results

March 30, 2025

These experiments were run using two files.

Firstly `continuous.sh` and `train.py`.

Each instance of `train.py` ran a algorithm in a envinronment one time.

To get 10 seeds per environment with 9 envrionments and 3 algorithms there are a total of 270 runs that need to be done.

The 270 runs were completed across two job submissions 1-90 and 91-270.

[2]:
```
import os
import numpy as np
import pandas as pd
```

1 Getting the results

A slight error in the `train.py` code meant that a large chunk of the results had the labels ppo and sac switched, which resulted in the first attempt at the experiments to fail. The result is that the results are split into three different directories from three different grid jobs. Each grid output directory will ony be used to get a single model type from.

[3]:
```
results_dirs = [f'/home/thompsjame1/grid-output/
  ↪467649{n}_rl_continuous_control_baseline_experiments' for n in [0,2]]
results_dirs.append('/home/thompsjame1/grid-output/
  ↪4676867_rl_continuous_control_baseline_experiments')
results_algo = ['td3', 'ppo', 'sac']
results_dirs = [os.path.join(d, "training_evaluations") for d in results_dirs]
results_dirs
```

[3]:
```
['/home/thompsjame1/grid-
output/4676490_rl_continuous_control_baseline_experiments/training_evaluations',
 '/home/thompsjame1/grid-
output/4676492_rl_continuous_control_baseline_experiments/training_evaluations',
 '/home/thompsjame1/grid-
output/4676867_rl_continuous_control_baseline_experiments/training_evaluations']
```

[4]:
```
results_path = [
    os.path.join(results_dir, result_path)
    for results_dir, algo_type in zip(results_dirs, results_algo)
    for result_path in os.listdir(results_dir)
    if result_path.startswith(algo_type)
]

len(results_path), results_path[2:5]
```

[4]:

```
(270,
 ['/home/thompsjame1/grid-output/4676490_rl_continuous_control_baseline_experime
nts/training_evaluations/td3_HalfCheetah-v5_1000000_2.npz',
  '/home/thompsjame1/grid-output/4676490_rl_continuous_control_baseline_experime
nts/training_evaluations/td3_HalfCheetah-v5_1000000_3.npz',
  '/home/thompsjame1/grid-output/4676490_rl_continuous_control_baseline_experime
nts/training_evaluations/td3_HalfCheetah-v5_1000000_4.npz'])
```

[5]:

```
results = []
for path in results_path:
    if not path.endswith('.npz'):
        continue

    eval_name = path.split('/')[-1]

    model = eval_name.split('_')[0]
    env = eval_name.split('_')[1]
    steps = eval_name.split('_')[2]
    seed = eval_name.split('_')[3].split('.')[0]
    npz = np.load(path)

    results.append({
        'model': model,
        'env': env,
        'total_steps': steps,
        'seed': seed,
        'timesteps': npz['timesteps'],
        'rewards': npz['results'],
        'ep_lengths': npz['ep_lengths'],
    })

df = pd.DataFrame(results)
df
```

[5]:

```
    model             env total_steps seed  \
0     td3  HalfCheetah-v5     1000000    0
1     td3  HalfCheetah-v5     1000000    1
2     td3  HalfCheetah-v5     1000000    2
3     td3  HalfCheetah-v5     1000000    3
4     td3  HalfCheetah-v5     1000000    4
..    …             …         …  …
```

```
265   sac       Pusher-v5     1000000    5
266   sac       Pusher-v5     1000000    6
267   sac       Pusher-v5     1000000    7
268   sac       Pusher-v5     1000000    8
269   sac       Pusher-v5     1000000    9

                                             timesteps  \
0    [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
1    [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
2    [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
3    [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
4    [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
..                                                 …
265  [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
266  [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
267  [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
268  [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…
269  [1000, 2000, 3000, 4000, 5000, 6000, 7000, 800…

                                               rewards  \
0    [[-2.47477, -1.326504, -0.692302, -1.338596, -…
1    [[-3.266775, -2.67324, -1.939755, -2.038325, -…
2    [[0.468081, -2.293828, -1.879588, -0.693806, -…
3    [[-1.491101, -2.19, -1.539083, -1.609354, -1.1…
4    [[-2.98534, -2.277705, 0.022336, -1.132101, -2…
..                                                 …
265  [[-47.668813, -55.833692, -67.94624, -51.25912…
266  [[-54.627242, -68.915077, -55.155968, -57.6722…
267  [[-54.29353, -57.748468, -53.698268, -57.73841…
268  [[-54.836784, -52.815107, -59.760734, -58.6271…
269  [[-58.600244, -52.496547, -55.779737, -64.5120…

                                            ep_lengths
0    [[1000, 1000, 1000, 1000, 1000, 1000, 1000, 10…
1    [[1000, 1000, 1000, 1000, 1000, 1000, 1000, 10…
2    [[1000, 1000, 1000, 1000, 1000, 1000, 1000, 10…
3    [[1000, 1000, 1000, 1000, 1000, 1000, 1000, 10…
4    [[1000, 1000, 1000, 1000, 1000, 1000, 1000, 10…
..                                                 …
265  [[100, 100, 100, 100, 100, 100, 100, 100, 100,…
266  [[100, 100, 100, 100, 100, 100, 100, 100, 100,…
267  [[100, 100, 100, 100, 100, 100, 100, 100, 100,…
268  [[100, 100, 100, 100, 100, 100, 100, 100, 100,…
269  [[100, 100, 100, 100, 100, 100, 100, 100, 100,…

[270 rows x 7 columns]
```

2 Validating the results

I would like to validate two things abotu the results.

1. That the model types are actually waht they say they are.
2. That the correct number of runs and seeds have been run

2.1 Validating model type

2.2 Checking correct number of runs

3 Visualizing the results

```
[8]: df_grouped = df.explode(['timesteps', 'rewards']).groupby(['model', 'env',␣
      ↪'timesteps']).agg({
         "rewards": lambda x: np.vstack(x),
     }).reset_index()
     df_grouped
```

```
[8]:       model          env  timesteps  \
     0       ppo       Ant-v5       1000
     1       ppo       Ant-v5       2000
     2       ppo       Ant-v5       3000
     3       ppo       Ant-v5       4000
     4       ppo       Ant-v5       5000
     …      …          …         …
     27004   td3  Walker2d-v5     996000
     27005   td3  Walker2d-v5     997000
     27006   td3  Walker2d-v5     998000
     27007   td3  Walker2d-v5     999000
     27008   td3  Walker2d-v5    1000000

                                                      rewards
     0      [[994.605428, 994.789186, 994.98996, 990.99214…
     1      [[997.7473, 984.486736, 995.396178, 995.329621…
     2      [[991.231405, 989.554447, 986.973473, 987.5504…
     3      [[989.156628, 985.691269, 982.338319, 987.3692…
     4      [[978.897593, 979.452877, 989.408207, 987.4948…
     …                                                    …
     27004  [[3586.390619, 3328.577188, 3469.250313, 3408…
     27005  [[3077.539078, 3103.373637, 3082.026325, 3024…
     27006  [[3202.509432, 3237.036607, 3191.531424, 3249…
     27007  [[3358.502255, 3330.063387, 3323.810543, 3330…
     27008  [[3329.715101, 3312.109436, 3322.453729, 3316…

     [27009 rows x 4 columns]
```

[9]:

```
df_grouped["mean_rewards"] = df_grouped["rewards"].apply(lambda x: np.mean(x))
df_grouped["max_rewards"] = df_grouped["rewards"].apply(lambda x: np.max(x))
df_grouped["min_rewards"] = df_grouped["rewards"].apply(lambda x: np.min(x))
df_grouped
```

[9]:

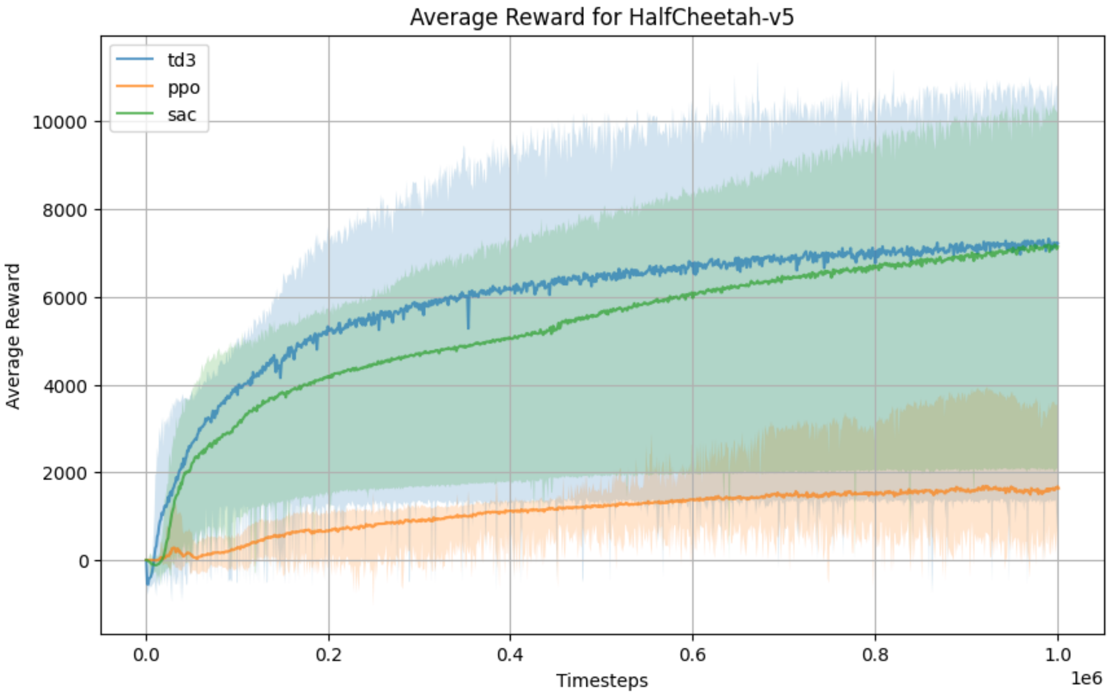
```
      model          env  timesteps  \
0       ppo       Ant-v5       1000
1       ppo       Ant-v5       2000
2       ppo       Ant-v5       3000
3       ppo       Ant-v5       4000
4       ppo       Ant-v5       5000
…     …          …        …
27004   td3  Walker2d-v5     996000
27005   td3  Walker2d-v5     997000
27006   td3  Walker2d-v5     998000
27007   td3  Walker2d-v5     999000
27008   td3  Walker2d-v5    1000000

                                                rewards  mean_rewards  \
0      [[994.605428, 994.789186, 994.98996, 990.99214…    993.814065
1      [[997.7473, 984.486736, 995.396178, 995.329621…    994.207132
2      [[991.231405, 989.554447, 986.973473, 987.5504…    987.942770
3      [[989.156628, 985.691269, 982.338319, 987.3692…    988.130807
4      [[978.897593, 979.452877, 989.408207, 987.4948…    984.505891
…                                                   …           …
27004  [[3586.390619, 3328.577188, 3469.250313, 3408…   3750.588397
27005  [[3077.539078, 3103.373637, 3082.026325, 3024…   3674.283195
27006  [[3202.509432, 3237.036607, 3191.531424, 3249…   3473.941411
27007  [[3358.502255, 3330.063387, 3323.810543, 3330…   3744.905717
27008  [[3329.715101, 3312.109436, 3322.453729, 3316…   3741.679792

       max_rewards  min_rewards
0      1004.952310   985.528747
1      1003.606960   984.486736
2      1006.239413   965.776645
3      1004.213439   958.616273
4      1014.399710   964.516934
…              …           …
27004  4787.474235   888.866982
27005  4833.346584  2126.675978
27006  4904.334854  1696.109651
27007  4885.951055  1829.380533
27008  4889.393571  2786.645381

[27009 rows x 7 columns]
```

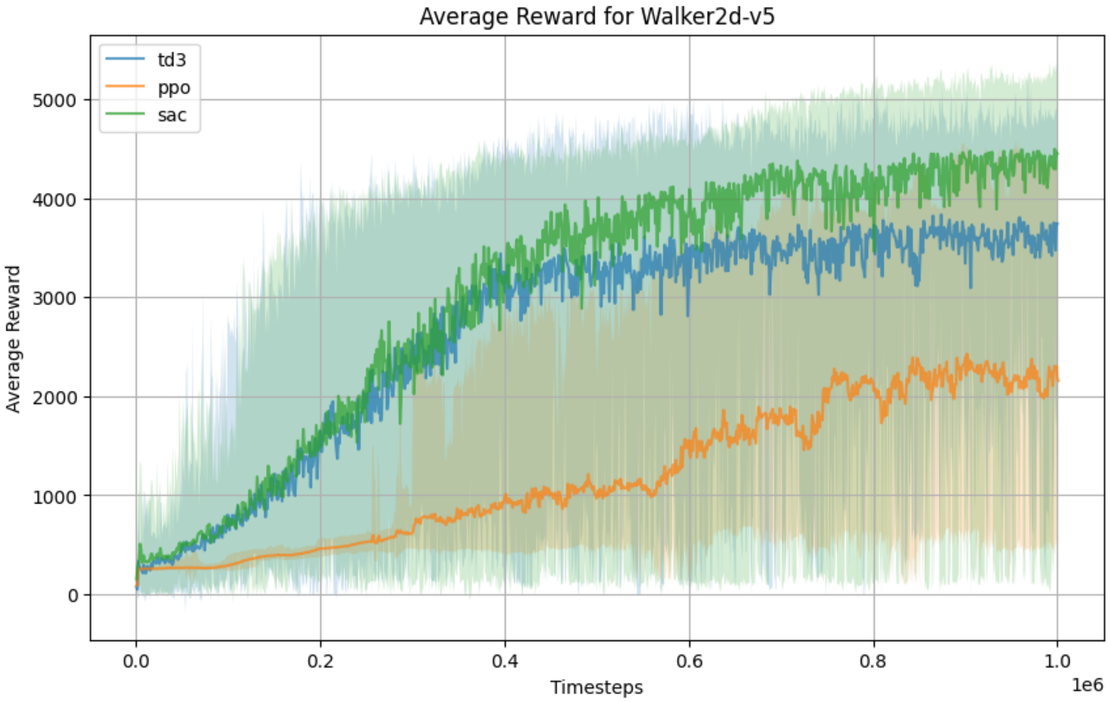
[10]:

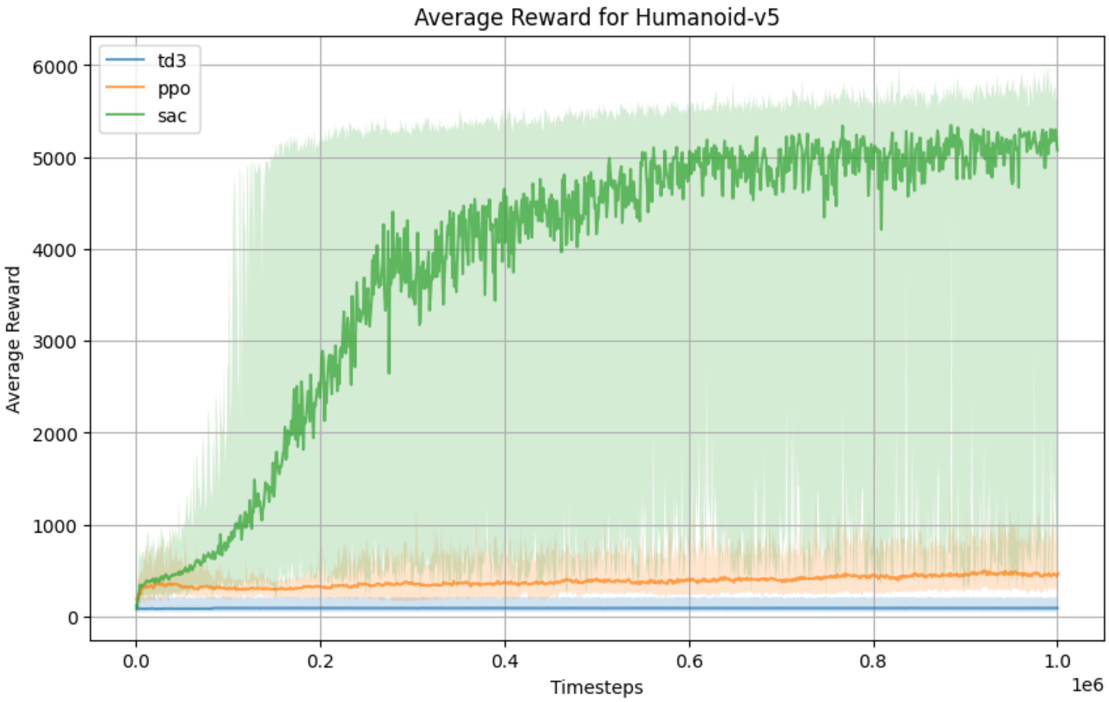
```
models = df['model'].unique()
envs = df['env'].unique()
seeds = df['seed'].unique()
```

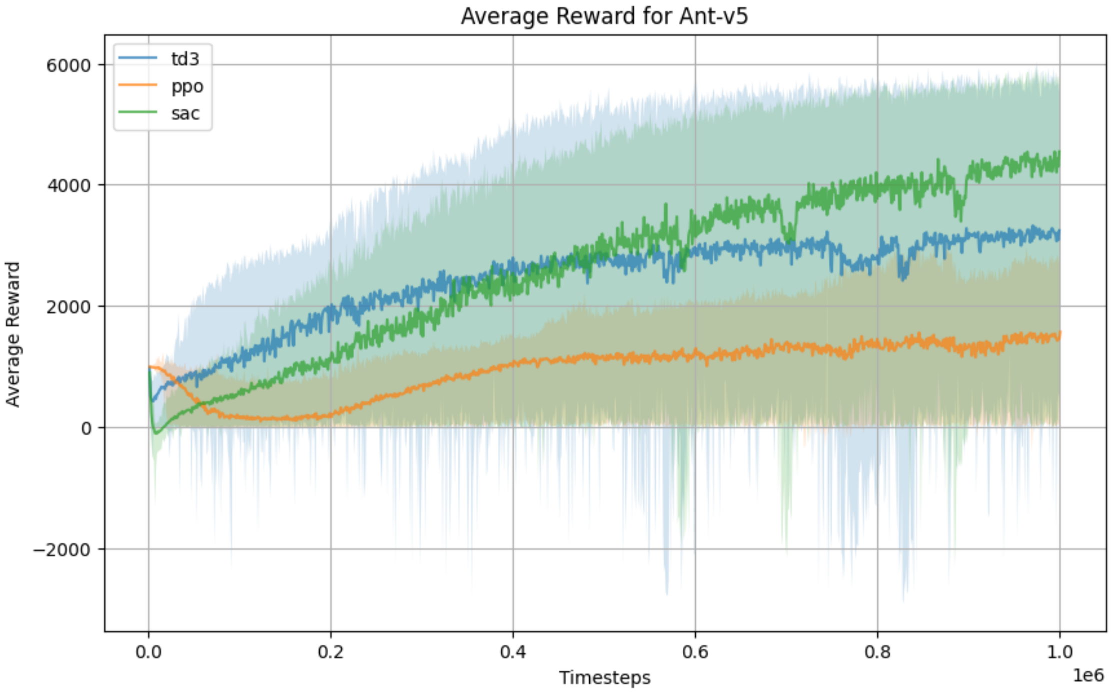
[11]:

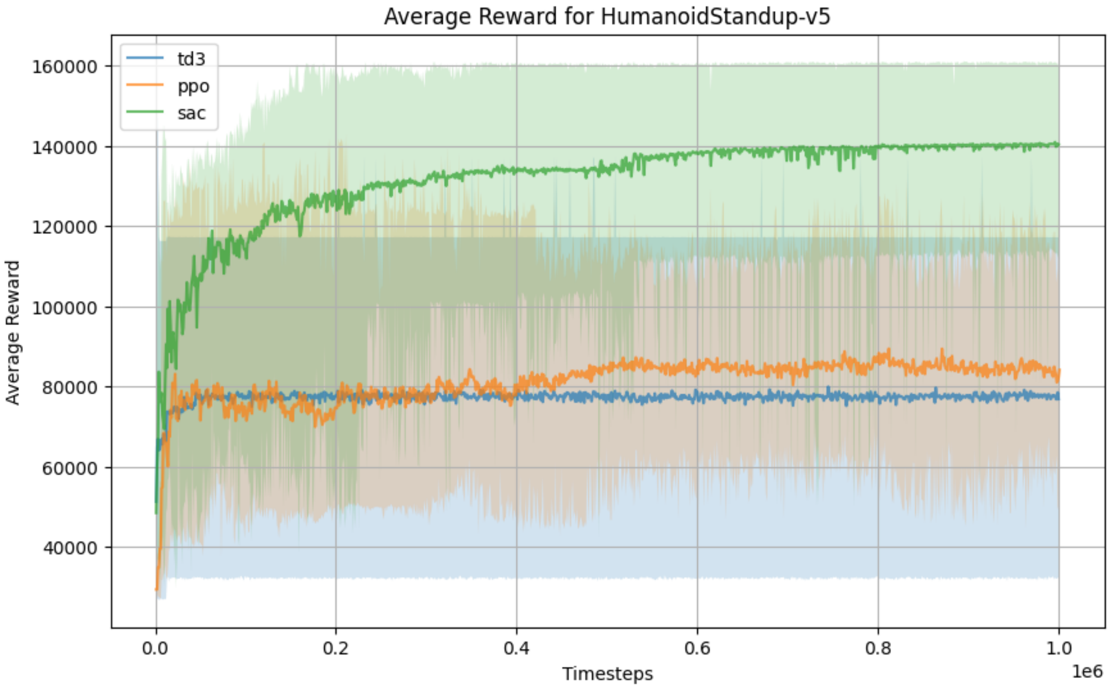
```
import matplotlib.pyplot as plt

for env in envs:
    plt.figure(figsize=(10, 6))
    for model in models:
        df_plot = df_grouped[(df_grouped['model'] == model) &␣
 ↪(df_grouped['env'] == env)]
        if df_plot.empty:
            continue

        # Plot the mean rewards as a line
        plt.plot(df_plot['timesteps'], df_plot['mean_rewards'], label=model,␣
 ↪alpha=0.7)

        # Fill between the min and max rewards to create the shaded area
        plt.fill_between(
            df_plot['timesteps'],
            df_plot['min_rewards'],
            df_plot['max_rewards'],
            alpha=0.2
        )

    plt.title(f'Average Reward for {env}')
    plt.xlabel('Timesteps')
    plt.ylabel('Average Reward')
    plt.legend()
    plt.grid(True)
    plt.show()
```

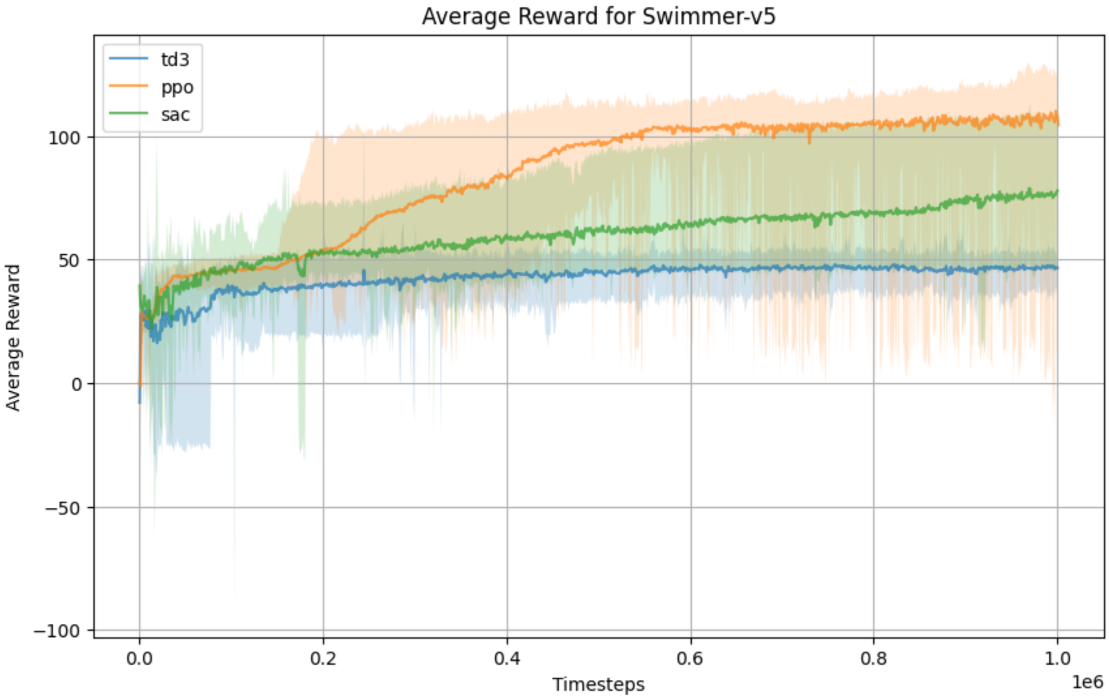

Average Reward for HalfCheetah-v5
td3
ppo
sac
Average Reward
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Timesteps

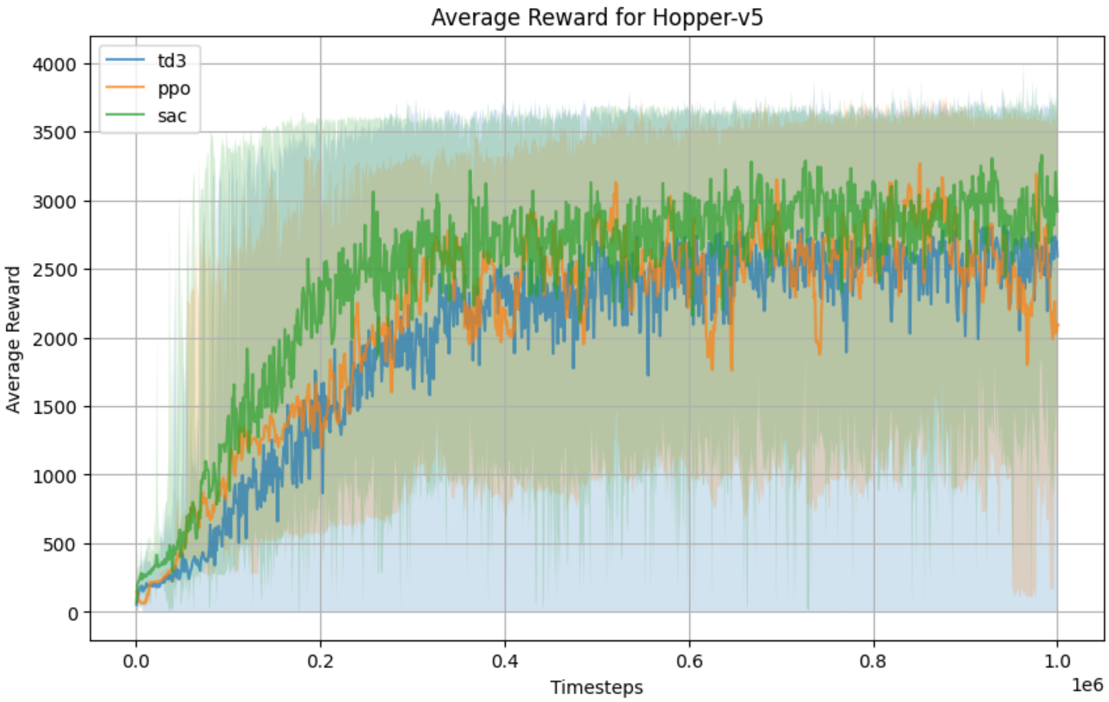

Average Reward for Walker2d-v5
td3
ppo
sac
Average Reward
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Timesteps

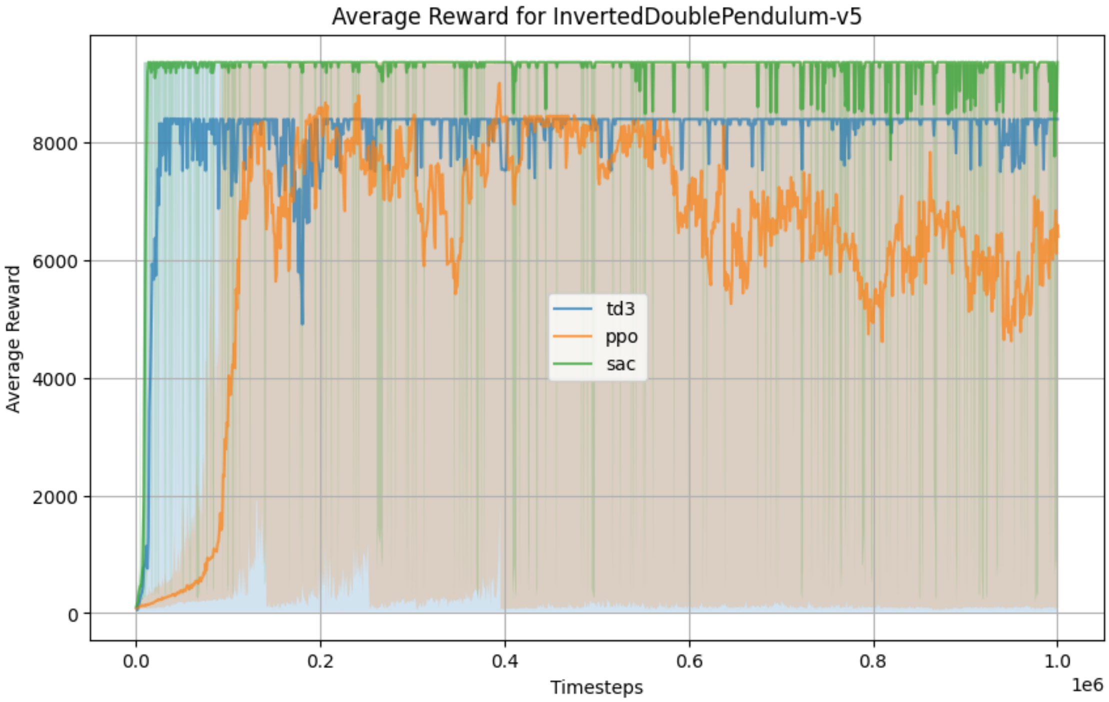

Average Reward for Humanoid-v5
td3
ppo
sac
Average Reward
0
1000
2000
3000
4000
5000
6000
Timesteps
0.0
0.2
0.4
0.6
0.8
1.0
1e6

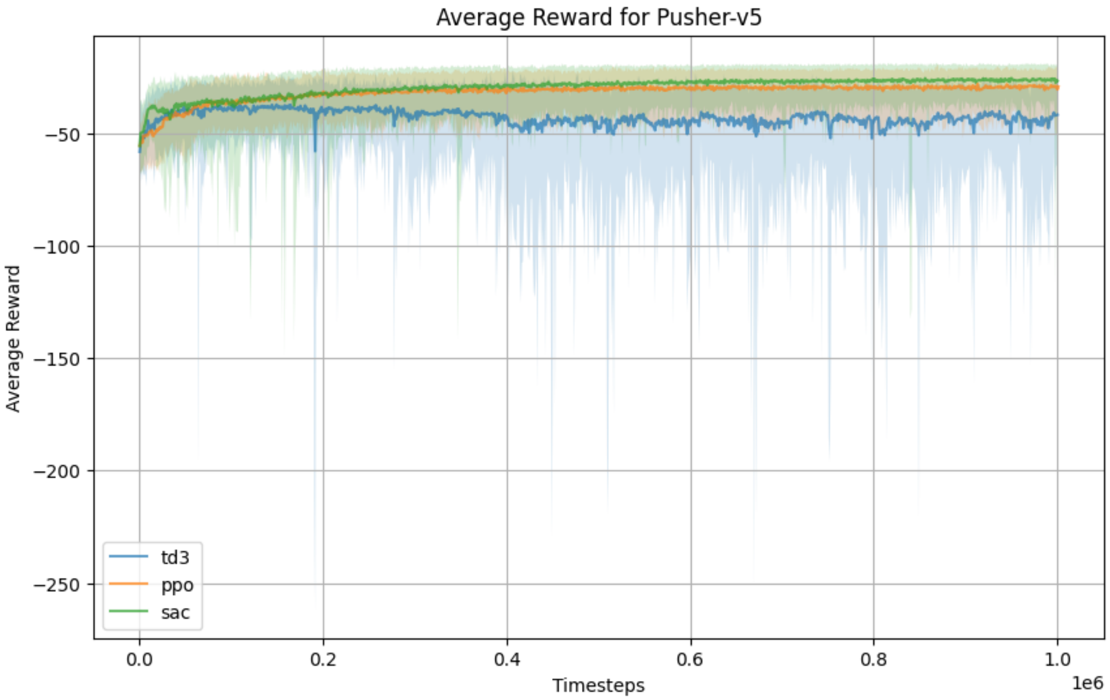

Average Reward for Ant-v5
td3
ppo
sac
Average Reward
−2000
0
2000
4000
6000
Timesteps
0.0
0.2
0.4
0.6
0.8
1.0
1e6


Average Reward for HumanoidStandup-v5
td3
ppo
sac
Average Reward
160000
140000
120000
100000
80000
60000
40000
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Timesteps


Average Reward for Swimmer-v5
td3
ppo
sac
Average Reward
100
50
0
−50
−100
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Timesteps


Average Reward for Hopper-v5
td3
ppo
sac
Average Reward
4000
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
Timesteps
1e6


Average Reward for InvertedDoublePendulum-v5
td3
ppo
sac
Average Reward
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
Timesteps
1e6


Average Reward for Pusher-v5
Average Reward
−50
−100
−150
−200
−250
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Timesteps
td3
ppo
sac