**1. Enrich the feature and model architecture**

**Use deeper, sequence‑aware feature extractors.**  
Your current environment flattens a short lookback window (10 days) and passes it to an MLP. A systematic study of RL agents for portfolio optimization found that *CNN‑based feature extractors with longer lookback windows significantly outperformed MLPs* in terms of risk‑adjusted returns[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=the%20core%20instruments,advancements%20in%20financial%20machine%20learning). CNNs excel at capturing local temporal patterns in price series and interactions across instruments[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=of%20annualized%20return%20and%20risk,whether%20the%20primary%20focus%20is). Long Short‑Term Memory (LSTM) or Transformer models can also capture long‑term dependencies. In stable‑baselines3 you can either:

* Switch to CnnPolicy or implement a custom CNN encoder that reshapes the state into a multi‑channel tensor (e.g., each technical indicator is a channel, days are the spatial dimension)[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=of%20annualized%20return%20and%20risk,whether%20the%20primary%20focus%20is).
* Increase the lookback\_period from 10 days to 20–30 days; the MDPI study reports better risk‑adjusted performance with longer lookbacks[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=the%20core%20instruments,advancements%20in%20financial%20machine%20learning).
* Alternatively, preprocess the data with an autoencoder or LSTM outside the RL model and feed compressed state vectors to the agent.

**Experiment with other RL algorithms.**  
Proximal Policy Optimization (PPO) is stable but can be conservative. The same MDPI study observed that *DQN and DDPG agents consistently outperformed market benchmarks such as the S&P 500*[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=the%20core%20instruments,advancements%20in%20financial%20machine%20learning) and that DDPG showed the best trade‑off between return and risk[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=of%20annualized%20return%20and%20risk,maximizing%20returns%20or%20minimizing%20risk). Stable‑baselines3 supports DDPG, TD3 and SAC. DDPG operates in continuous action space and can learn finer weight adjustments; TD3 adds target‑policy smoothing, which reduces overestimation bias. Running a few of these algorithms and comparing them to PPO may reveal a better fit for your data.

**2. Adjust the rebalancing schedule and include transaction costs**

**Rebalance less frequently and penalize turnover.**  
Continuous daily rebalancing increases transaction costs and can erode returns. The MDPI paper found that *periodic rebalancing (e.g., every 10–20 days) was more efficient than continuous rebalancing because it reduced slippage and transaction costs*[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=the%20core%20instruments,RL%20agents%20to%20dynamic%20market). Your current rebalance\_period of 10 days is reasonable, but you might experiment with 21‑day (monthly) or 63‑day (quarterly) rebalancing.

**Model transaction costs and slippage.**  
Real‑world trading incurs costs. You can approximate this by deducting a fixed fraction (e.g., 5–10 bp) of trade value whenever the weights change. In the MDPI study, transaction costs and slippage were set at 5 bp and 2 bp respectively[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=One%20of%20the%20central%20challenges,essential%20in%20rapidly%20changing%20market), and including them changed the optimal rebalancing frequency. In your environment’s step() method, subtract a cost proportional to the L1 change in weights since the previous step. This discourages unnecessary turnover and may produce more realistic, risk‑adjusted returns.

**3. Improve reward functions and risk management**

**Explore risk‑adjusted objectives.**  
Your environment currently uses “mean‑CVaR,” which penalizes the conditional tail loss. Research shows that different reward formulations can materially change performance. For example, a KTH thesis tested daily return, Sharpe ratio and Value‑at‑Risk (VaR) rewards; each reward outperformed the benchmark in one market regime but not another[diva-portal.org](https://www.diva-portal.org/smash/get/diva2:1849139/FULLTEXT01.pdf#:~:text=%E2%80%A2%20and%20the%20trade,05%2023). Consider:

* **Sharpe‑ratio‑based reward:** reward = (return – risk‑free) / portfolio standard deviation. You can approximate standard deviation using an exponential moving variance.
* **Log‑utility reward:** reward = log(1 + portfolio\_return), which naturally penalizes large drawdowns.
* **Probabilistic Sharpe ratio:** the working paper on RL allocation suggests evaluating significance using the probabilistic Sharpe ratio to account for estimation error[econstor.eu](https://www.econstor.eu/bitstream/10419/271267/1/qms-rp2023-01.pdf#:~:text=variance%20approaches,the%20use%20of%20RL%20systems). This can help you avoid overfitting to spurious high Sharpe ratios.

Tune the risk coefficient (risk\_coefficient) and the alpha level for CVaR (e.g., 0.01 vs 0.05) to see how strongly the agent penalizes tail risk; the literature finds that RL agents can reduce left‑tail risk even when outperformance isn’t universal[econstor.eu](https://www.econstor.eu/bitstream/10419/271267/1/qms-rp2023-01.pdf#:~:text=variance%20approaches,the%20use%20of%20RL%20systems).

**4. Enhance feature engineering**

**Add macro and cross‑sectional features.**  
At present, you compute volatility, momentum and moving averages from ETF prices. The MDPI study notes that adding technical indicators provided marginal improvements[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=the%20core%20instruments,advancements%20in%20financial%20machine%20learning). Try adding:

* Factor exposures (e.g., Fama–French factors, sector indicators, or macro variables like interest rates and VIX).
* Cross‑sectional rankings (e.g., each ETF’s momentum relative to the group).
* Measures of drawdown, skewness or realized volatility.

A richer state space may allow the agent to detect patterns not captured by simple moving averages.

**Standardize across cross‑sections.**  
You standardize each window separately. Ensure there’s no look‑ahead bias: fit the StandardScaler only on the training set and apply it to validation/prediction. Also consider using robust scaling (median/IQR) if your features contain outliers.

**5. Refine the network and hyperparameter search**

* **Increase network depth and width.** A two‑layer 256‑unit MLP may be insufficient. The MDPI article suggests that deeper networks (three to four layers) with more units improved performance for CNN and MLP extractors[mdpi.com](https://www.mdpi.com/1999-4893/17/12/570#:~:text=the%20core%20instruments,advancements%20in%20financial%20machine%20learning). Try architectures like (512, 256, 128) with dropout and layer normalization.
* **Use Optuna for hyperparameter tuning.** Random sampling is coarse. Libraries like Optuna can search learning rate, batch size, n\_steps, discount factor gamma, entropy coefficient, and clipping range. Explore more seeds and iterations to reduce variance.
* **Parallelize environments.** PPO benefits from multiple parallel environments (e.g., 4–8) to stabilize gradient estimates. Use make\_vec\_env with n\_envs > 1 to speed up training and improve sample diversity.

**6. Rethink action constraints and baseline integration**

* **Relax weight clipping and long–short limits.** The environment clamps weights to [–0.2, 0.8] per asset and imposes a 1.20/0.20 long/short split. Such tight bounds may force the agent toward equal weights. Experiment with broader limits (e.g., –0.5 to 1.5) or allow the agent to output continuous weights that are then normalized to sum to 1 (with optional leverage constraint).
* **Use change‑based actions.** Instead of predicting absolute weights, predict weight *adjustments* relative to the previous portfolio. This can simplify learning and incorporate transaction costs naturally.
* **Portfolio‑enhancement RL.** Recent papers propose using RL to adjust a baseline portfolio rather than build one from scratch. For example, a PPO‑based agent can be trained to apply multiplicative tilts to an equal‑weighted or factor‑weighted baseline (see “regret‑optimized portfolio enhancement” literature). This reduces the search space and often yields more stable improvements.

**7. Ensure robust evaluation**

* **Walk‑forward analysis with more windows and seeds.** Ten‑year training windows may not expose the agent to enough market regimes. Consider overlapping windows (e.g., 5‑year rolling) and more outer iterations.
* **Use statistical tests.** Alongside the t‑test you already compute, evaluate deflated or probabilistic Sharpe ratios[econstor.eu](https://www.econstor.eu/bitstream/10419/271267/1/qms-rp2023-01.pdf#:~:text=variance%20approaches,the%20use%20of%20RL%20systems) to ensure any outperformance is statistically significant after adjusting for multiple trials. This is important for a dissertation to avoid data‑mining bias.
* **Include an equal‑risk‑contribution benchmark.** RL sometimes improves risk profiles even when returns match the 1/N portfolio[econstor.eu](https://www.econstor.eu/bitstream/10419/271267/1/qms-rp2023-01.pdf#:~:text=variance%20approaches,the%20use%20of%20RL%20systems). Compare drawdowns and downside risk to demonstrate value.