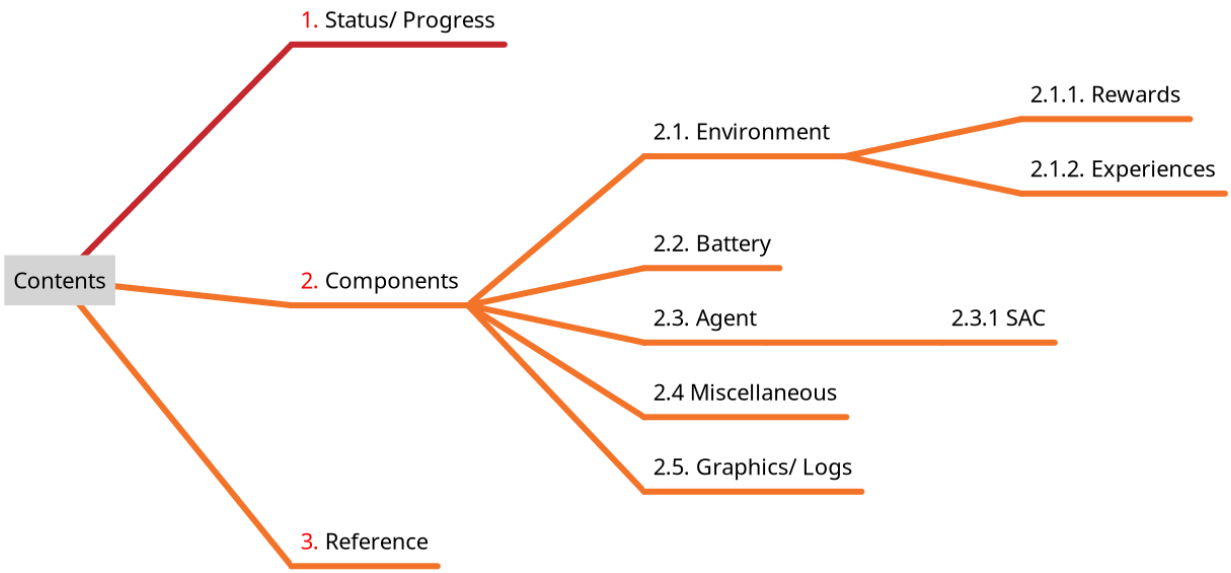
Training Prosumer Agents with Reinforcement Learning.

>>> ****Biweekly Report 4.** ($25^{th}\,Apr - 6^{th}\,May : 2024$)



---

1. Status/ Progress

Current Iteration

- [x] simplified reward functions
- [x] policy gradient learn/inference with SAC
- [x] organized episodic experiences as csv
- [x] improvement in structure, tests, & typos fixes

Next Iteration (Plan)

- [ ] vectorized environment for parallel learning
- [ ] test out integration of replay buffer for experience replay

---

2. Components

Taking feedback from weekly catchups into account and updated understanding of the system following changes were made to different components.

2.1. Environment Development

2.1.1. **Rewards**

- reward functions were simplified as suggested and run
  - **discouraging the difference as** $-abs(\text{net exchange})$
  - discouraging the difference as $-(\text{net exchange})^2$ given,
  - $\text{net exchange} = \text{Power Household} + \text{Power PV} + \text{action}$
- the resulting grahics for minimized absolute exchange as reward is in the Graphics/Logs section.

2.1.2 **Experiences**

- CSV files were extracted per episode with original timestep indices during sampling action.

2.2. Battery Module

- Minor fixes were made to the issue where the return of `get_current_soc` method was not as intended.

2.3. Agent

- Algorithm in use:
  - Proximal Policy Optimization
  - Soft Actor Critic

2.4. Miscellaneous

- Improvement in running main.py by adding argument parser: `python main.py --learn --save_stats --save_graphics` to run the policy learning, saving agents experiences to action that

were sampled from the policy, and saving resulting figures from experiences/dataset respectively as shown in the section Graphics/Logs below.

- Added test for environment and battery units.

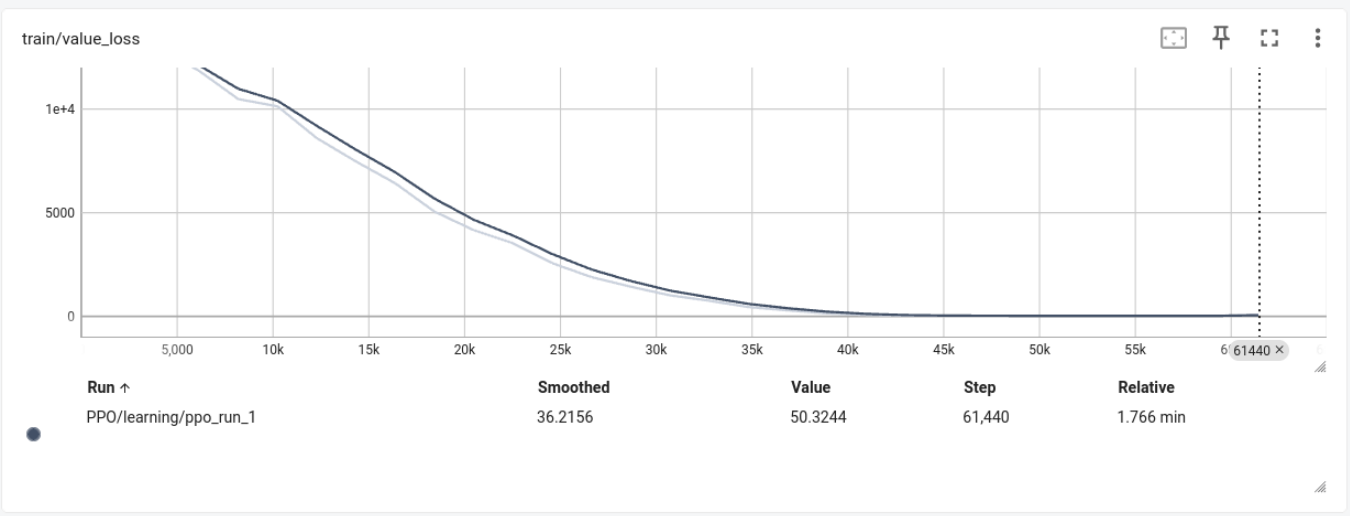
2.5. Graphics/ Logs

```
hyper_params: dict = {
    "dataset_len": total_dataset
    "learning_rate": 0.0003,
    "total_timesteps": 60000
    "episode_limit": 50
}
```

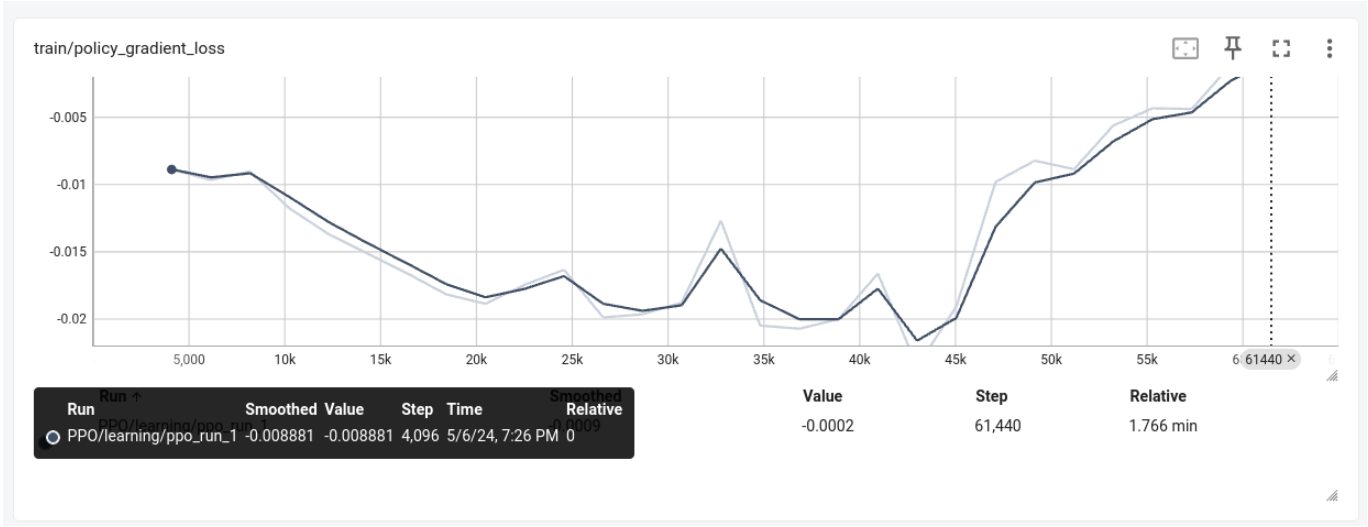
During Learning

PPO

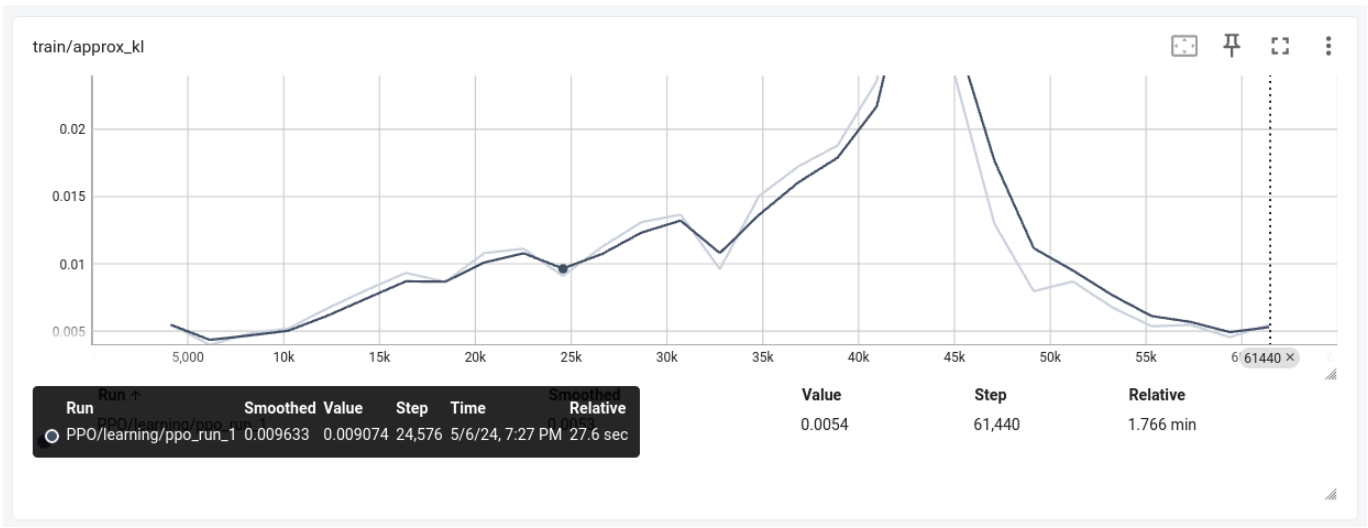
Value Loss : Error between value function outputs and estimate



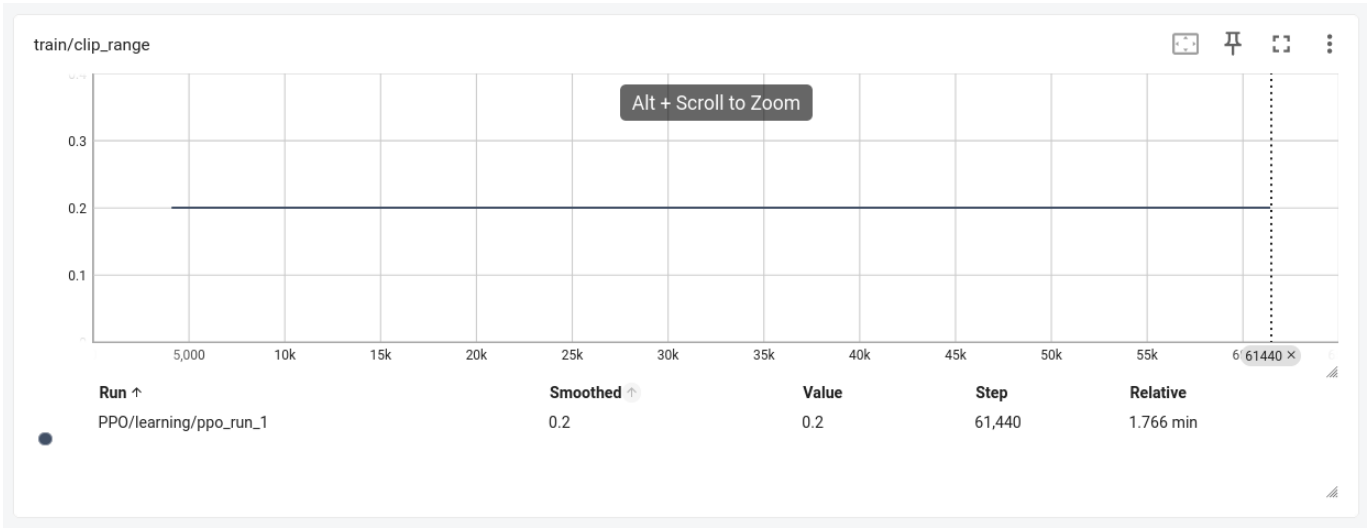
Policy Gradient Loss : Loss function policy network uses



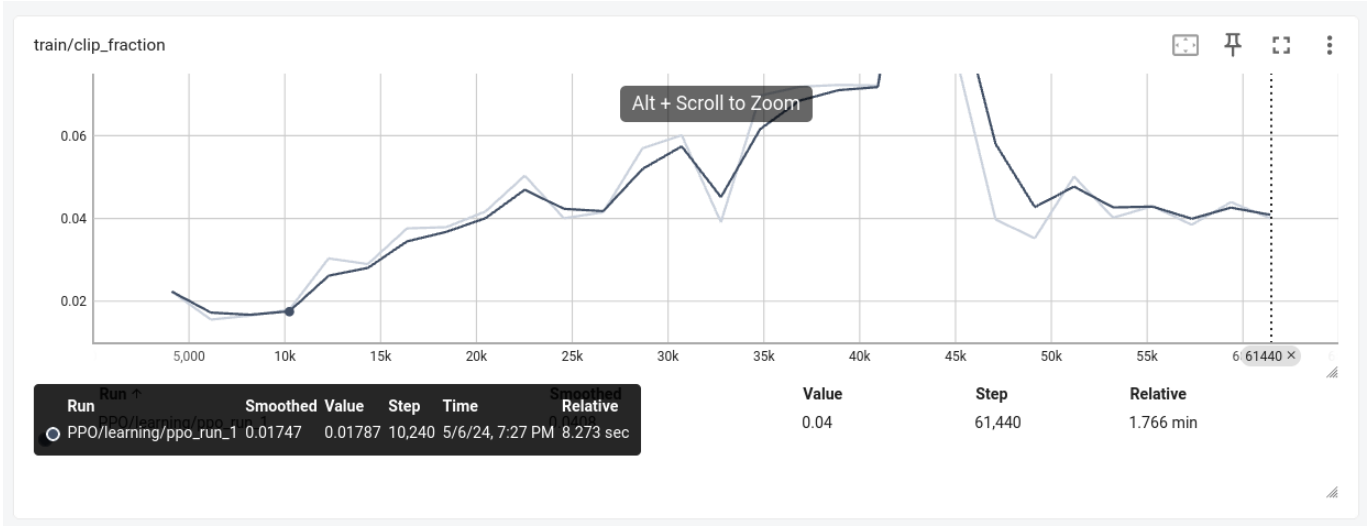
Approximate KL : Approximate mean KL divergence b/w policies



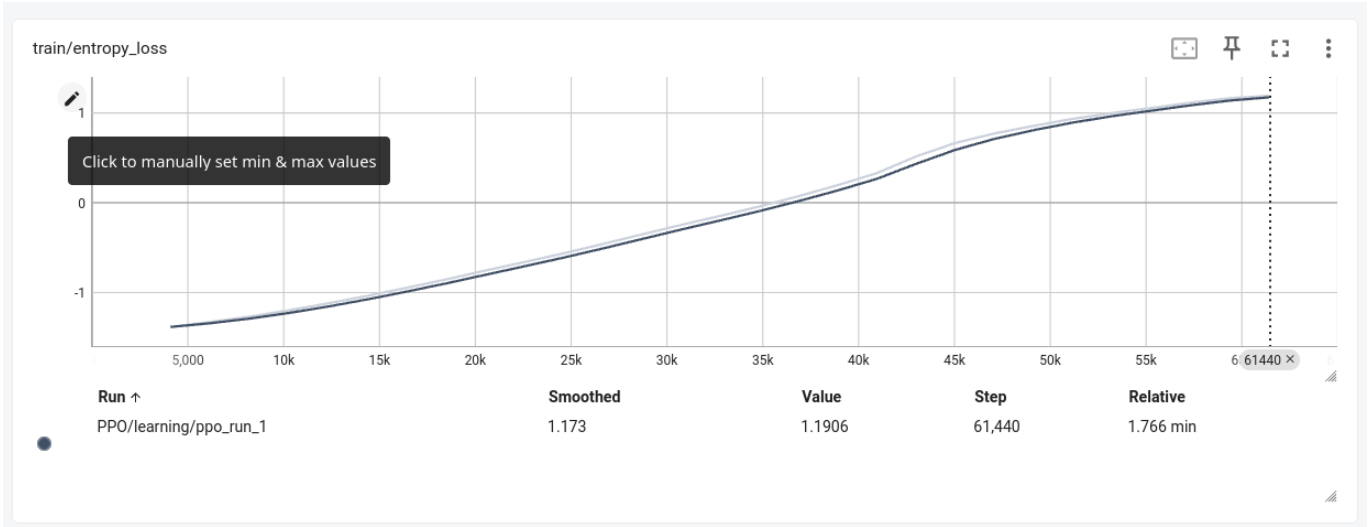
Clip Range : clip range of surrogate loss of PPO



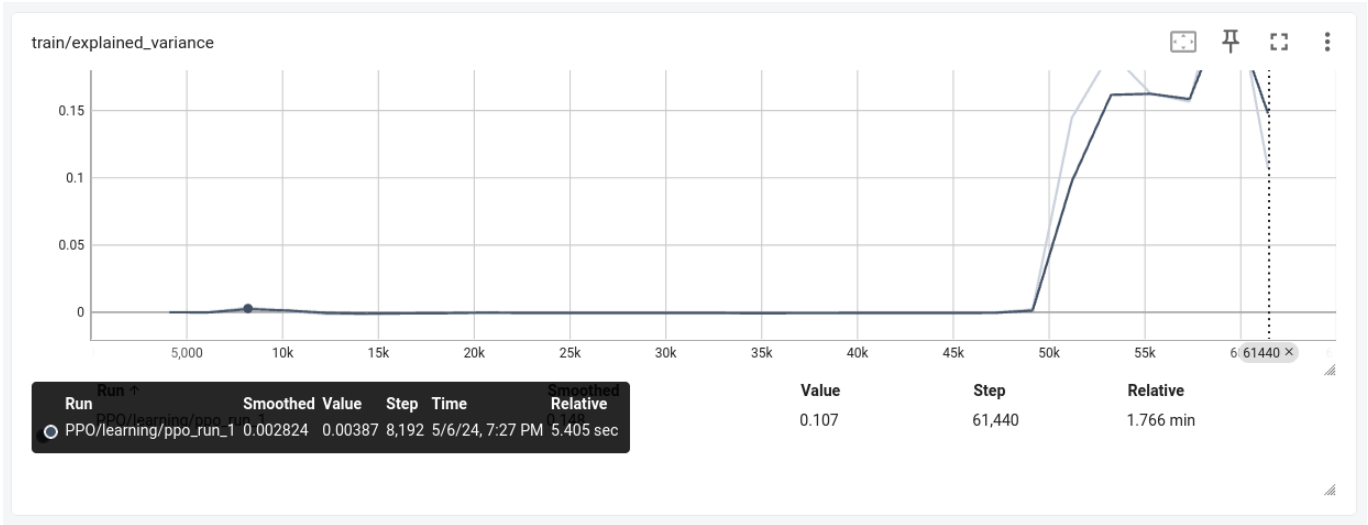
Clip Fraction : average of clipped / inside the clipping region



Entropy Loss : Mean value of Entropy loss



Explained Variance: Fraction of the return variance explained by the value function



SAC

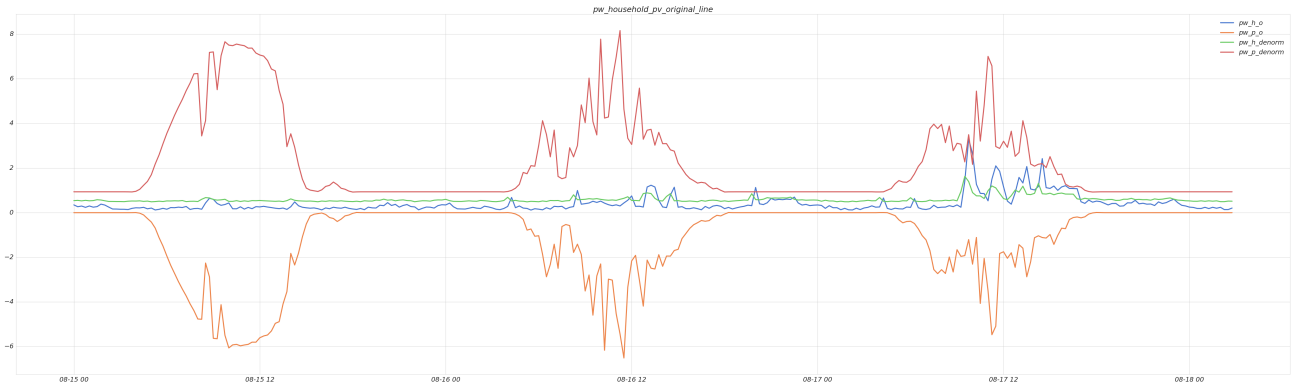
The tensorboard logging for SAC model is still to be resolved

---

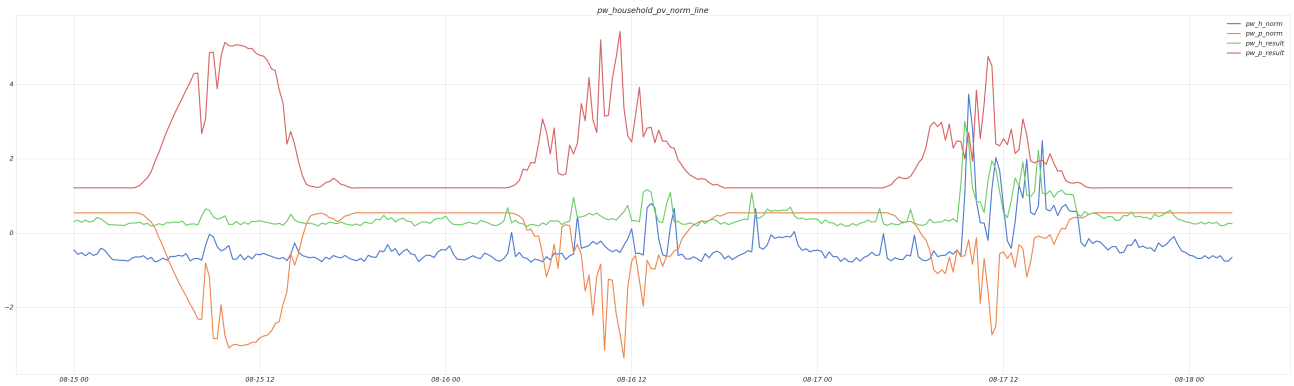
During Sampling

Generated PV/ Demanded Household

- Before normalization/ After Denormalization



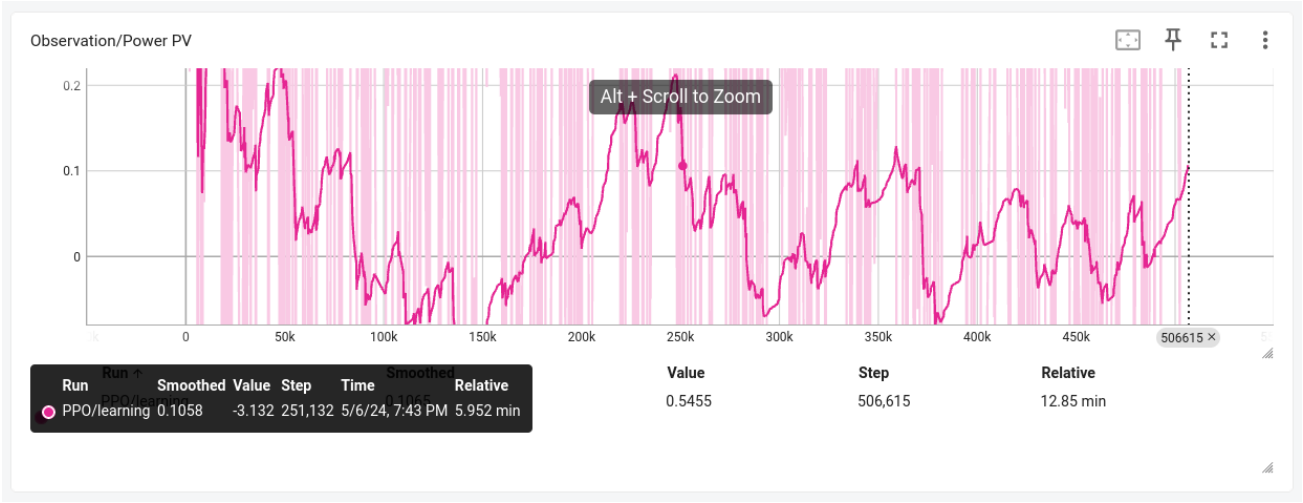
- After normalization / Before Denormalization

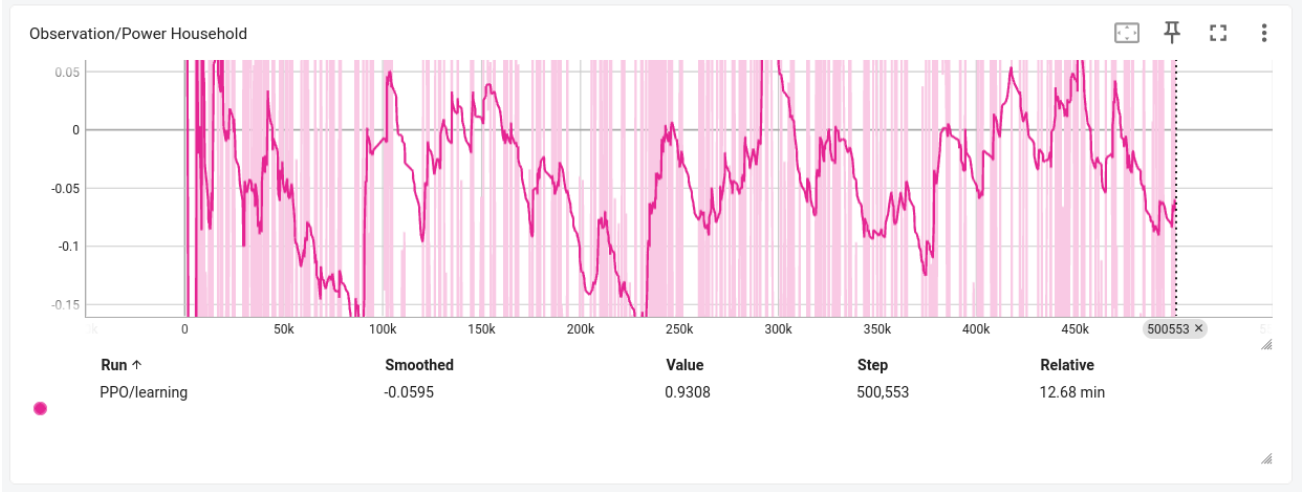


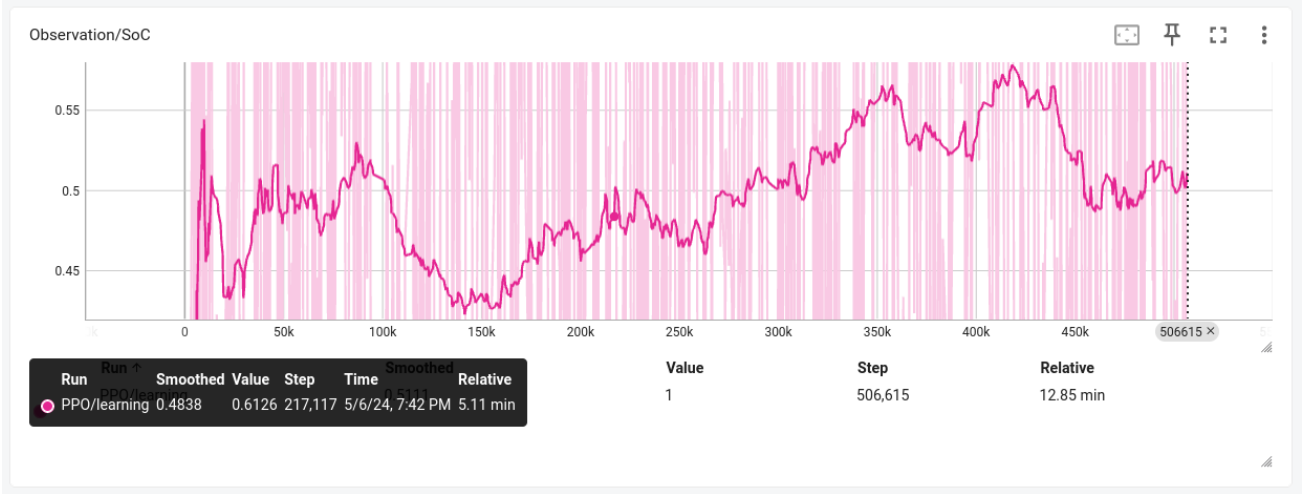
PPO

Tensorboard Logs with Smoothing(Exponential Mooving Average)

- Inputs


Observation/Power PV
Alt + Scroll to Zoom
0.2
0.1
0
0
50k
100k
150k
200k
250k
300k
350k
400k
450k
506615
Run
Smoothed
Value
Step
Time
Relative
PPO/learning
0.1058
-3.132
251,132
5/6/24, 7:43 PM
5.952 min
Value
0.5455
Step
506,615
Relative
12.85 min

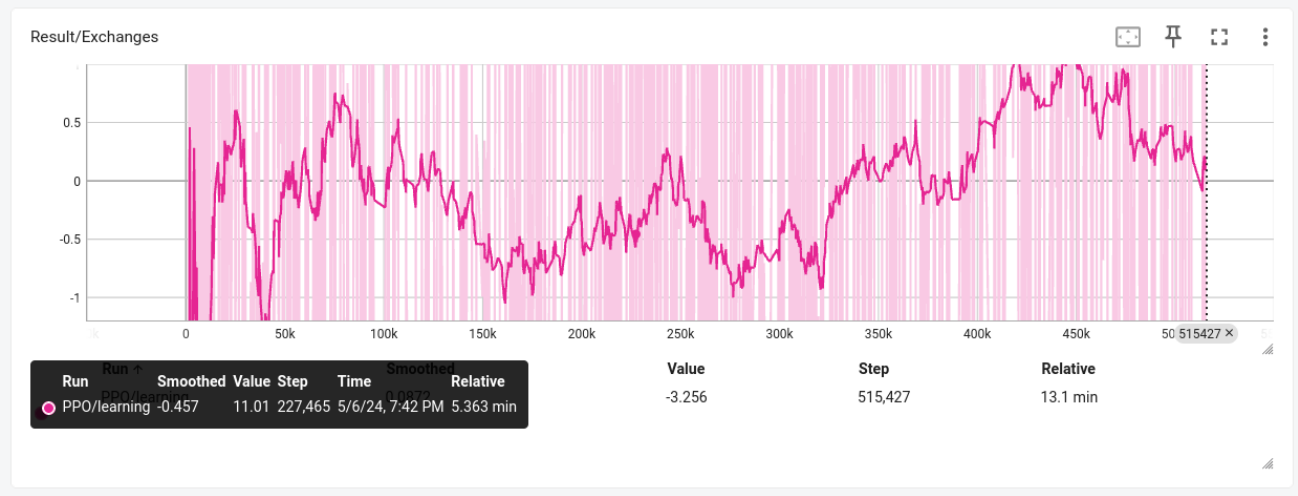

Observation/Power Household
0.05
0
-0.05
-0.1
-0.15
0
50k
100k
150k
200k
250k
300k
350k
400k
450k
500553
Run
Smoothed
Value
Step
Relative
PPO/learning
-0.0595
0.9308
500,553
12.68 min


Observation/SoC
0.55
0.5
0.45
0
50k
100k
150k
200k
250k
300k
350k
400k
450k
506615
Run
Smoothed
Value
Step
Time
Relative
PPO/learning
0.4838
0.6126
217,117
5/6/24, 7:42 PM
5.11 min
Value
1
Step
506,615
Relative
12.85 min

- Action

- Reward

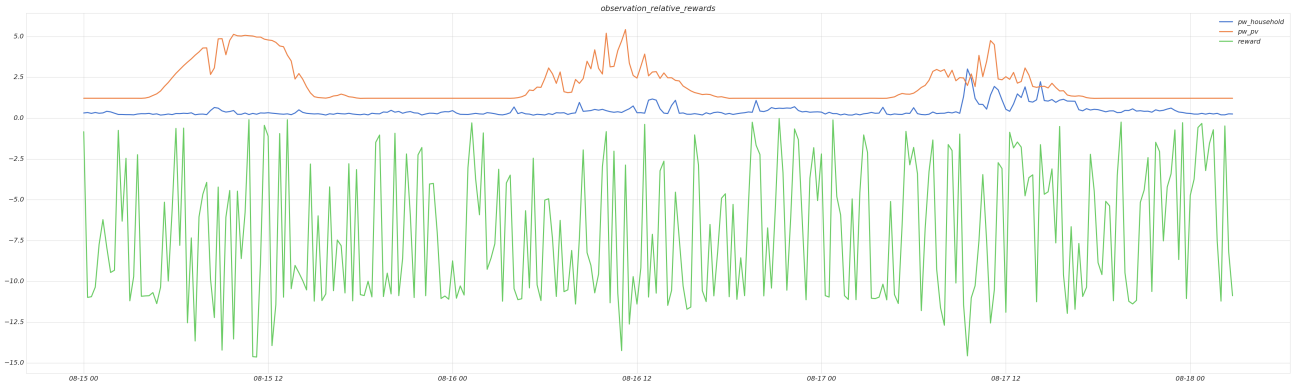
- Exchanges



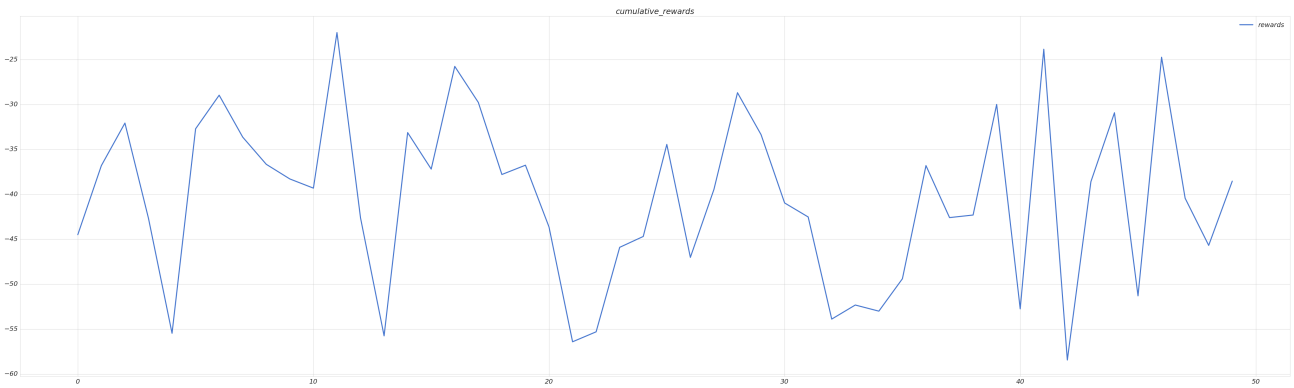
- Overall experiences

- Observation relative action

- Observation relative rewards

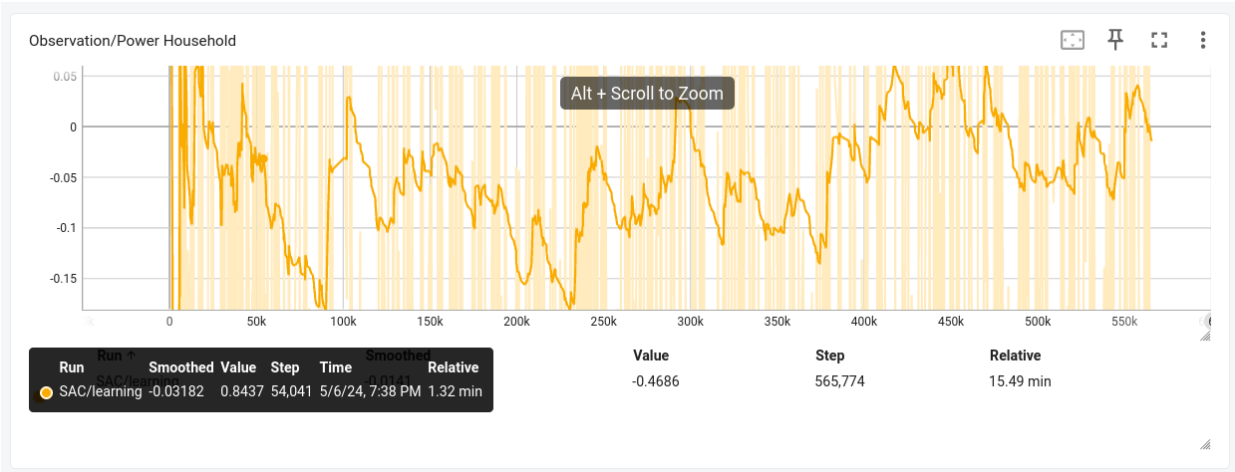


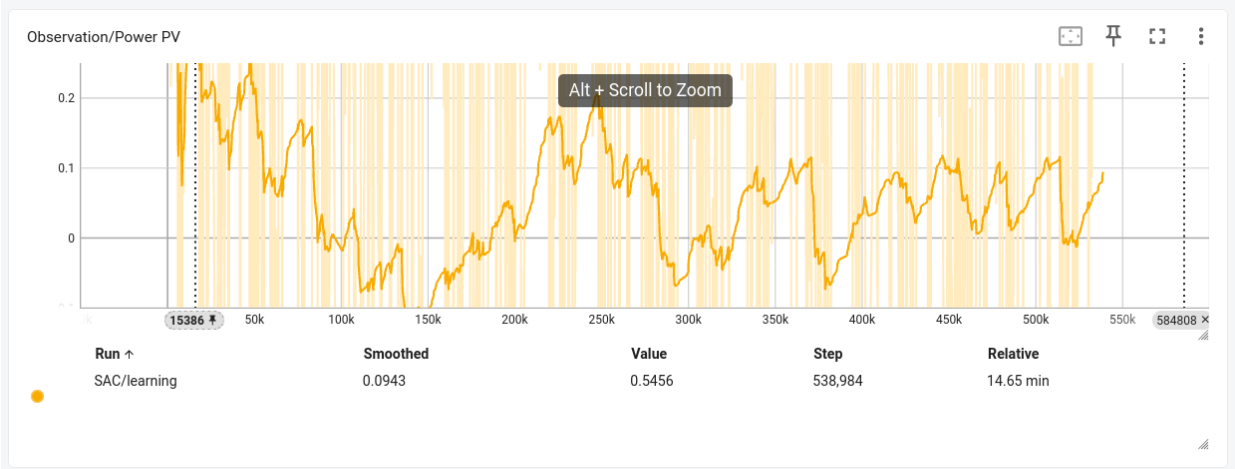
- Cumulative rewards

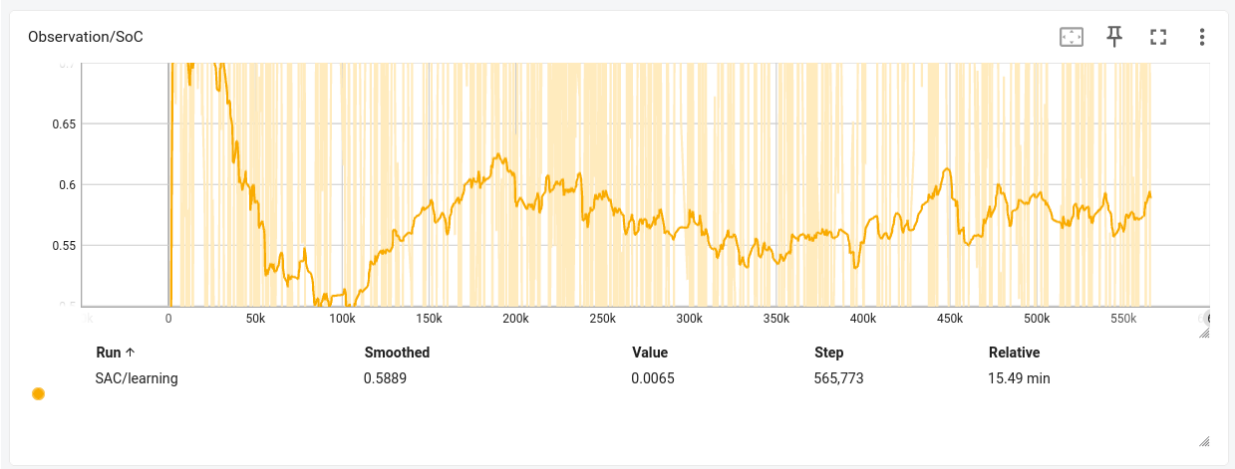


SAC

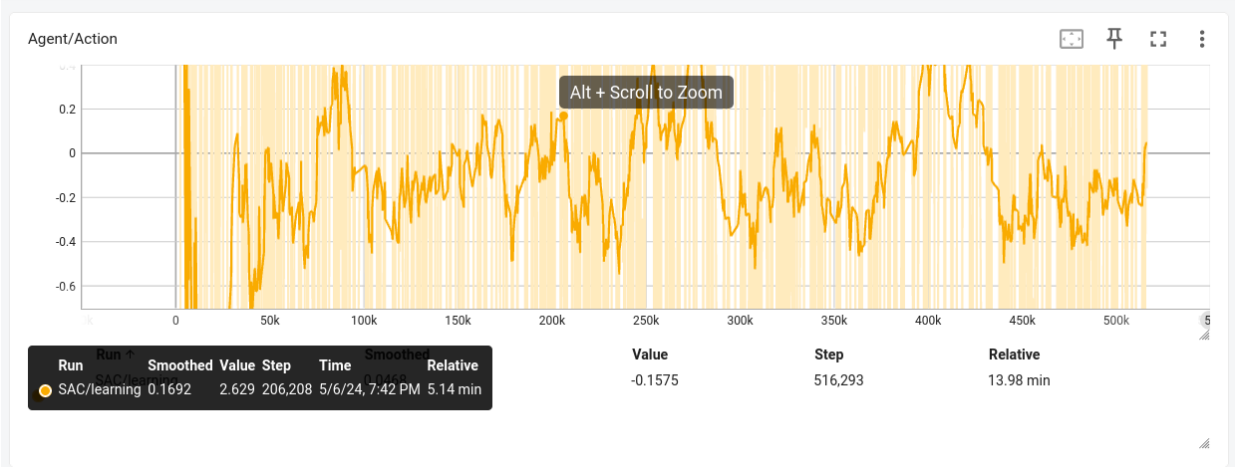
- Tensorboard Logs with Smoothing(Exponential Mooving Average)
    - Inputs



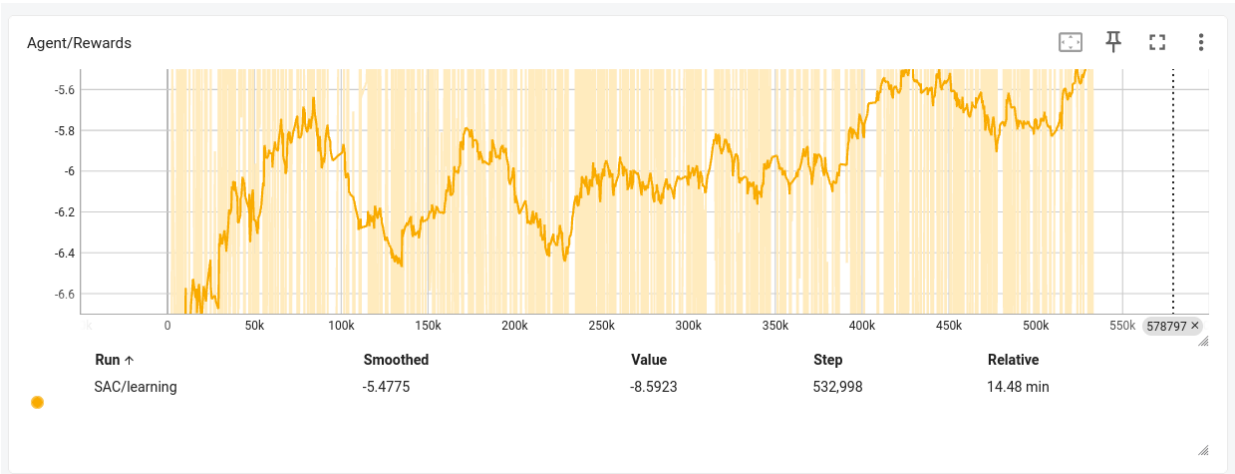




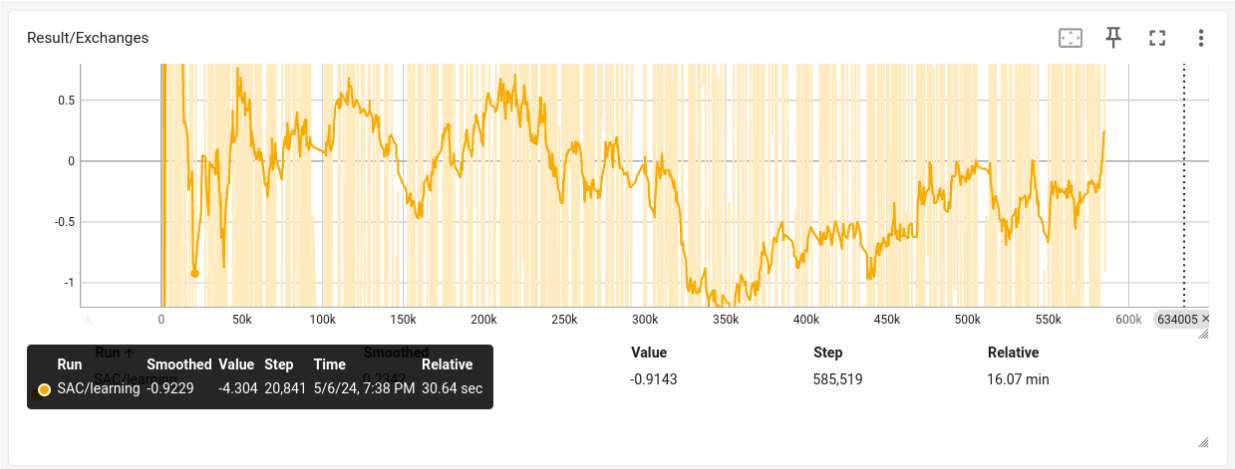
- Action



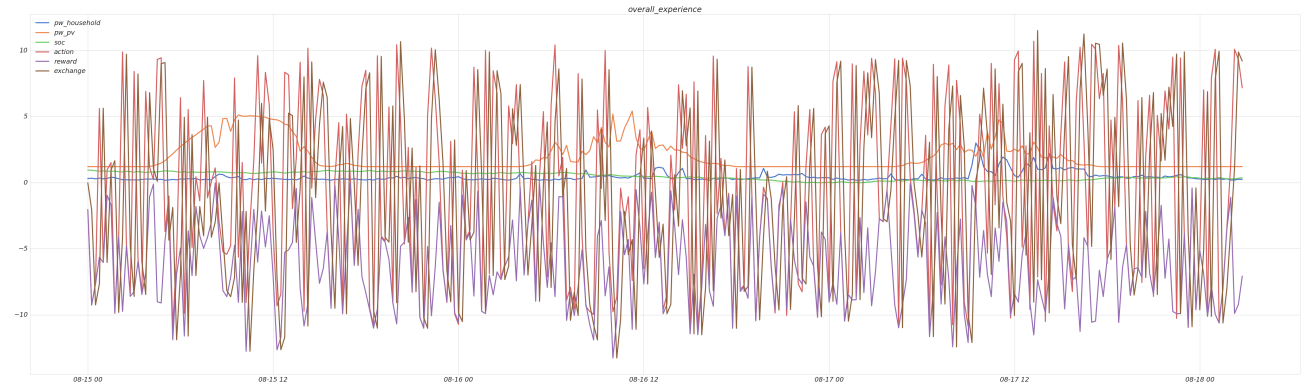
- Reward`



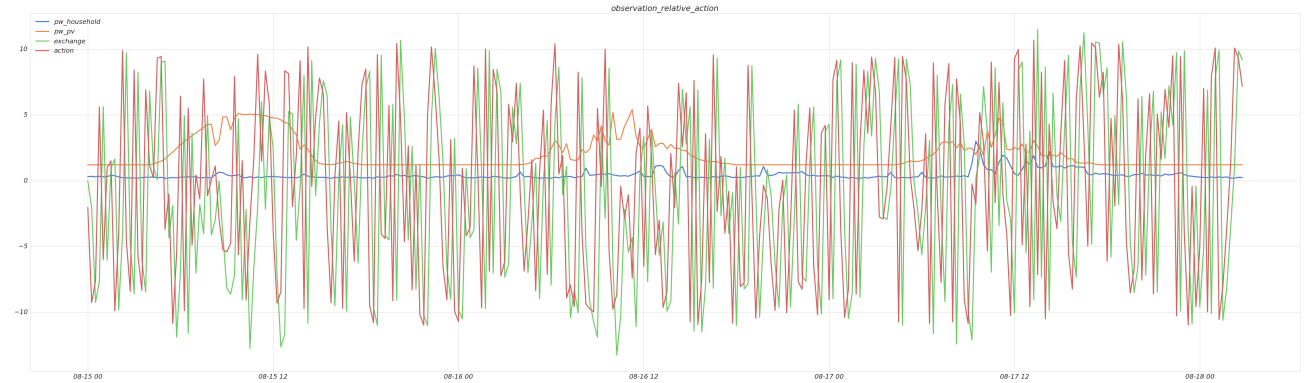
- Exchanges



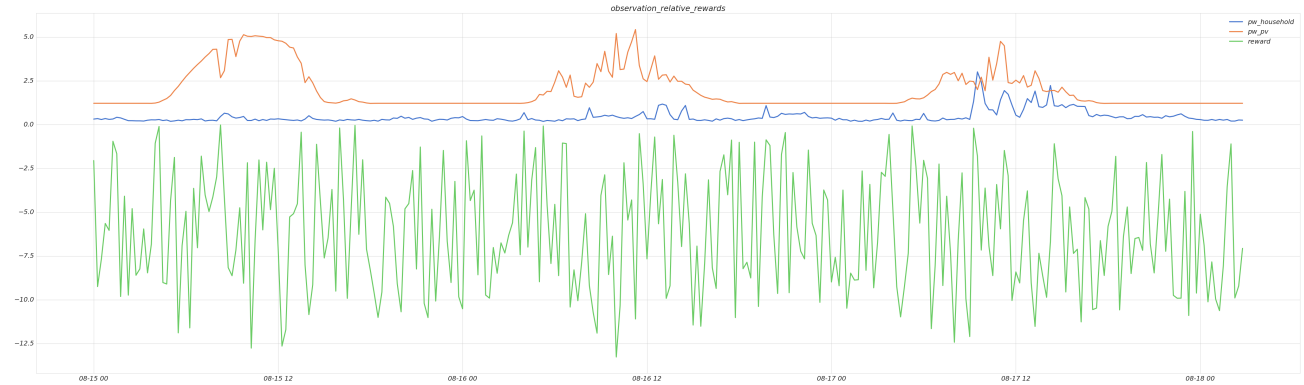
- Overall experiences



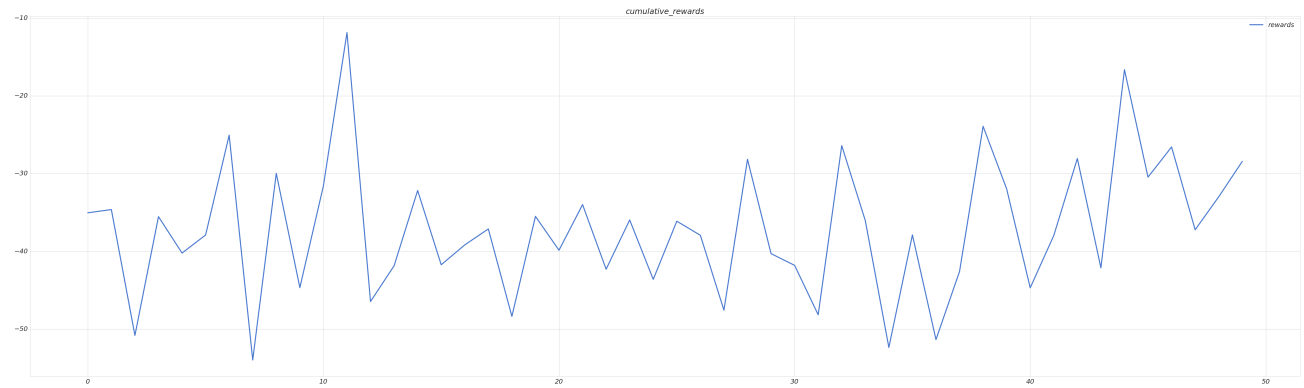
- Observation relative action



- Observation relative rewards



- Cumulative rewards



5. References

- [1.] SAC — Stable Baselines3 2.3.2 documentation
- [2.] Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor arXiv:1801.01290 [cs.LG]