PPO Algorithm Report

I recently created a custom gym environment for Reinforcement Learning in which there is a simple maze having obstacles death pits and a goal. Then I applied PPO model to train my Agent and test it.

OUTPUT

```
---------------------------------
| rollout/           |          |
|    ep_len_mean     | 27.1     |
|    ep_rew_mean     | -0.995   |
| time/              |          |
|    fps             | 796      |
|    iterations      | 1        |
|    time_elapsed    | 2        |
|    total_timesteps | 2048     |
---------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 32.1        |
|    ep_rew_mean          | -0.811      |
| time/                   |             |
|    fps                  | 493         |
|    iterations           | 2           |
|    time_elapsed         | 8           |
|    total_timesteps      | 4096        |
| train/                  |             |
|    approx_kl            | 0.008726994 |
|    clip_fraction        | 0.0923      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.38       |
|    explained_variance   | -0.0656     |
|    learning_rate        | 0.0003      |
|    loss                 | 0.0802      |
|    n_updates            | 10          |
|    policy_gradient_loss | -0.0143     |
|    value_loss           | 0.184       |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 29.5        |
|    ep_rew_mean          | -0.525      |
| time/                   |             |
|    fps                  | 440         |
|    iterations           | 3           |
|    time_elapsed         | 13          |
|    total_timesteps      | 6144        |
| train/                  |             |
|    approx_kl            | 0.008517519 |
|    clip_fraction        | 0.121       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.36       |
|    explained_variance   | 0.104       |
|    learning_rate        | 0.0003      |
|    loss                 | 0.0718      |
|    n_updates            | 20          |
|    policy_gradient_loss | -0.0139     |
|    value_loss           | 0.158       |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 22.9        |
|    ep_rew_mean          | -0.139      |
| time/                   |             |
|    fps                  | 402         |
|    iterations           | 4           |
|    time_elapsed         | 20          |
|    total_timesteps      | 8192        |
| train/                  |             |
|    approx_kl            | 0.008899604 |
|    clip_fraction        | 0.137       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.33       |
|    explained_variance   | 0.279       |
|    learning_rate        | 0.0003      |
|    loss                 | 0.101       |
|    n_updates            | 30          |
|    policy_gradient_loss | -0.0217     |
|    value_loss           | 0.213       |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 25.4        |
|    ep_rew_mean          | 0.256       |
| time/                   |             |
|    fps                  | 387         |
|    iterations           | 5           |
|    time_elapsed         | 26          |
|    total_timesteps      | 10240       |
| train/                  |             |
|    approx_kl            | 0.010190347 |
|    clip_fraction        | 0.15        |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.27       |
|    explained_variance   | 0.248       |
|    learning_rate        | 0.0003      |
|    loss                 | 0.105       |
|    n_updates            | 40          |
|    policy_gradient_loss | -0.0242     |
|    value_loss           | 0.29        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 22.2        |
|    ep_rew_mean          | 0.368       |
| time/                   |             |
|    fps                  | 381         |
|    iterations           | 6           |
|    time_elapsed         | 32          |
|    total_timesteps      | 12288       |
| train/                  |             |
|    approx_kl            | 0.012173075 |
|    clip_fraction        | 0.168       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.21       |
|    explained_variance   | 0.253       |
|    loss                 | 0.0674      |
|    n_updates            | 50          |
|    policy_gradient_loss | -0.0221     |
|    value_loss           | 0.215       |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 18.2        |
|    ep_rew_mean          | 0.708       |
| time/                   |             |
|    fps                  | 378         |
|    iterations           | 7           |
|    time_elapsed         | 37          |
|    total_timesteps      | 14336       |
| train/                  |             |
|    approx_kl            | 0.013072819 |
|    clip_fraction        | 0.106       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.17       |
|    explained_variance   | 0.23        |
|    learning_rate        | 0.0003      |
|    loss                 | 0.165       |
|    n_updates            | 60          |
|    policy_gradient_loss | -0.0138     |
|    value_loss           | 0.217       |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 13.9        |
|    ep_rew_mean          | 0.751       |
| time/                   |             |
|    fps                  | 376         |
|    iterations           | 8           |
|    time_elapsed         | 43          |
|    total_timesteps      | 16384       |
| train/                  |             |
|    approx_kl            | 0.022492044 |
|    clip_fraction        | 0.246       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.05       |
|    explained_variance   | 0.248       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.0123     |
|    n_updates            | 70          |
|    policy_gradient_loss | -0.0254     |
|    value_loss           | 0.0966      |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 12          |
|    ep_rew_mean          | 0.81        |
| time/                   |             |
|    fps                  | 371         |
|    iterations           | 9           |
|    time_elapsed         | 49          |
|    total_timesteps      | 18432       |
| train/                  |             |
|    approx_kl            | 0.009243469 |
|    clip_fraction        | 0.1         |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.971      |
|    explained_variance   | 0.112       |
|    learning_rate        | 0.0003      |
|    loss                 | 0.0205      |
|    n_updates            | 80          |
|    policy_gradient_loss | -0.0129     |
|    value_loss           | 0.0884      |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 10.2        |
|    ep_rew_mean          | 0.789       |
| time/                   |             |
|    fps                  | 367         |
|    iterations           | 10          |
|    time_elapsed         | 55          |
|    total_timesteps      | 20480       |
| train/                  |             |
|    approx_kl            | 0.014355665 |
|    clip_fraction        | 0.179       |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.88       |
|    explained_variance   | 0.145       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.0433     |
|    n_updates            | 90          |
|    policy_gradient_loss | -0.0185     |
|    value_loss           | 0.0492      |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 9.37        |
|    ep_rew_mean          | 0.896       |
| time/                   |             |
|    fps                  | 364         |
|    iterations           | 11          |
|    time_elapsed         | 61          |
|    total_timesteps      | 22528       |
| train/                  |             |
|    approx_kl            | 0.009391051 |
|    clip_fraction        | 0.128       |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.798      |
|    explained_variance   | 0.112       |
|    learning_rate        | 0.0003      |
|    loss                 | 0.0951      |
|    n_updates            | 100         |
|    policy_gradient_loss | -0.0139     |
|    value_loss           | 0.0666      |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 7.77        |
|    ep_rew_mean          | 0.892       |
| time/                   |             |
|    fps                  | 362         |
|    iterations           | 12          |
|    time_elapsed         | 67          |
|    total_timesteps      | 24576       |
| train/                 |             |
|    approx_kl            | 0.014307864 |
|    clip_fraction        | 0.23        |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.674      |
|    explained_variance   | 0.14        |
|    learning_rate        | 0.0003      |
|    loss                 | -0.0881     |
|    n_updates            | 110         |
|    policy_gradient_loss | -0.0304     |
|    value_loss           | 0.027       |
-----------------------------------------
-----------------------------------------
| rollout/               |             |
|    ep_len_mean          | 7.23        |
|    ep_rew_mean          | 0.938       |
| time/                  |             |
|    fps                  | 360         |
|    iterations           | 13          |
|    time_elapsed         | 73          |
|    total_timesteps      | 26624       |
| train/                 |             |
|    approx_kl            | 0.013636721 |
|    clip_fraction        | 0.204       |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.543      |
|    explained_variance   | 0.116       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.0566     |
|    n_updates            | 120         |
|    policy_gradient_loss | -0.0321     |
|    value_loss           | 0.0225      |
-----------------------------------------
-----------------------------------------
| rollout/               |             |
|    ep_len_mean          | 6.71        |
|    ep_rew_mean          | 0.943       |
| time/                  |             |
|    fps                  | 358         |
|    iterations           | 14          |
|    time_elapsed         | 79          |
|    total_timesteps      | 28672       |
| train/                 |             |
|    approx_kl            | 0.016369337 |
|    clip_fraction        | 0.216       |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.411      |
|    explained_variance   | 0.215       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.0336     |
|    n_updates            | 130         |
|    policy_gradient_loss | -0.0396     |
|    value_loss           | 0.00671     |
-----------------------------------------
----------------------------------------
| rollout/               |            |
|    ep_len_mean          | 6.5        |
|    ep_rew_mean          | 0.945      |
| time/                  |            |
|    fps                  | 353        |
|    time_elapsed         | 86         |
|    total_timesteps      | 30720      |
| train/                  |            |
|    approx_kl            | 0.01753281 |
|    clip_fraction        | 0.19       |
|    clip_range           | 0.2        |
|    entropy_loss         | -0.277     |
|    explained_variance   | 0.84       |
|    learning_rate        | 0.0003     |
|    loss                 | -0.0615    |
|    n_updates            | 140        |
|    policy_gradient_loss | -0.044     |
|    value_loss           | 0.000247   |
----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 6.17        |
|    ep_rew_mean          | 0.948       |
| time/                   |             |
|    fps                  | 352         |
|    iterations           | 16          |
|    time_elapsed         | 92          |
|    total_timesteps      | 32768       |
| train/                  |             |
|    approx_kl            | 0.029048003 |
|    clip_fraction        | 0.0485      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.172      |
|    explained_variance   | 0.178       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.0442     |
|    n_updates            | 150         |
|    policy_gradient_loss | -0.0225     |
|    value_loss           | 0.00647     |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 6.05        |
|    ep_rew_mean          | 0.95        |
| time/                   |             |
|    fps                  | 351         |
|    iterations           | 17          |
|    time_elapsed         | 98          |
|    total_timesteps      | 34816       |
| train/                  |             |
|    approx_kl            | 0.003926838 |
|    clip_fraction        | 0.0212      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.105      |
|    explained_variance   | 0.183       |
|    learning_rate        | 0.0003      |
|    loss                 | 0.00191     |
|    n_updates            | 160         |
|    policy_gradient_loss | -0.012      |
|    value_loss           | 0.00645     |
-----------------------------------------
----------------------------------------
| rollout/                |            |
|    ep_len_mean          | 6.32       |
|    ep_rew_mean          | 0.947        |
| time/                   |              |
|    fps                  | 352          |
|    iterations           | 18           |
|    time_elapsed         | 104          |
|    total_timesteps      | 36864        |
| train/                  |              |
|    approx_kl            | 0.02614202   |
|    clip_fraction        | 0.103        |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.119       |
|    explained_variance   | 0.811        |
|    learning_rate        | 0.0003       |
|    loss                 | -0.0419      |
|    n_updates            | 170          |
|    policy_gradient_loss | 0.00854      |
|    value_loss           | 4.46e-05     |
----------------------------------------
----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 6.37         |
|    ep_rew_mean          | 0.926        |
| time/                   |              |
|    fps                  | 353          |
|    iterations           | 19           |
|    time_elapsed         | 110          |
|    total_timesteps      | 38912        |
| train/                  |              |
|    approx_kl            | 0.10057163   |
|    clip_fraction        | 0.342        |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.138       |
|    explained_variance   | 0.88         |
|    learning_rate        | 0.0003       |
|    loss                 | -0.0493      |
|    n_updates            | 180          |
|    policy_gradient_loss | -0.0497      |
|    value_loss           | 0.000155     |
----------------------------------------
-----------------------------------------
| rollout/                |               |
|    ep_len_mean          | 6.15          |
|    ep_rew_mean          | 0.948         |
| time/                   |               |
|    fps                  | 353           |
|    iterations           | 20            |
|    time_elapsed         | 115           |
|    total_timesteps      | 40960         |
| train/                  |               |
|    approx_kl            | 0.022954864   |
|    clip_fraction        | 0.0797        |
|    clip_range           | 0.2           |
|    entropy_loss         | -0.107        |
|    explained_variance   | 0.0882        |
|    learning_rate        | 0.0003        |
|    loss                 | 0.00329       |
|    n_updates            | 190           |
|    policy_gradient_loss | -0.0199       |
|    value_loss           | 0.0152        |
-----------------------------------------
```

```
| rollout/                |              |
|    ep_len_mean          | 6.77         |
|    ep_rew_mean          | 0.922        |
| time/                   |              |
|    fps                  | 354          |
|    iterations           | 21           |
|    time_elapsed         | 121          |
|    total_timesteps      | 43008        |
| train/                  |              |
|    approx_kl            | 0.06700014 |
|    clip_fraction        | 0.0243       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.0842      |
|    explained_variance   | 0.781        |
|    learning_rate        | 0.0003       |
|    loss                 | -0.0516      |
|    n_updates            | 200          |
|    policy_gradient_loss | 0.00309      |
|    value_loss           | 6.15e-05     |
-----------------------------------------
------------------------------------------
| rollout/                |              |
|    ep_len_mean          | 6.47         |
|    ep_rew_mean          | 0.945        |
| time/                   |              |
|    fps                  | 355          |
|    iterations           | 22           |
|    time_elapsed         | 126          |
|    total_timesteps      | 45056        |
| train/                  |              |
|    approx_kl            | 0.011297056 |
|    clip_fraction        | 0.254        |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.17        |
|    explained_variance   | 0.165        |
|    learning_rate        | 0.0003       |
|    loss                 | -0.0369      |
|    n_updates            | 210          |
|    policy_gradient_loss | -0.0404      |
|    value_loss           | 0.0068       |
------------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 6.21         |
|    ep_rew_mean          | 0.948        |
| time/                   |              |
|    fps                  | 355          |
|    iterations           | 23           |
|    time_elapsed         | 132          |
|    total_timesteps      | 47104        |
| train/                  |              |
|    approx_kl            | 0.20410594 |
|    clip_fraction        | 0.323        |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.104       |
|    explained_variance   | 0.83         |
|    learning_rate        | 0.0003       |
|    loss                 | -0.0523      |
|    n_updates            | 220          |
|    value_loss           | 0.000185   |
----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 6.14        |
|    ep_rew_mean          | 0.949       |
| time/                   |             |
|    fps                  | 356         |
|    iterations           | 24          |
|    time_elapsed         | 137         |
|    total_timesteps      | 49152       |
| train/                  |             |
|    approx_kl            | 0.016302086 |
|    clip_fraction        | 0.169       |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.0759     |
|    explained_variance   | 0.923       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.018      |
|    n_updates            | 230         |
|    policy_gradient_loss | -0.025      |
|    value_loss           | 9e-05       |
-----------------------------------------
------------------------------------------
| rollout/                |              |
|    ep_len_mean          | 6.07         |
|    ep_rew_mean          | 0.949        |
| time/                   |              |
|    fps                  | 356          |
|    iterations           | 25           |
|    time_elapsed         | 143          |
|    total_timesteps      | 51200        |
| train/                  |              |
|    approx_kl            | 0.0046497174 |
|    clip_fraction        | 0.0191       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.0429      |
|    explained_variance   | 0.967        |
|    learning_rate        | 0.0003       |
|    loss                 | -0.00558     |
|    n_updates            | 240          |
|    policy_gradient_loss | -0.0109      |
|    value_loss           | 4.05e-05     |
------------------------------------------
 100%
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51,200/50,000  [ 0:02:22 < 0:00:00 , 365 it/s ]Training finished.
Saving model to ppo_maze_model.zip
```

```
--- Testing the trained agent ---
```

```
--- Starting Test Episode 1 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
```

```
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 1 finished. Total reward: 0.95
```

```
--- Starting Test Episode 2 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 2 finished. Total reward: 0.95
```

```
--- Starting Test Episode 3 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 3 finished. Total reward: 0.95
```

```
--- Starting Test Episode 4 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 4 finished. Total reward: 0.95
```

```
--- Starting Test Episode 5 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 5 finished. Total reward: 0.95
```

```
--- Starting Test Episode 6 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 6 finished. Total reward: 0.95
```

```
--- Starting Test Episode 7 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 7 finished. Total reward: 0.95
```

```
--- Starting Test Episode 8 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
```

```
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 8 finished. Total reward: 0.95
```

```
--- Starting Test Episode 9 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 9 finished. Total reward: 0.95
```

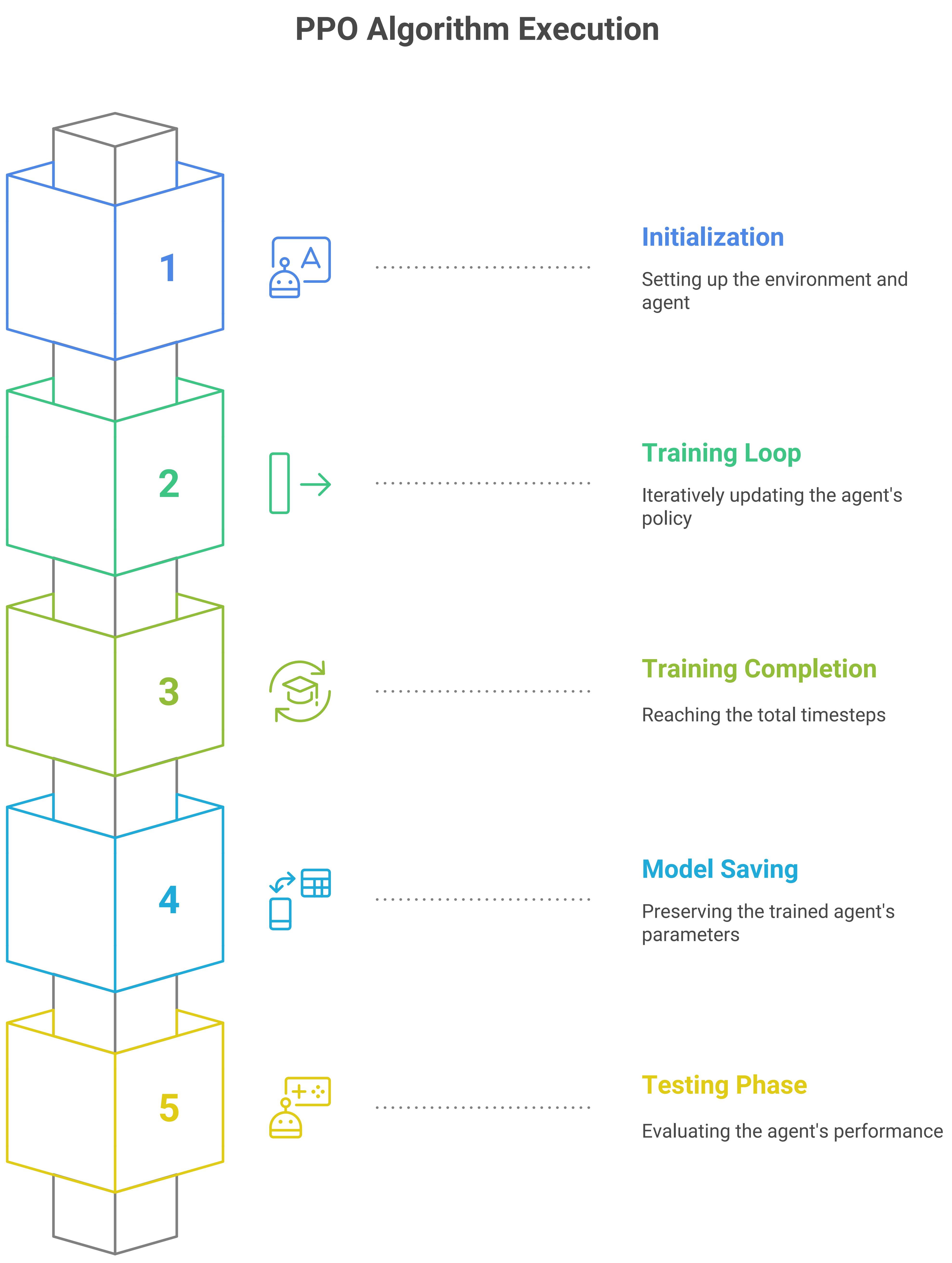
```
--- Starting Test Episode 10 ---
Step: 1, Action: 1, Obs: [1 0], Reward: -0.01, Term: False, Trunc: False
Step: 2, Action: 1, Obs: [2 0], Reward: -0.01, Term: False, Trunc: False
Step: 3, Action: 3, Obs: [2 1], Reward: -0.01, Term: False, Trunc: False
Step: 4, Action: 3, Obs: [2 2], Reward: -0.01, Term: False, Trunc: False
Step: 5, Action: 3, Obs: [2 3], Reward: -0.01, Term: False, Trunc: False
Step: 6, Action: 1, Obs: [3 3], Reward: 1.00, Term: True, Trunc: False
Episode 10 finished. Total reward: 0.95
Closing environments...
```

```
Training and testing script finished.
```

EXPLANATION:

Overall Process:

1. **Initialization:** The script starts, creates the environment, and initializes the PPO agent.
2. **Training Loop (Iterative Process):** The agent goes through multiple iterations. In each iteration:
    - **Rollout Phase:** The agent interacts with the environment for a certain number of steps (n_steps per environment, which is 2048 in your case as n_steps was likely 2048 and n_envs=1, or if you used n_steps=256 with n_envs=8, for example, though your train_ppo.py uses n_envs=1 and likely a default n_steps like 2048 or the one specified). It collects experiences (observations, actions, rewards, etc.).
    - **Training Phase:** The collected experiences are used to update the agent's policy (actor) and value function (critic) networks.
    - **Logging:** Metrics are calculated and printed.
3. **Training Completion:** The loop finishes after total_timesteps (you set it to 50,000, but the progress bar shows it completed 51,200, which is normal as it completes the current rollout).
4. **Model Saving:** The trained agent's parameters are saved.
5. **Testing Phase:** The trained agent is evaluated on a set number of new episodes in the environment, usually with rendering turned on and deterministic actions.


PPO Algorithm Execution
1
Initialization
Setting up the environment and agent
2
Training Loop
Iteratively updating the agent's policy
3
Training Completion
Reaching the total timesteps
4
Model Saving
Preserving the trained agent's parameters
5
Testing Phase
Evaluating the agent's performance

Understanding the Logged Metrics:

These metrics are typically averaged over the episodes completed within the *current rollout phase*.

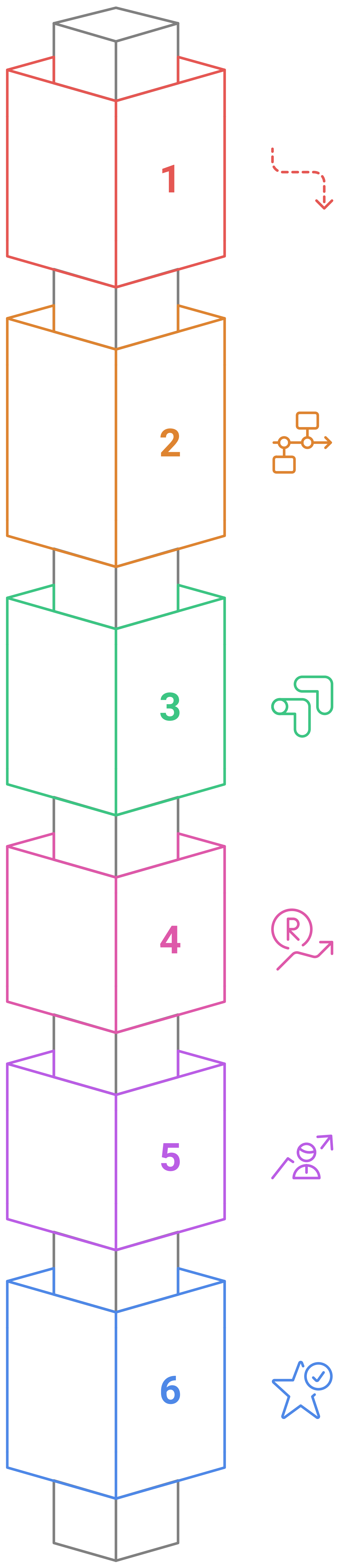
rollout/ section: (Information about the data collection phase)

- **ep_len_mean**: The average length (number of steps) of episodes completed during the most recent rollout.
  - **What's happening:** Initially, it's 27.1, 32.1, etc. This means the agent is taking, on average, that many steps to finish an episode (either by reaching the goal 'G' or a death pit 'D').
  - **Trend:** You want this to **decrease** if the agent is learning to reach the goal more efficiently (fewer steps). And indeed, it goes down significantly to around 6.07 by the end. This is a very good sign!

- **ep_rew_mean**: The average total reward obtained per episode during the most recent rollout.
    - **What's happening:** It starts negative (-0.995, -0.811). This is because:
        - Reaching 'G' gives +1.0 reward.
        - Reaching 'D' gives -1.0 reward.
        - Each step gives 0 reward (unless you uncommented the -0.01 step penalty. *Looking at your test output, it seems you did enable the -0.01 step penalty because the rewards are 0.95, 0.94, etc., instead of a clean 1.0*).

        - If the agent hits death pits frequently or takes many steps to sometimes find the goal, the average reward will be low or negative.
    - **Trend:** You want this to **increase** towards the maximum possible reward. It starts negative, then increases to around 0.949 by the end. This is excellent! It means the agent is consistently reaching the goal and minimizing penalties. The value 0.949 with a 6-step solution (0.95 = 1.0 - 5 * 0.01) makes perfect sense if the optimal path is 6 steps long from S to G.

Agent Learning Progress



Initial Episode Length

The agent starts with longer episodes, indicating inefficient navigation.

Decreasing Episode Length

The agent begins to shorten episodes, showing improved efficiency.

Final Episode Length

The agent achieves the shortest episodes, demonstrating optimal navigation.

Initial Reward

The agent starts with negative rewards, reflecting frequent errors.

Increasing Reward

The agent's rewards improve, indicating better decision-making.

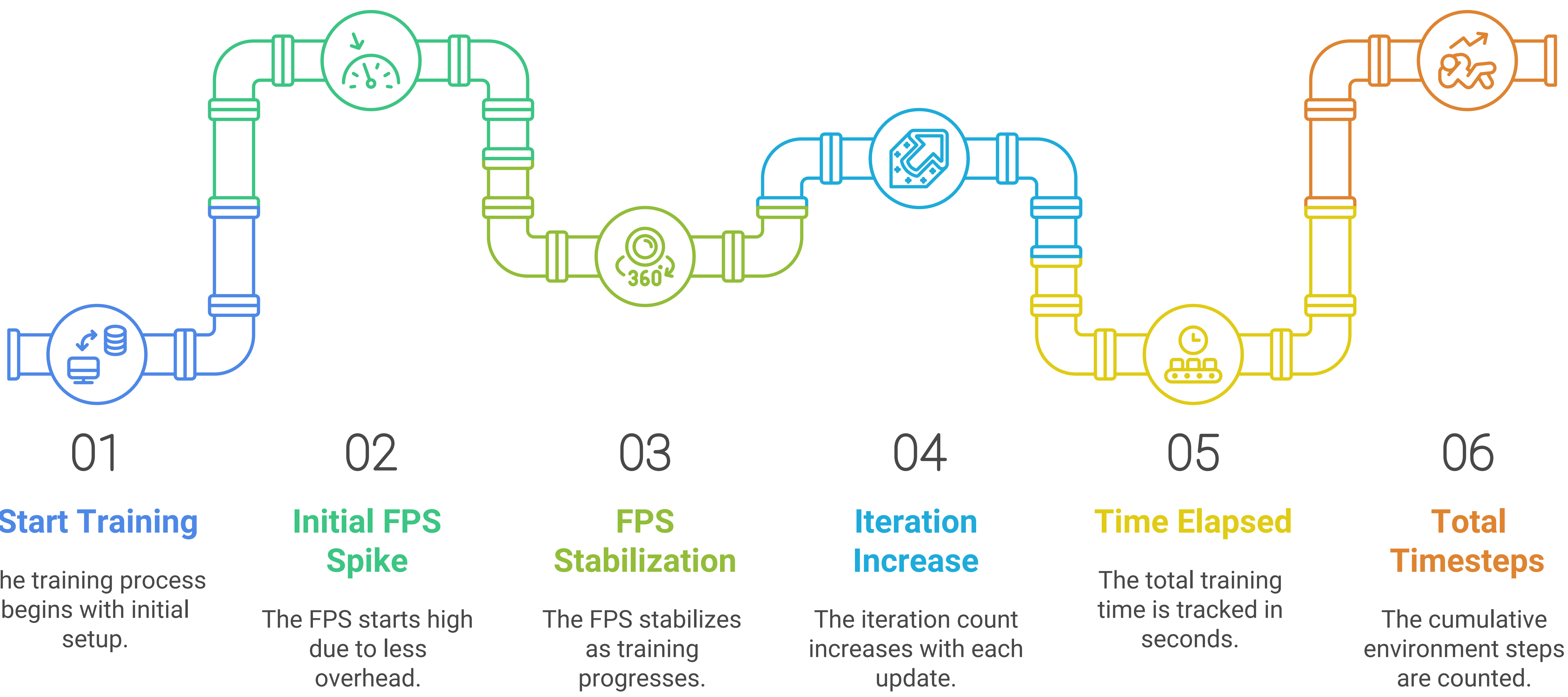
Final Reward

The agent reaches near-optimal rewards, demonstrating successful learning.

time/ section: (Information about computational performance and progress)

- **fps**: Frames Per Second. This indicates how many environment steps are being processed per second (including both environment interaction and model inference/training).
    - **What's happening:** Starts high (796) and then stabilizes around 350-380. The initial FPS might be higher because the first iteration might involve less network training overhead or just setup.
    - **Trend:** Generally, higher is better for faster training, but it depends on the complexity of the environment and model. It seems stable here.
- **iterations**: The number of times the agent has completed a full rollout and performed a training update.
    - **What's happening:** Increases by 1 for each block of output.
- **time_elapsed**: Total wall-clock time in seconds since training started.
- **total_timesteps**: The cumulative number of environment steps taken across all episodes and all parallel environments so far. This is the main counter for training progress.
    - **What's happening:** Increases by n_steps * n_envs with each iteration. If n_steps was 2048 and n_envs=1 (default for PPO if not specified and train_ppo.py uses n_envs=1), then it increases by 2048 each time. This matches your output (2048, 4096, 6144, ...).

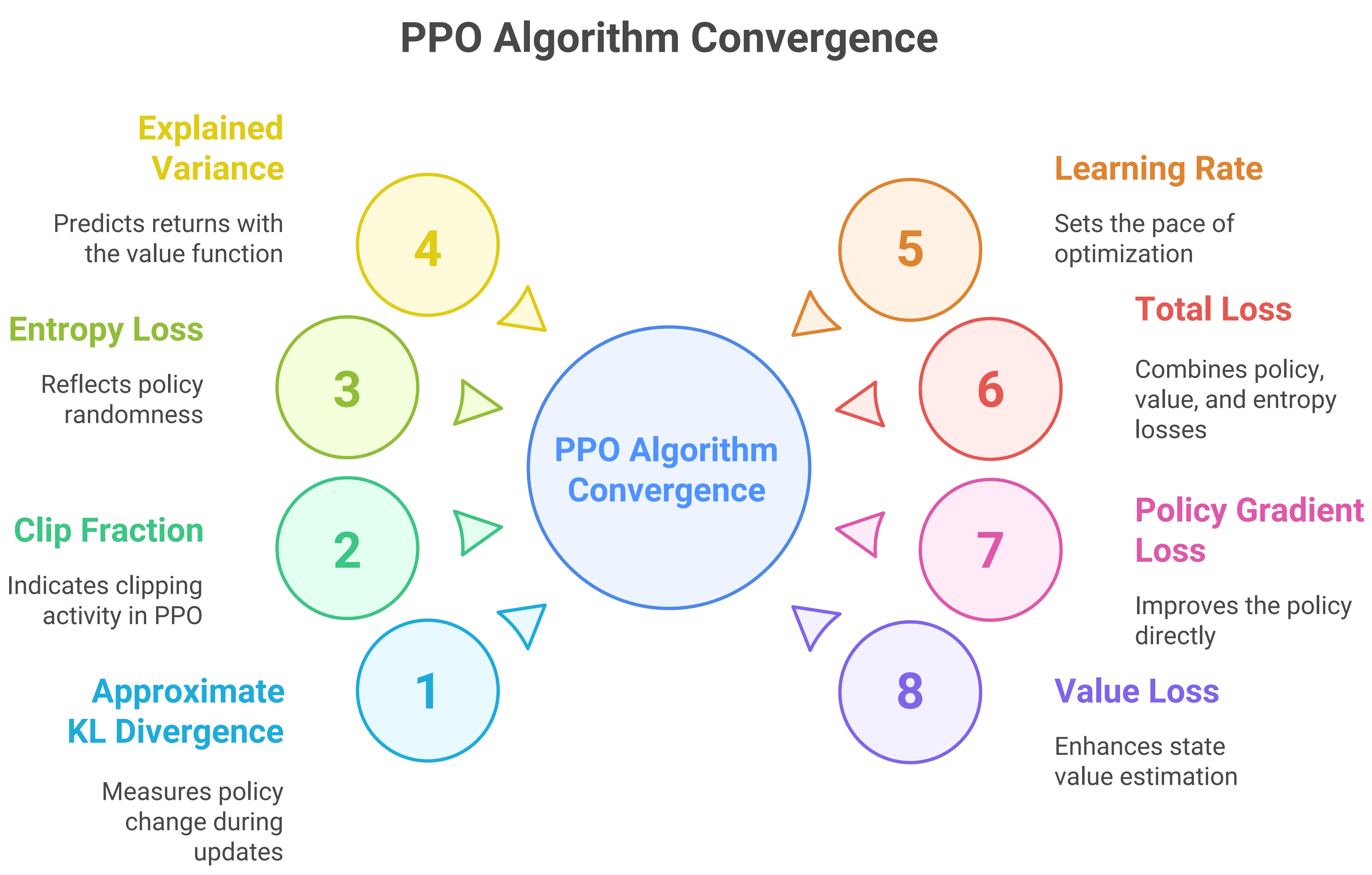
PPO Algorithm Training Process


01
Start Training
The training process begins with initial setup.
02
Initial FPS Spike
The FPS starts high due to less overhead.
03
FPS Stabilization
The FPS stabilizes as training progresses.
04
Iteration Increase
The iteration count increases with each update.
05
Time Elapsed
The total training time is tracked in seconds.
06
Total Timesteps
The cumulative environment steps are counted.

train/ section: (Information about the PPO neural network training phase)*(This section appears after the first rollout because training happens after data is collected)*

- **approx_kl (Approximate KL Divergence)**: An estimate of the Kullback-Leibler divergence between the policy before the update and the policy after the update.
    - **What it is:** Measures how much the policy changed during the update.

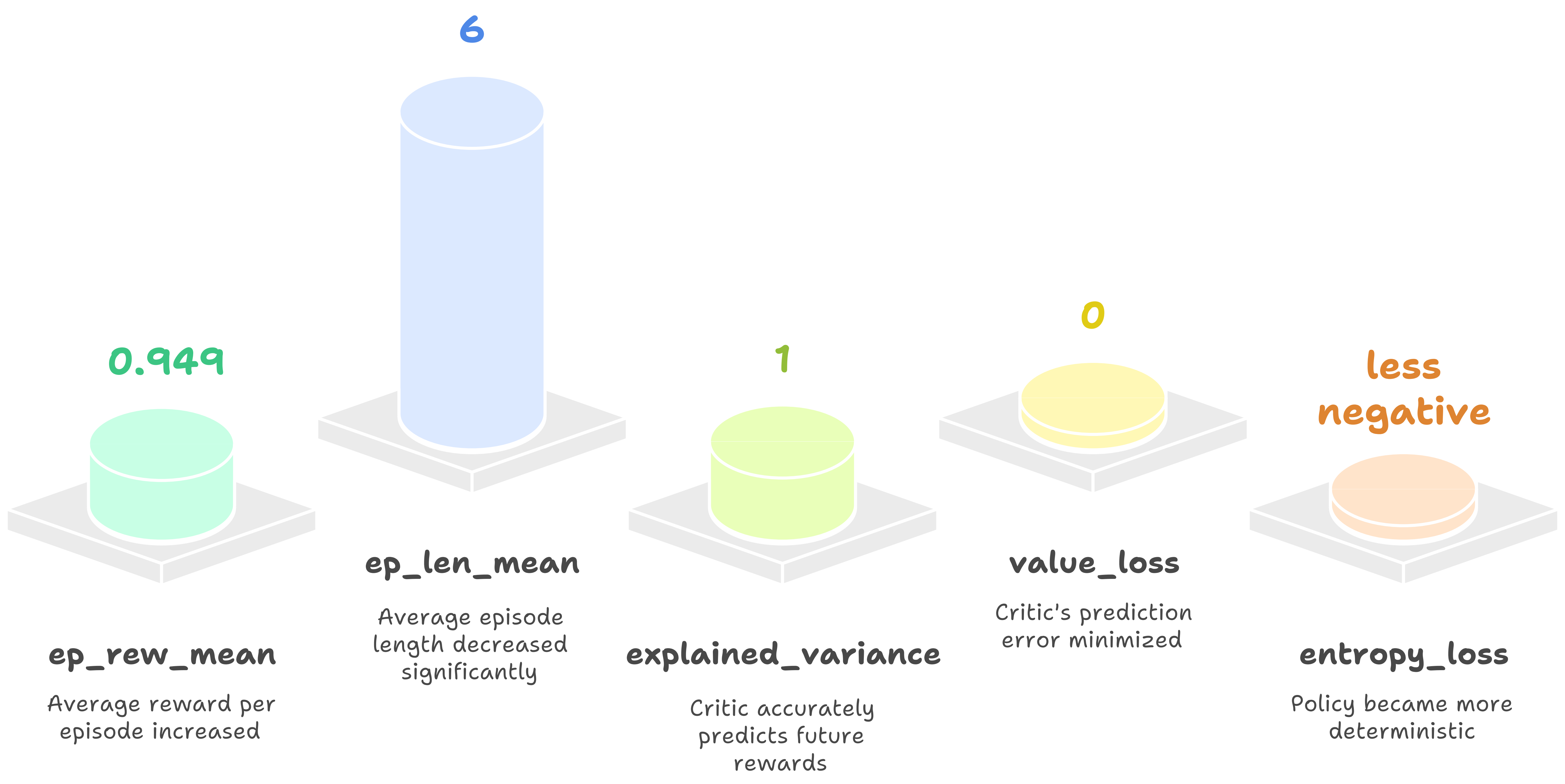
    - **Trend:** You generally don't want this to be too large, as PPO tries to keep policy changes within a "trust region." If it's consistently very high, it might indicate instability (e.g., learning rate too high). Your values (e.g., 0.008, 0.01, occasionally spiking to 0.1 or 0.2) are generally in a reasonable range for PPO.
- **clip_fraction**: The fraction of times the PPO clipping objective was active.
    - **What it is:** PPO clips the probability ratio between the new and old policy to prevent excessively large updates. This metric tells you how often that clipping happened.
    - **Trend:** If this is very high (e.g., close to 1), it might mean the policy is trying to change too much, and the clipping is constantly kicking in. If it's very low, the policy changes are small. Your values (e.g., 0.09 to 0.3) are common.
- **clip_range**: The current value of the clipping parameter epsilon in PPO.
    - **What it is:** This is typically a hyperparameter you set (default is often 0.2). It might be annealed (changed over time) in some implementations, but here it's constant at 0.2.
- **entropy_loss**: The negative of the policy's entropy.
    - **What it is:** Entropy measures the randomness of the policy. The ent_coef encourages higher entropy (more exploration). The loss term is negative entropy, so minimizing this loss means maximizing entropy.
    - **Trend:** As the agent becomes more confident and its policy becomes more deterministic (less random exploration), the entropy decreases, so entropy_loss (negative entropy) will increase (become less negative). Your values go from -1.38 towards -0.0429. This is expected and good: the agent starts by exploring (high entropy, very negative entropy_loss) and then becomes more certain about the best actions (lower entropy, less negative entropy_loss).
- **explained_variance**: A measure of how well the value function (critic) predicts the actual returns.
    - **What it is:** Ranges from -infinity to 1. A value of 1 means the value function perfectly predicts the returns. A value of 0 means it's doing no better than predicting the mean return. Negative values mean it's doing worse than predicting the mean.
    - **Trend:** You want this to **increase towards 1**. It starts negative (-0.0656), which is common early in training, and improves significantly to 0.967. This indicates your value function is learning well to predict how good states are.
- **learning_rate**: The current learning rate used by the optimizer.
    - **What it is:** Your learning_rate is constant at 0.0003. Some schedules might decrease it over time.
- **loss (Total Loss)**: The overall PPO loss, which is a combination of policy loss, value loss, and entropy bonus.
    - **Trend:** You generally want this to **decrease**, but its absolute value isn't always directly interpretable on its own. It's more important to look at its components and the ep_rew_mean. Your values fluctuate but show a general downward trend initially and then stabilize/fluctuate.
- **n_updates**: The cumulative number of times the PPO optimization step (gradient descent on the collected batch) has been performed.
    - **What it is:** Increases by n_epochs (e.g., 10 if that was your setting) after each rollout. This matches your output (10, 20, 30...).
- **policy_gradient_loss (or pg_loss)**: The part of the loss function that directly tries to improve the policy (actor).
    - **Trend:** This is the core PPO objective (the clipped surrogate objective). Its sign can be positive or negative depending on the formulation. You want it to improve in a way that leads to higher rewards.
- **value_loss (or vf_loss)**: The loss for the value function (critic). Typically a mean squared error between predicted values and target values (e.g., GAE).
    - **Trend:** You want this to **decrease**, indicating the critic is getting better at estimating state values. Your value_loss decreases dramatically from 0.184 down to very small values like 4.05e-05. This is excellent and consistent with the high explained_variance.


PPO Algorithm Convergence
Explained Variance
Predicts returns with the value function
4
Entropy Loss
Reflects policy randomness
3
Clip Fraction
Indicates clipping activity in PPO
2
Approximate KL Divergence
Measures policy change during updates
1
PPO Algorithm Convergence
Learning Rate
Sets the pace of optimization
5
Total Loss
Combines policy, value, and entropy losses
6
Policy Gradient Loss
Improves the policy directly
7
Value Loss
Enhances state value estimation
8

ANALYSIS of AGENT

- **Successful Learning:** The agent clearly learned to solve the maze.
  - ep_rew_mean increased from negative values to around 0.949 (very close to the max achievable reward if there's a small step penalty).
  - ep_len_mean decreased from ~30 steps to ~6 steps. This is a huge improvement in efficiency.
- **Good Value Function:** explained_variance approaching 1 and value_loss becoming very small indicate the critic learned to accurately predict future rewards. This is crucial for PPO's stability and performance.
- **Policy Improvement:** The decreasing ep_len_mean and increasing ep_rew_mean show the policy (actor) improved significantly.
- **Entropy Decay:** The entropy_loss increased (became less negative), showing the agent's policy became more deterministic as it found a good solution.

Performance Metrics of PPO Agent


ep_rew_mean
Average reward per episode increased
ep_len_mean
Average episode length decreased significantly
explained_variance
Critic accurately predicts future rewards
value_loss
Critic's prediction error minimized
less negative
entropy_loss
Policy became more deterministic