

# Continuum Robot PPO report

|                 |                     |
|-----------------|---------------------|
| Model Name      | ./model/reward_f0d5 |
| Time            | 2025-04-21 00:14    |
| Model_Type      | PPO                 |
| Timesteps       | 22000000            |
| Control Mode    | 1                   |
| Device          | cuda                |
| Network Arch    | [1024, 1024, 512]   |
| Average Error   | 0.7132445379193008  |
| batch           | 16384               |
| buffer_size     | 100                 |
| train_freq      | 1                   |
| learning_starts | 1                   |
| n_steps         | 2048                |
| n_epochs        | 10                  |
| learning_rate   | 0.0003              |
| n_env           | 48                  |
| best_value_loss | 0.18904808710018794 |

## Reward Function Source

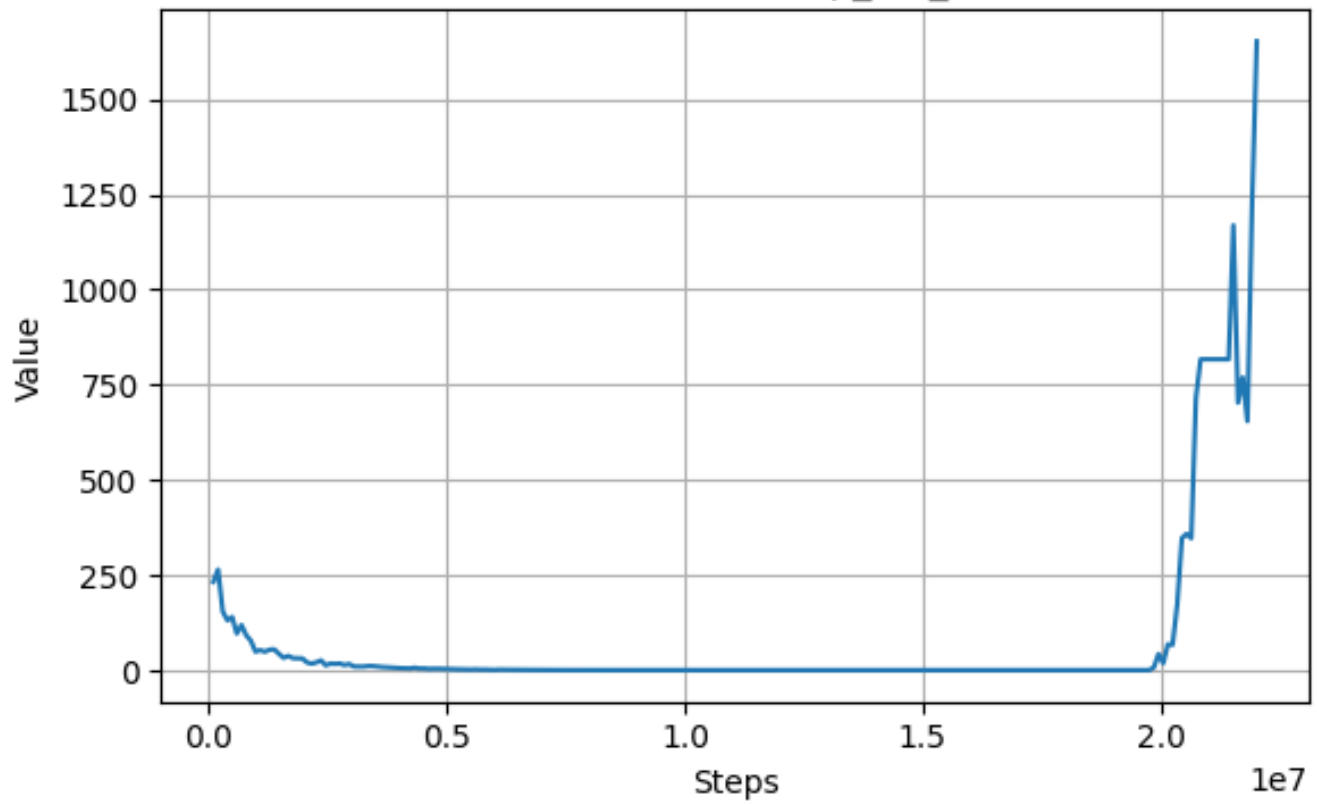
```
def my_custom_reward1(distance):  
    return -distance + np.exp(-distance**2) * 5
```

## Done Function Source

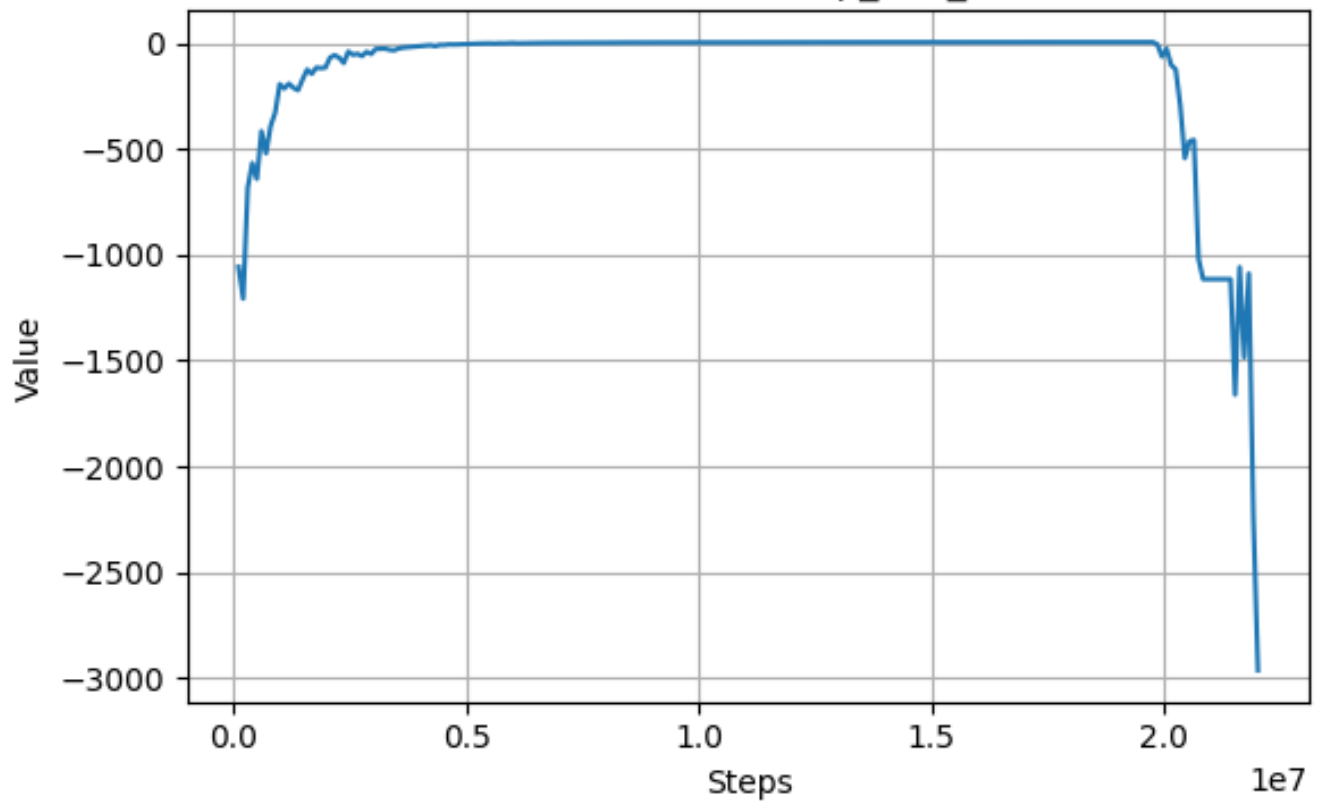
```
def my_custom_done4(reward, step):  
    return reward>-0.5
```

## Training Reward Curve

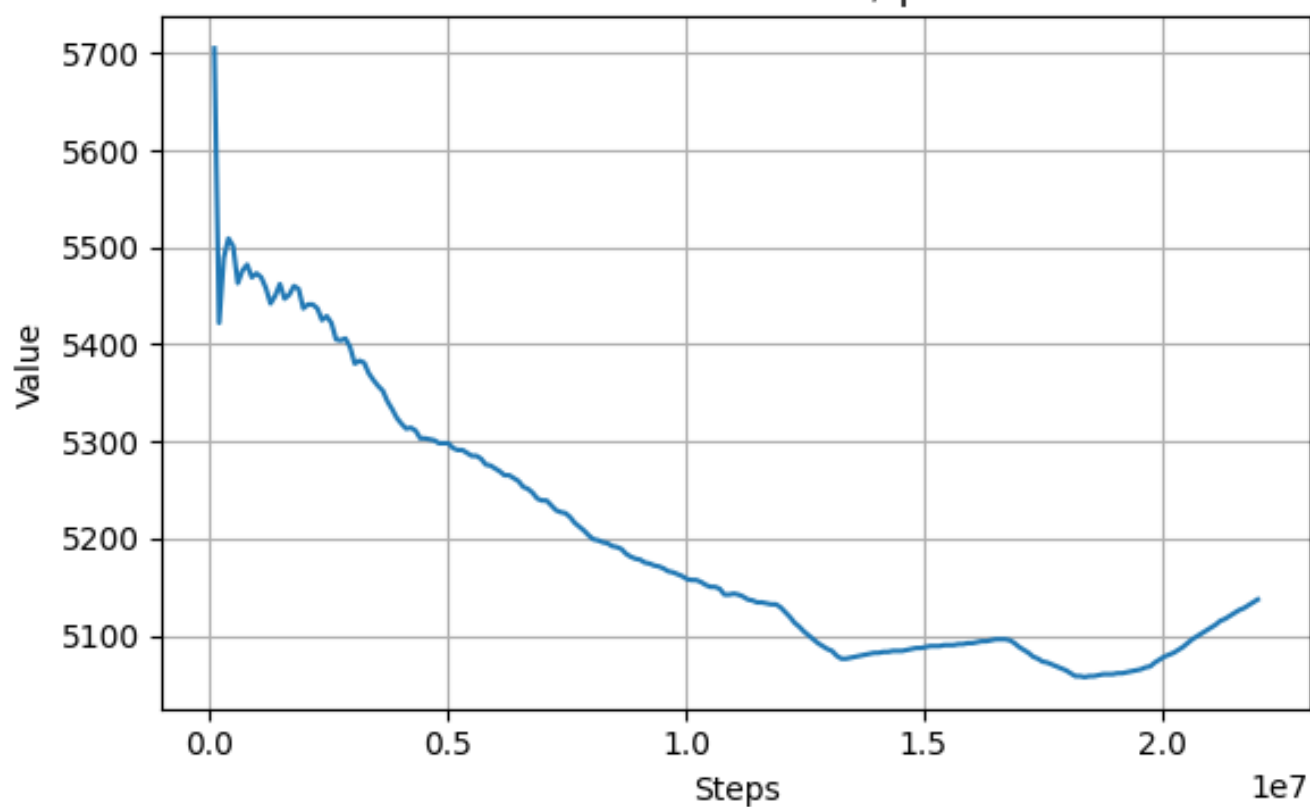
TensorBoard: rollout/ep\_len\_mean



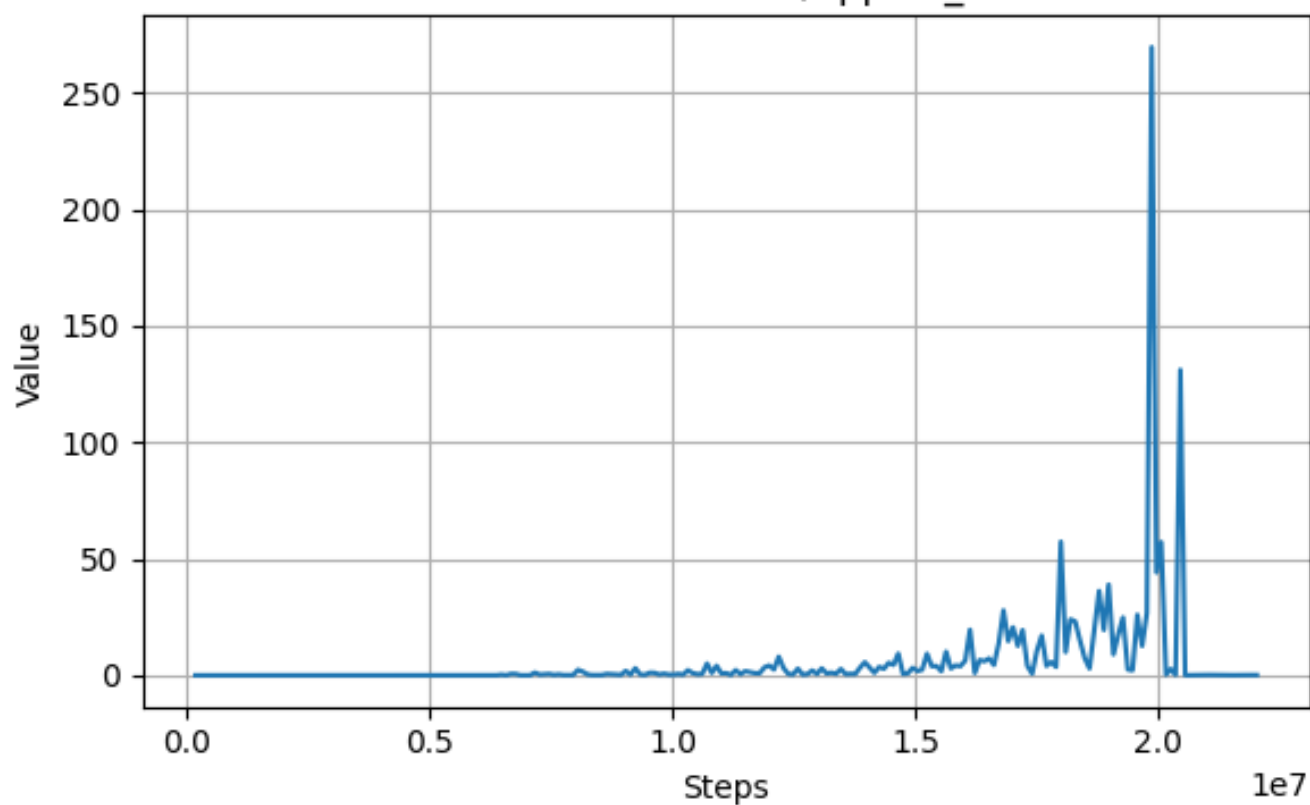
TensorBoard: rollout/ep\_rew\_mean



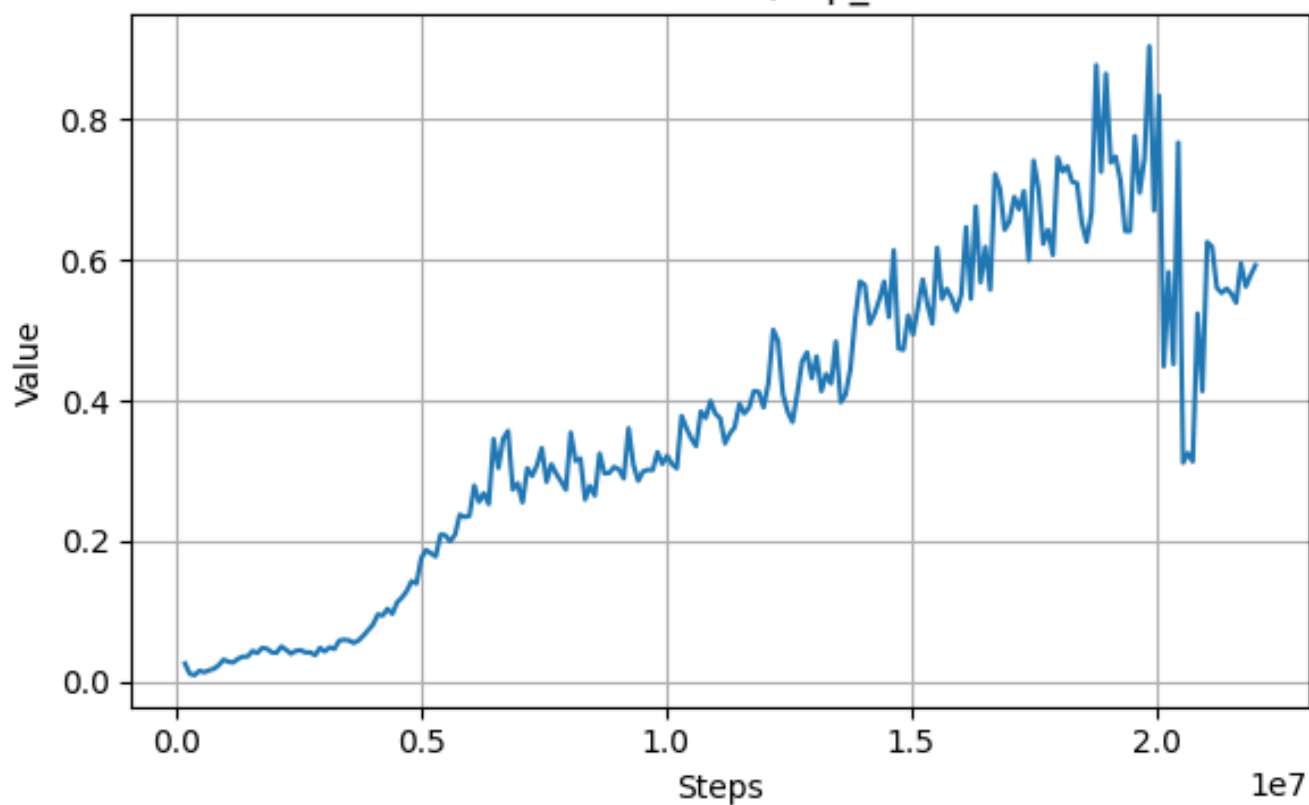
TensorBoard: time/fps



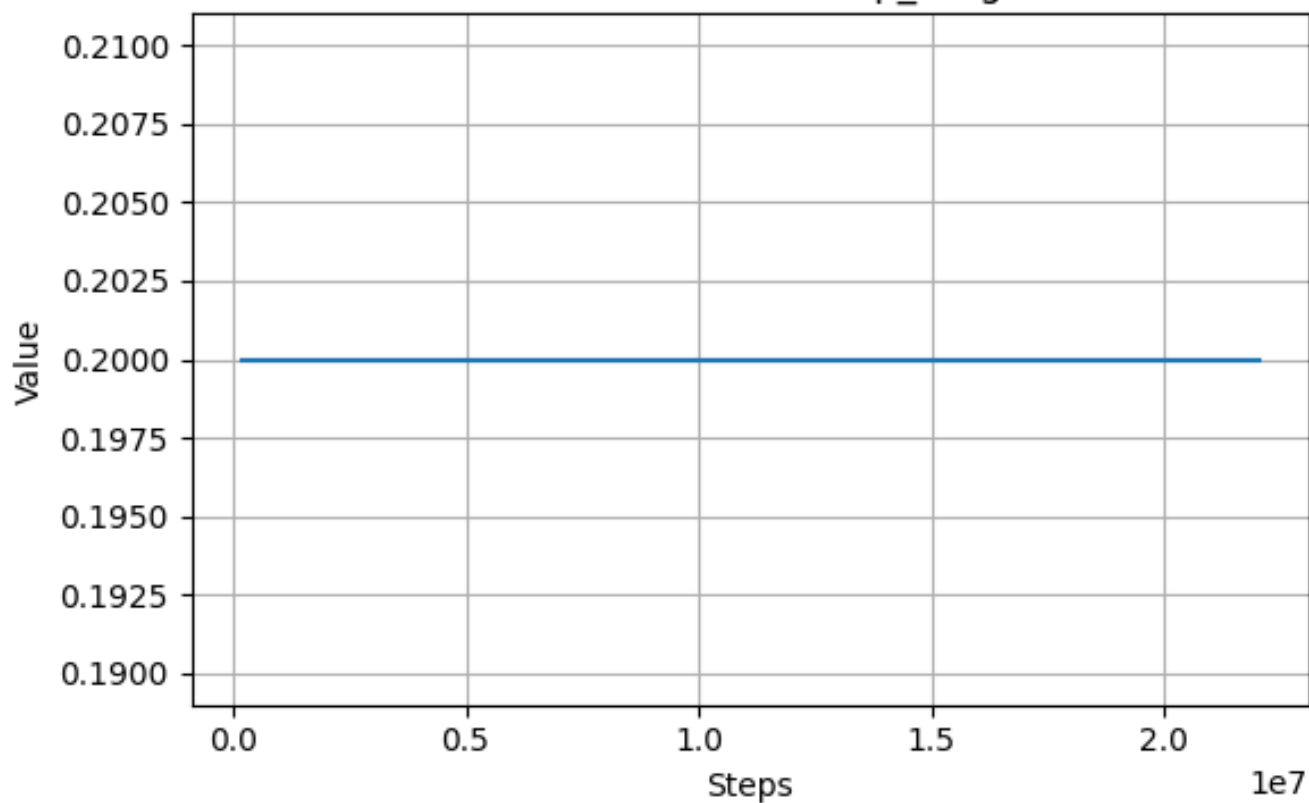
TensorBoard: train/approx\_kl



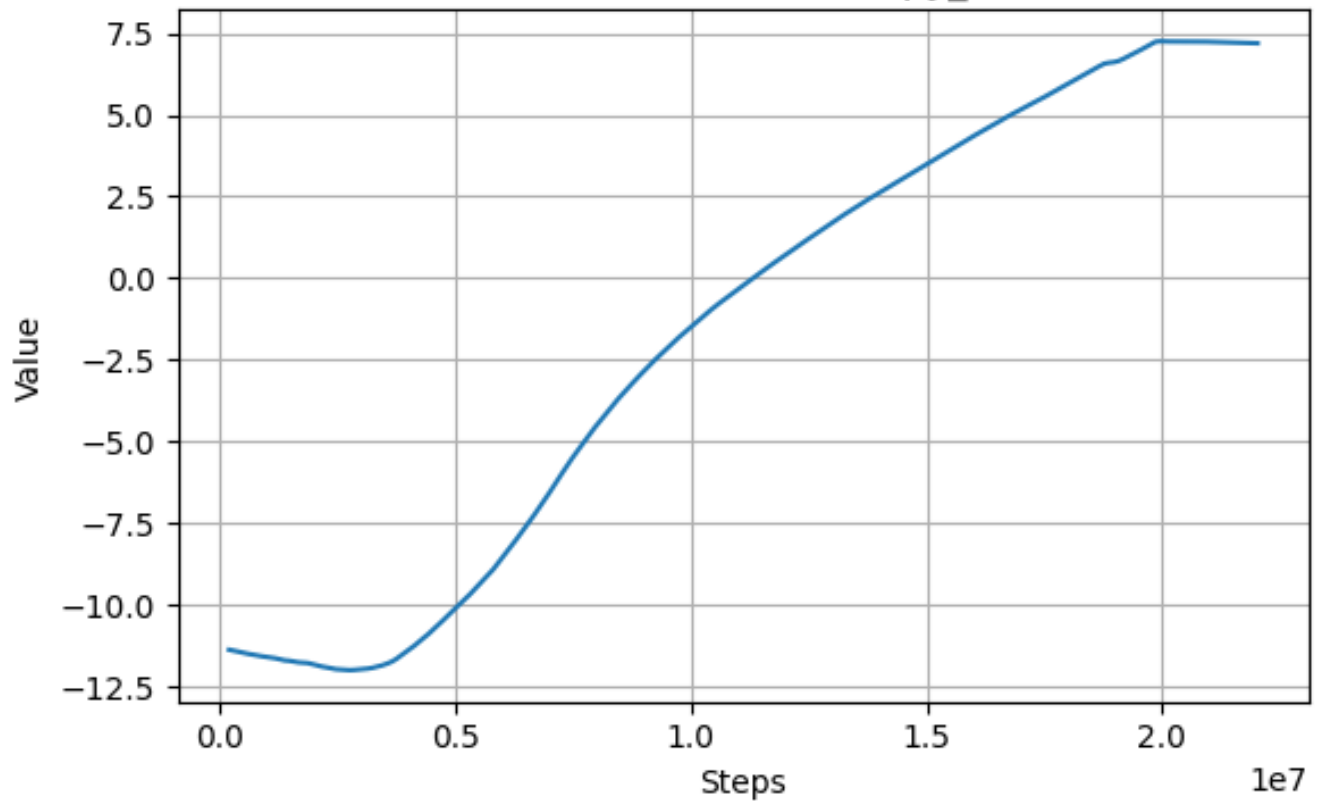
TensorBoard: train/clip\_fraction



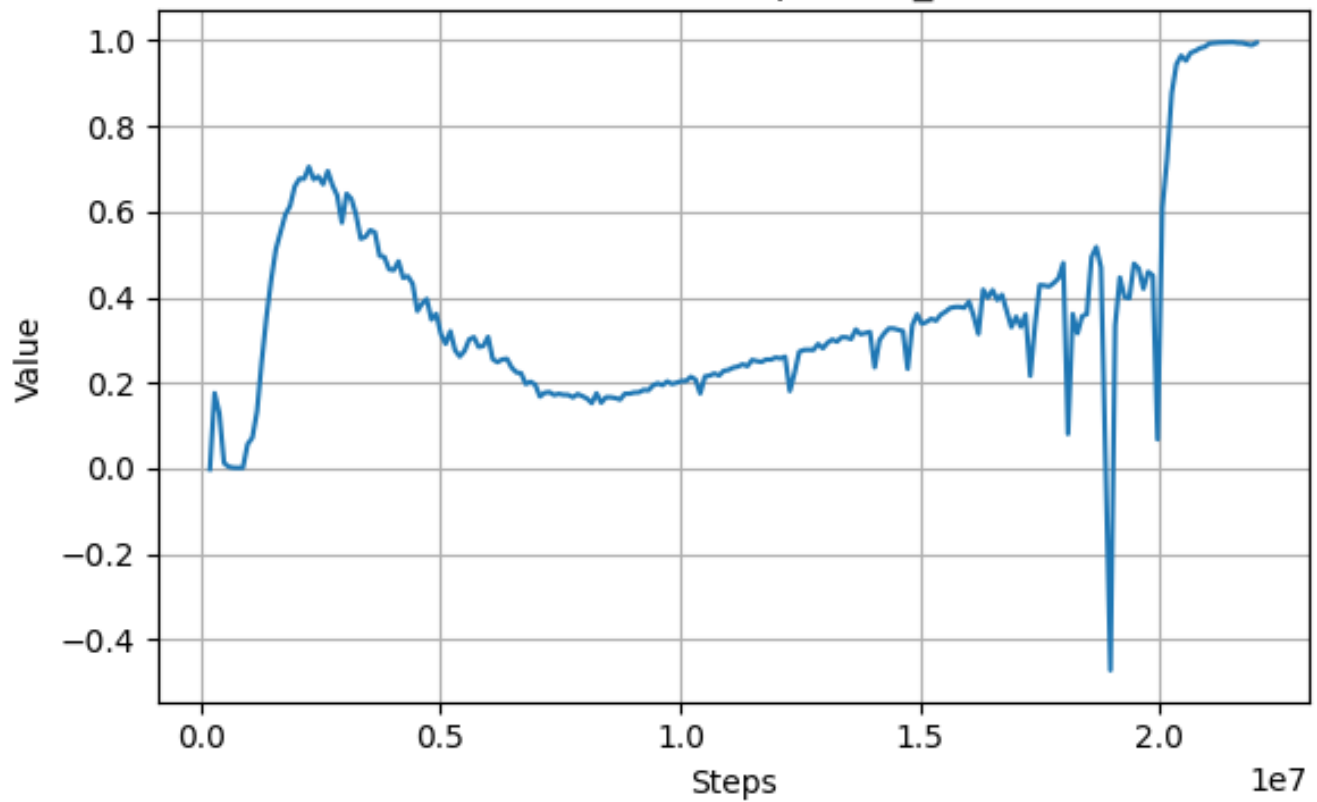
TensorBoard: train/clip\_range



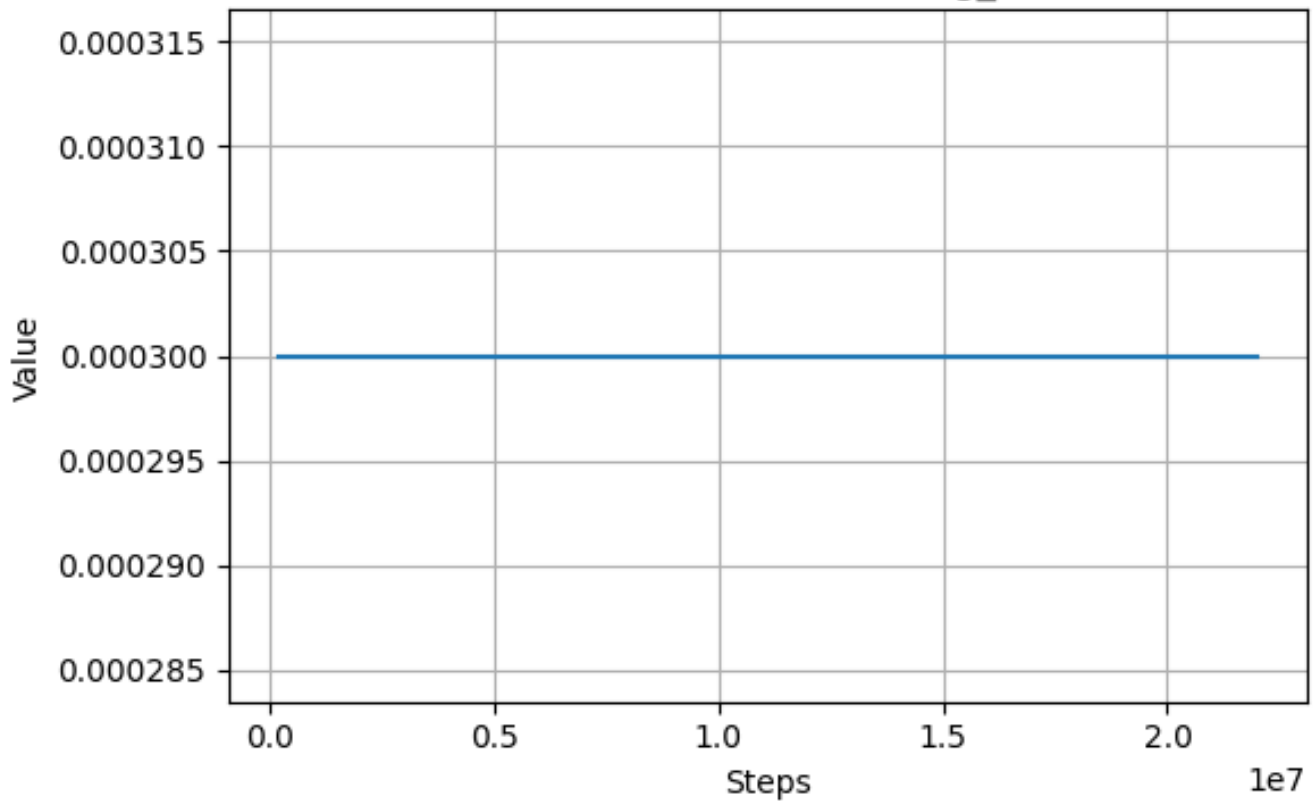
TensorBoard: train/entropy\_loss



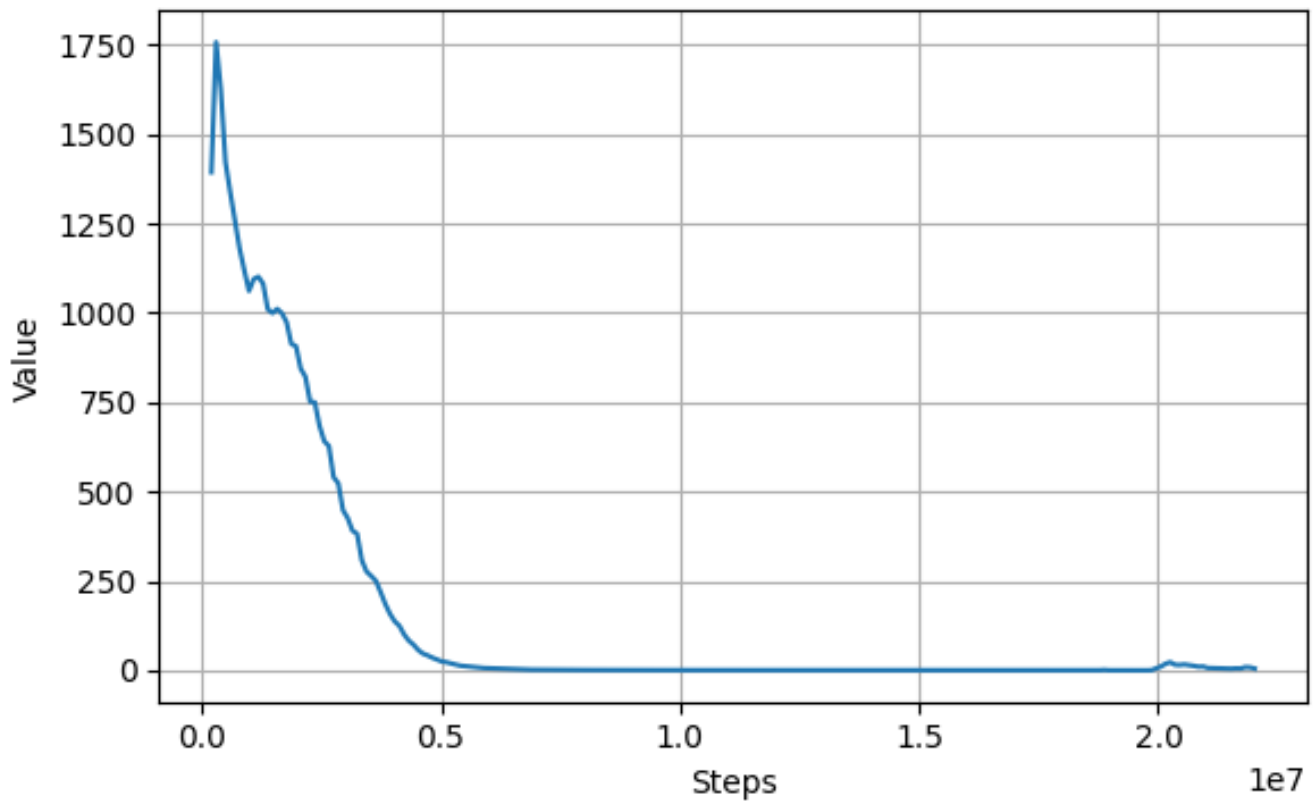
TensorBoard: train/explained\_variance



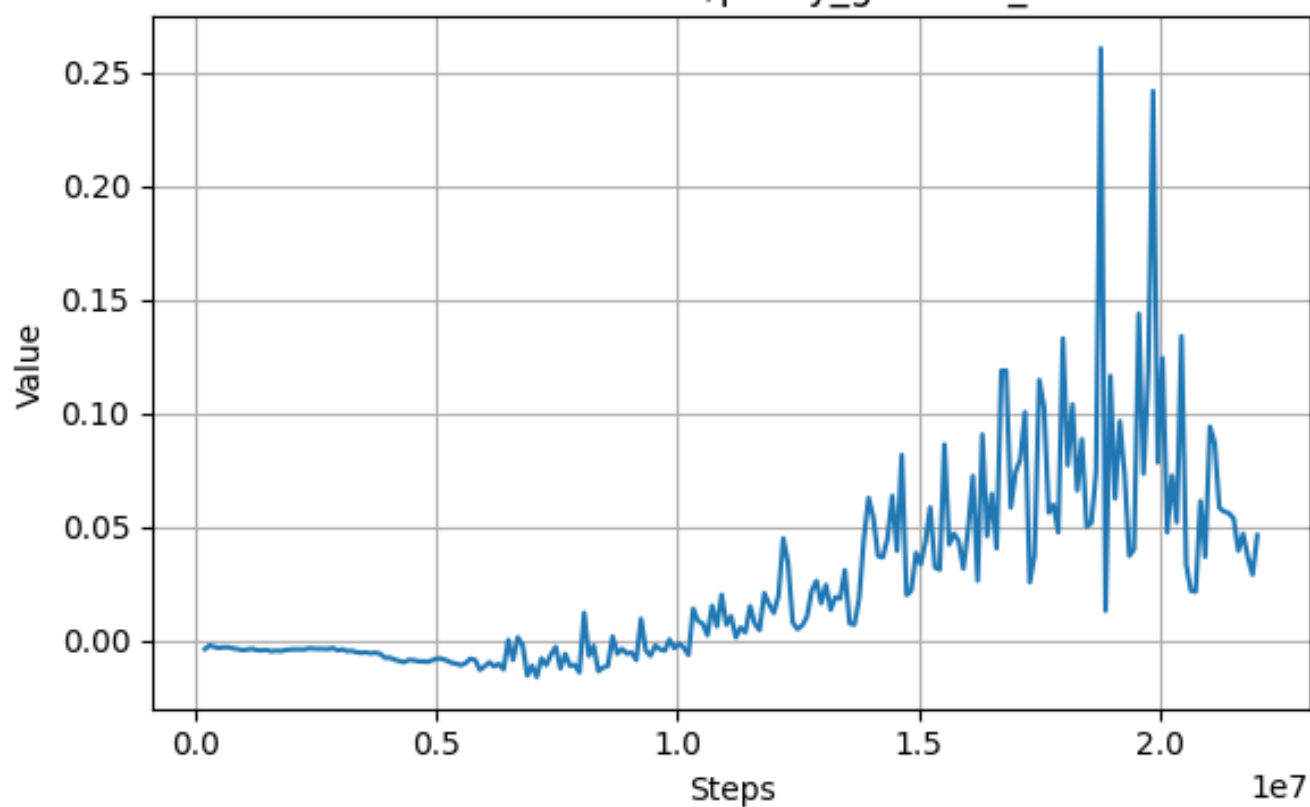
TensorBoard: train/learning\_rate



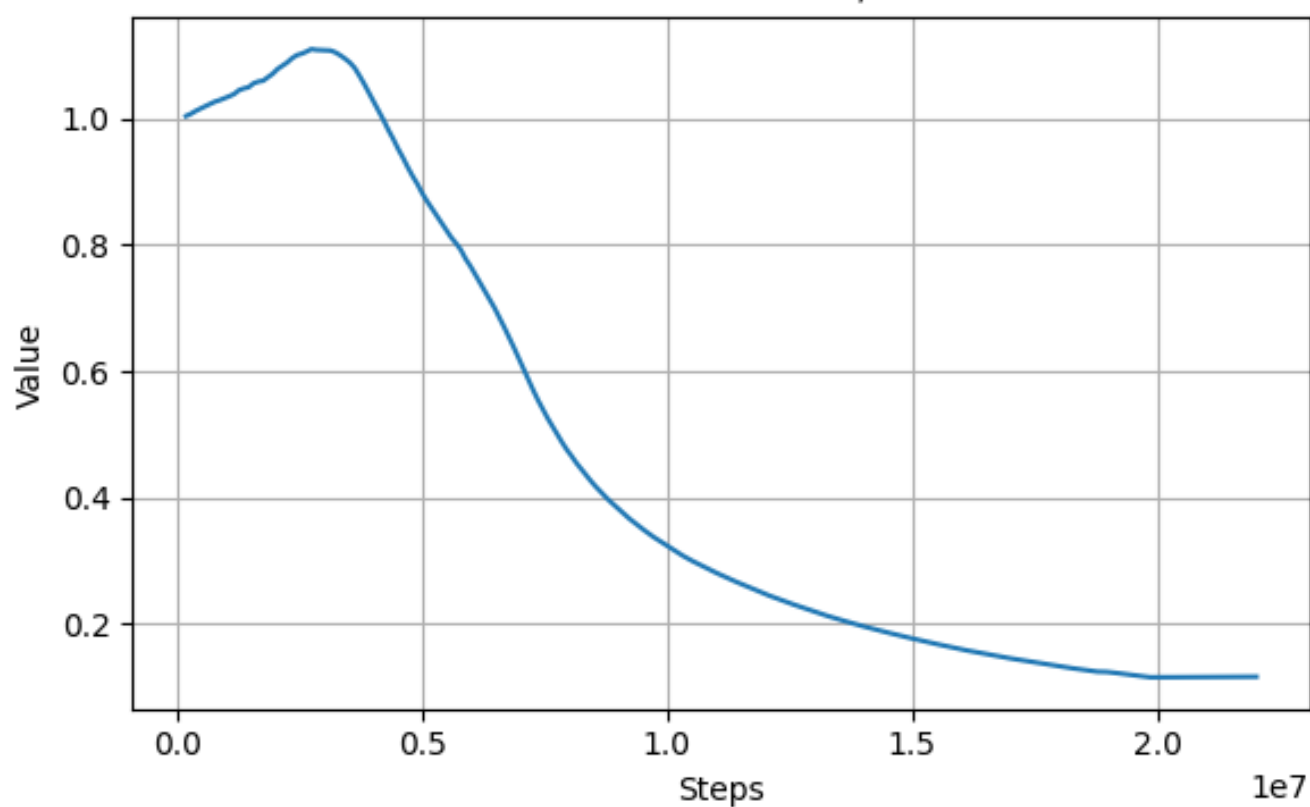
TensorBoard: train/loss



TensorBoard: train/policy\_gradient\_loss



TensorBoard: train/std



TensorBoard: train/value\_loss

