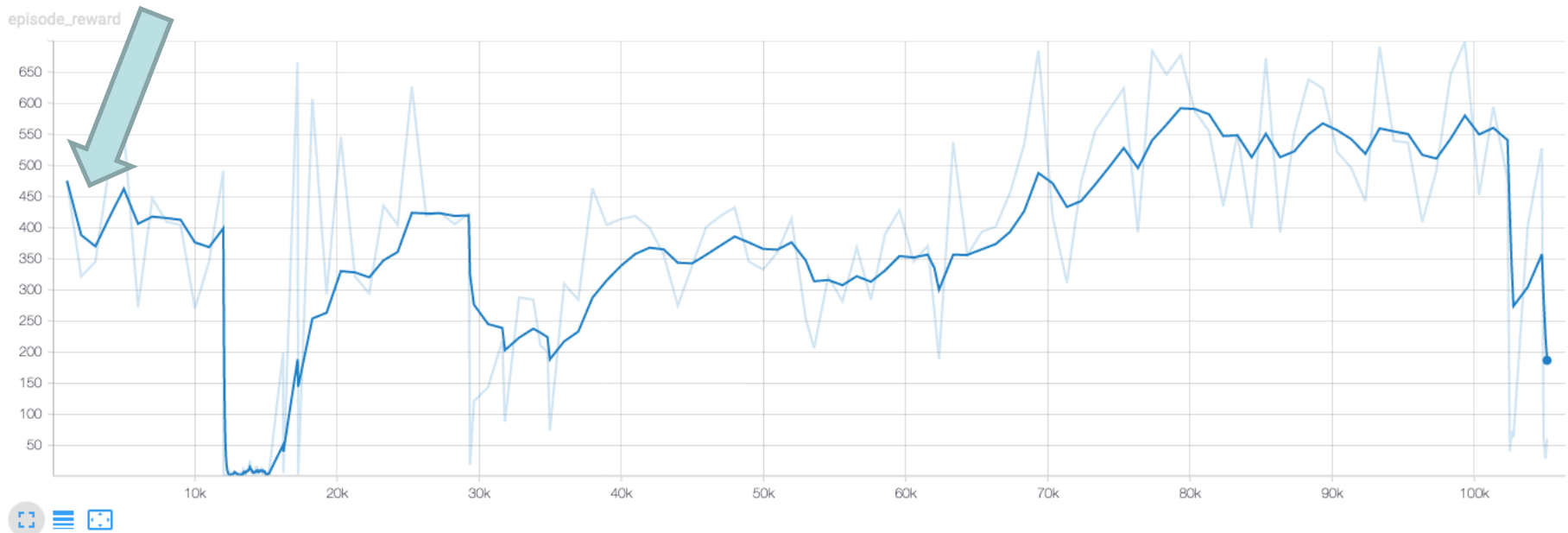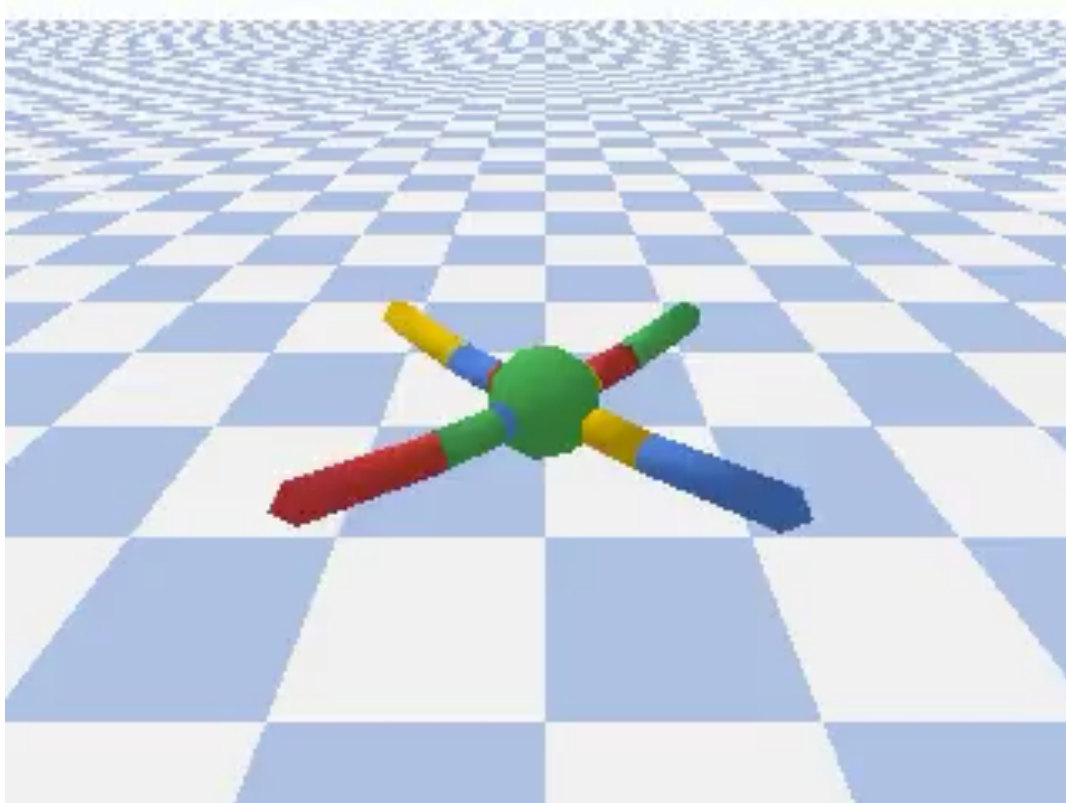
Deep Reinforcement Learning: TD3

Experiment: Walking Ant

Experiment: Walking Ant
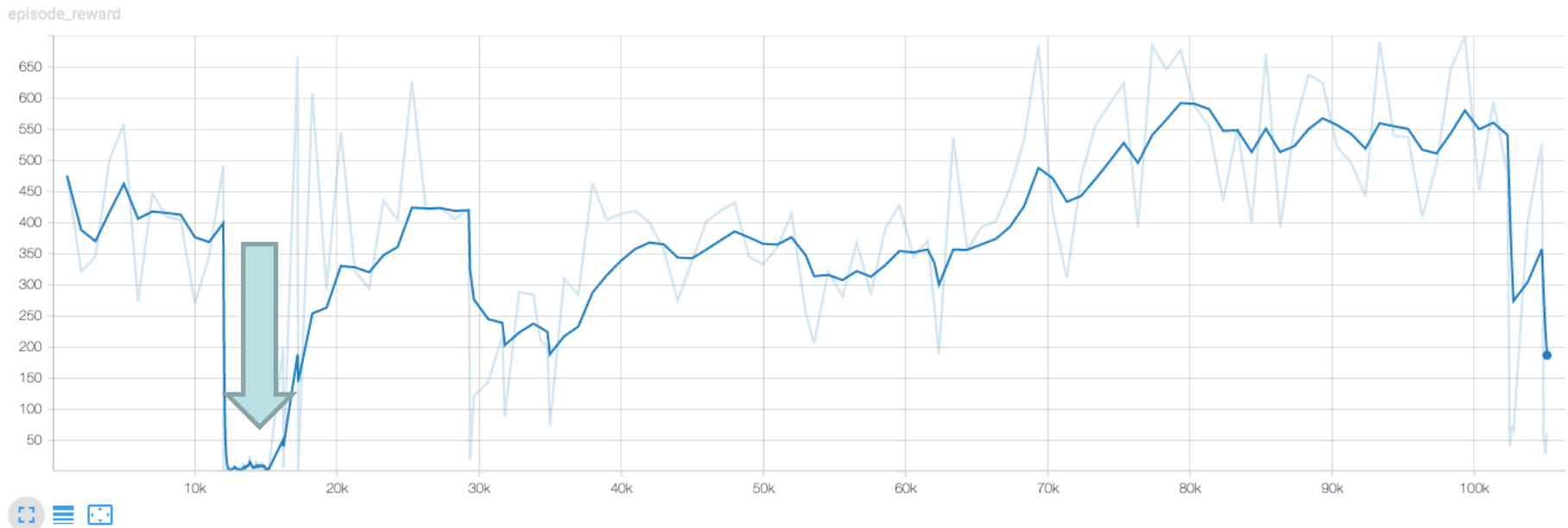
Deep Reinforcement Learning: TD3

Experiment: Walking Ant
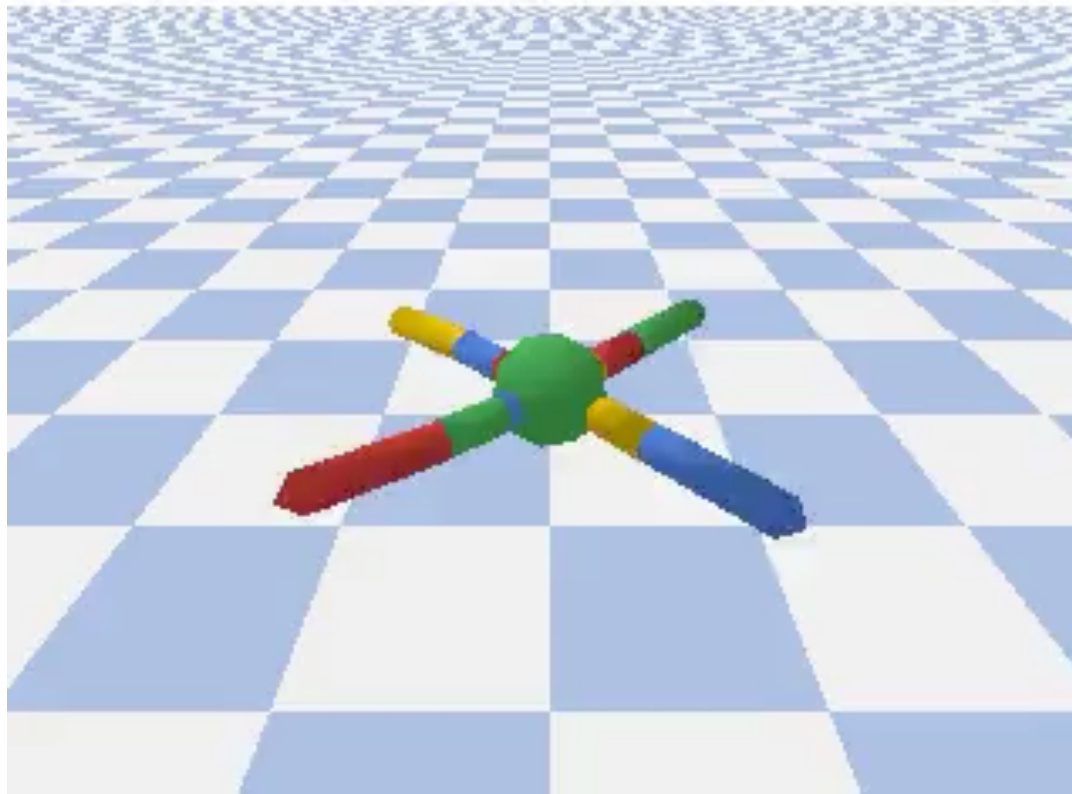
Experiment: Walking Ant
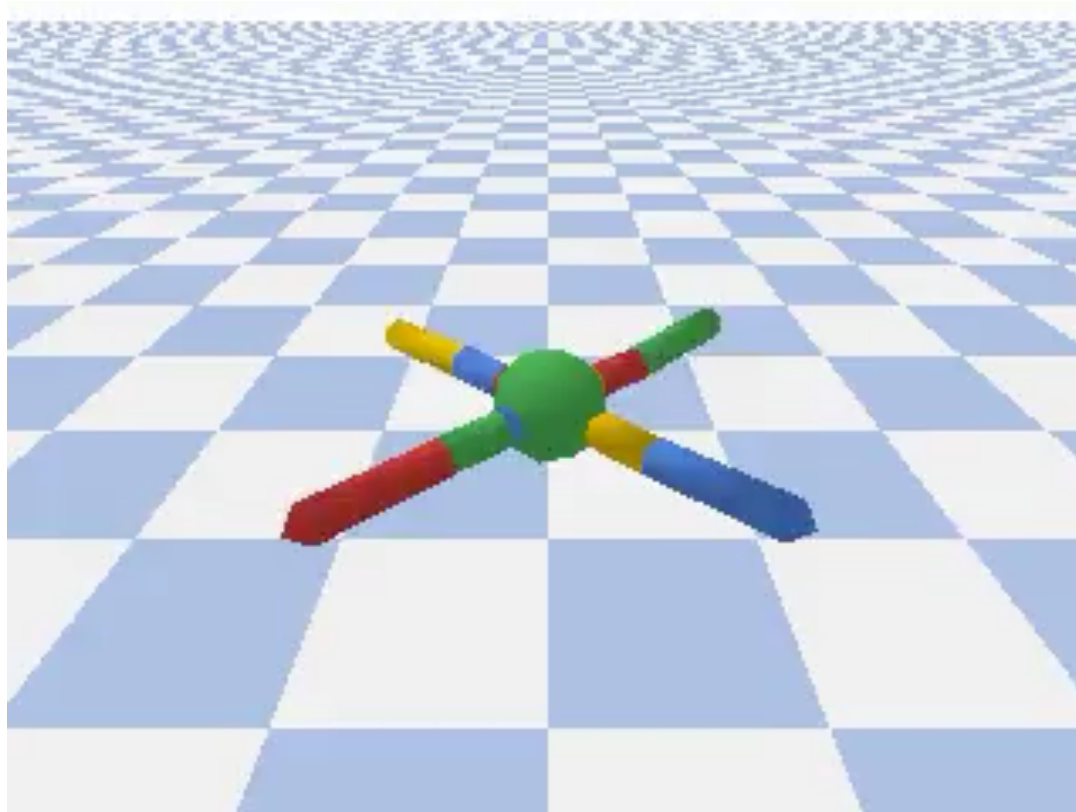
Experiment: Walking Ant
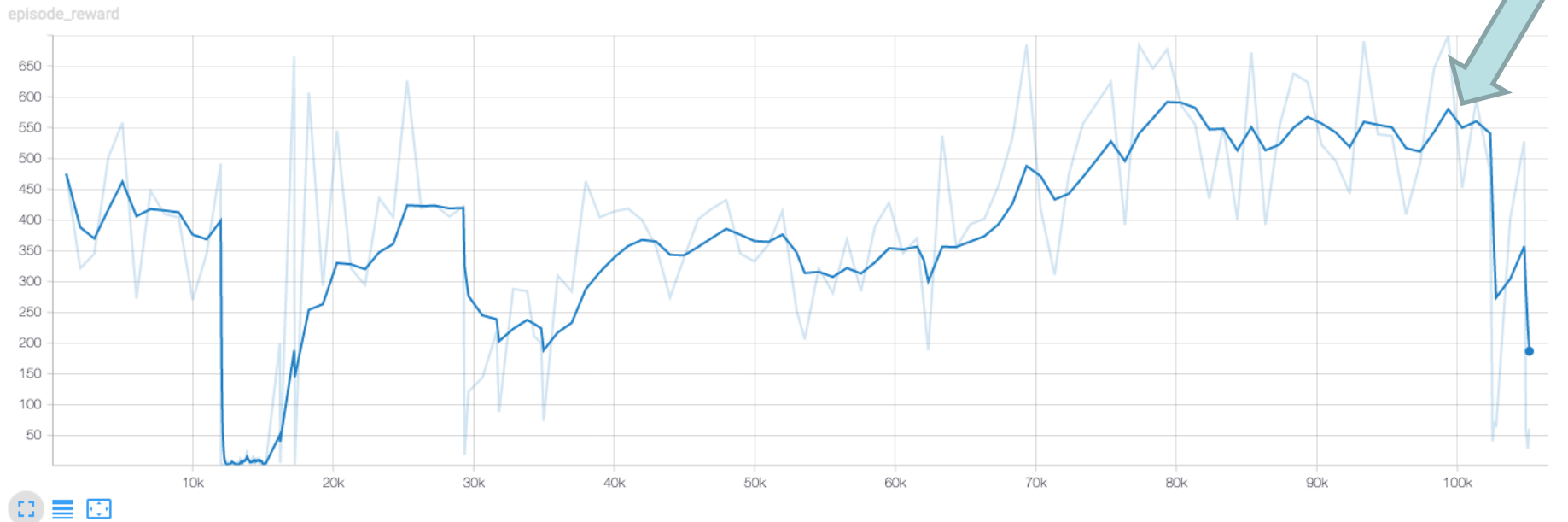
Experiment: Walking Ant

Thank you!

Deep Reinforcement Learning: TD3

Literature

- [1] https://spinningup.openai.com/en/latest/algorithms/td3.html (02.12.2019)

- [2] https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html (02.12.2019)

- [3] https://towardsdatascience.com/q-learning-54b841f3f9e4 (05.12.2019)

- [4] https://spinningup.openai.com/en/latest/algorithms/ddpg.html#the-q-learning-side-of-ddpg (18.12.2019)

- [5] https://www.mlq.ai/deep-reinforcement-learning-twin-delayed-ddpg-algorithm/ (18.12.2019)

- [6] https://gym.openai.com/envs/Pendulum-v0/ (25.12.2019)

- [7] https://towardsdatascience.com/q-learning-54b841f3f9e4

- [8] https://arxiv.org/abs/1509.02971

- Github: https://github.com/mrmarthy/DLSeminar/blob/master/PendulumTD3.ipynb