Team members

- Navya Priya Nandimandalam, navyapriya_n@tamu.edu, 136004813
- Alisha Raj, alisharaj2607@tamu.edu, 735007776

Topic

Portfolio management - compare standard and improved Tabular Q-Learning (Double Q, UCB- ε) with Deep RL Baselines (DDPG, TD3, PPO) for risk-aware portfolio rebalancing in a Gym-based trading environment.

Proposed starting point

We will **fork** the Gym-style crypto portfolio environment from **wassname/rl-portfolio-management** (MIT). We'll keep the environment and tests, modernize dependencies, and add our own agents/wrappers.

What we plan to do (and how)

We'll fork the wassname Gym-style portfolio repo and add a discrete rebalancing wrapper that turns portfolio allocation into a clean MDP. On top, we'll implement tabular Q-learning for the discrete env and DDPG for a matched continuous-action variant. Using identical rewards (returns minus costs, drawdown penalty) and walk-forward splits, we'll compare stability and performance across seeds and transaction-cost regimes. We will extend our comparison by including two additional baseline algorithms (TD3 and PPO) from Stable-Baselines3 to benchmark against modern actor–critic and policy-gradient methods. On the tabular side, we'll integrate algorithmic enhancements such as UCB-ε exploration (to balance exploration and exploitation more effectively) and Double Q-learning (to reduce overestimation bias). We'll ship a reproducible evaluation harness with baselines (equal-weight, momentum) and metrics (Return, Sharpe, MaxDD, Turnover).

Questions

- **Why sequential?** Today's allocation changes tomorrow's wealth and future feasible trades due to transaction costs and position limits -> decisions compound over time.

- **State space (S):** recent per-asset returns/volatility (e.g., last 20 bars), current portfolio weights (including cash), and a lightweight "regime" flag from rolling mean/vol.

- **Action space (A):**
  - **Discrete version (for tabular Q/DQN):** per-asset rebalancing actions in {−Δ, 0, +Δ} (e.g., Δ = 5%), then project to the probability simplex (weights ≥0, sum=1).
  - **Continuous version (for DDPG/TD3/PPO):** target weight vector in a Box space, projected to the simplex.
- **Transition (P):** apply action → rebalance → pay transaction costs → observe next prices → update wealth and weights.
- **Reward (R):** one-step portfolio log-return **minus** λ·transaction_cost, with a small penalty on excess drawdown. (We'll tune λ; default 10–50 bps costs). For the tabular agent, we will also test Double Q-learning to reduce overestimation bias and UCB-ε exploration, to improve exploration–exploitation balance.

Scope of work (what we'll code)

1. **Env wrapper (discrete)**: new Gym wrapper `CryptoPortfolioDiscrete-v0` adding {-Δ,0,+Δ} actions + simplex projection + costs.
2. **Tabular Q-learning (baseline RL):** NumPy implementation with a simple learning-rate schedule and optional SARSA(0) variant. We will test both the standard ε-greedy exploration and improved UCB-ε exploration, along with a Double Q-learning variant to reduce overestimation bias.
3. **DDPG (comparison):** PyTorch or Stable-Baselines3 DDPG with identical reward/costs/splits for an apples-to-apples stability comparison.
4. **TD3 and PPO agents:** Implemented via Stable-Baselines3 using identical rewards, splits, and metrics to compare continuous-action baselines.
5. **Evaluation:** single script to compute Return, Sharpe, Max Drawdown, and Turnover; K-seed runs; walk-forward split (train/val/test).
6. **Reproducibility:** fixed seeds, config files, and saved metrics/plots.

Simplest first result

- **Setup:** 3–5 liquid assets, fixed Δ=5%, transaction cost = 10 bps, daily bars.

- **Algorithm: Tabular Q-learning** on the discrete env.
- **Baselines:** (i) equal-weight buy-and-hold, (ii) naïve momentum (top-k by 60-day return, monthly rebalance), (iii) stay-in-cash.
- **Goal:** beat equal-weight on validation with reasonable turnover and lower drawdown than pure momentum.

Stretch goals

- **Stability study:** Compare standard and improved Tabular Q-learning (Double Q, UCB-ε) against deep RL baselines (DQN, DDPG, TD3, PPO) across multiple random seeds, higher transaction costs (25–50 bps), and varying market regimes.
- **Exploration analysis:** study sensitivity of UCB-ε and Double Q-learning to transaction costs and reward scaling.
- **Actor-Critic** (if time): small value head for advantage estimates to stabilize updates.
- **Synthetic regimes:** a tiny regime-switching price simulator to test robustness.

What’s interesting / non-trivial

Most public repos jump straight to deep continuous control (DDPG/TD3/PPO). We will:

- **Introduce a discrete rebalancing MDP** so that **tabular Q-learning** is feasible and interpretable.
- Hold reward and cost functions constant and directly compare Tabular Q-learning, its improved variant (Double Q-learning), and deep RL baselines (DQN, DDPG, TD3, PPO) on stability (variance across seeds, cost sensitivity, and regime shifts).
- Provide a **reproducible evaluation harness** (metrics + seeds) others can reuse.

Acknowledgments

We’ll cite and credit **wassname/rl-portfolio-management** for the core environment and ideas; all new wrappers, agents, and evaluation code will be our own and clearly marked.