

✓ Mục tiêu Bước 4

Sử dụng CPUSchedulerEnv từ bước 3 và huấn luyện 1 agent (ví dụ: PPO) để học chính sách scheduling tối ưu bằng cách tương tác với môi trường.

🧱 Chuẩn bị

Bạn đã có:

- `cpuscheduler_env.py` ✓
- Môi trường chuẩn Gym ✓
- Baseline model từ XGBoost ✓ (dùng nếu cần warm-start)
- Thư viện `stable-baselines3` ✓

✓ Gợi ý cấu trúc Bước 4: `train_rl_agent.py`

🔧 1. Import + Env

```
python
Sao chépChỉnh sửa
from stable_baselines3 import PPO
from stable_baselines3.common.callbacks import EvalCallback
from cpuscheduler_env import CPUSchedulerEnv

env = CPUSchedulerEnv(num_processes=5, max_steps=50)
```

🧠 2. Khởi tạo PPO Agent

```
python
Sao chépChỉnh sửa
model = PPO(
    policy="MlpPolicy",
    env=env,
    verbose=1,
    n_steps=64,      # mỗi rollout
    batch_size=32,
    gamma=0.95,
    learning_rate=3e-4,
    tensorboard_log="./ppo_cpu_scheduler_tensorboard/"
)
```

3. Huấn luyện agent

```
python
Sao chépChỉnh sửa
# Tùy chọn: theo dõi với EvalCallback
eval_env = CPUSchedulerEnv(num_processes=5)
eval_callback = EvalCallback(
    eval_env,
    best_model_save_path="./best_model/",
    log_path="./logs/",
    eval_freq=1000,
    deterministic=True,
    render=False
)

model.learn(total_timesteps=50000, callback=eval_callback)
```

4. Lưu mô hình

```
python  
Sao chépChỉnh sửa  
model.save("ppo_cpu_scheduler")
```

 File hoàn chỉnh: **train_rl_agent.py**