Reinforcement Learning for Dynamic Cryptocurrency Portfolio Management

An Empirical Deep Reinforcement Learning Framework for Weekly, Cost-Aware Crypto Allocation

Jose Márquez Jaramillo, Taylor Hawks

October 11, 2025

Johns Hopkins University

Motivation and Research Goals

- **Cryptocurrency markets** are highly volatile and exhibit strong regime shifts, making static allocation rules (e.g., equal or market-cap weights) often sub-optimal.
- **Reinforcement Learning (RL)** provides a natural framework for *adaptive portfolio rebalancing* by treating allocation as a sequential decision problem under uncertainty.
- Traditional optimization methods (mean–variance, risk parity) assume stationarity and ignore transaction costs; RL can model *dynamic, cost-aware* adjustments.

Research Goal:

- Develop and compare **deep RL agents** (LinUCB, DQN, A2C, REINFORCE+Baseline) for **long-only, weekly-rebalanced crypto portfolios**.
- Evaluate performance across three dimensions: **profitability, risk, and trading efficiency**.

Key references: [2–4].

Problem Statement and Contributions

Problem Statement

- Learn a **policy** $\pi_\theta(s_t)$ that outputs portfolio weights w_t maximizing long-run, risk-adjusted growth under realistic transaction costs.
- The environment provides daily market observations; the agent rebalances **weekly (every 7 days)** with proportional trading fees.

Objective

$$\pi^\star = \arg\max_{\pi} \mathbb{E}_\pi \left[\sum_{t=1}^{T} r_t \right],$$

$$r_t = \log(w_{t-1}^\top y_t) - c\|w_t - w_{t-1}\|_1.$$

Main Contributions

- A **cost- and risk-aware RL framework** for long-only cryptocurrency portfolio management using realistic Binance fees.
- A unified environment for **LinUCB, DQN, A2C, REINFORCE+Baseline** with shared state, action, and reward design.
- Comprehensive evaluation of RL agents and classical baselines (Equal-Weight, Market-Cap, Mean–Variance) across **profitability, risk, and efficiency**.

References: [1, 2, 4].

Formulation: Portfolio Management as an MDP

Formal Definition

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle, \quad \pi^\star = \arg\max_{\pi} \mathbb{E}_\pi \left[\sum_{t=1}^{T} r_t \right].$$

- **State** $\mathcal{S}$: market and portfolio context $(X_{t-L+1:t}, w_{t-1}, m_t)$.
- **Action** $\mathcal{A}$: portfolio weights w_t on the simplex $(w_{t,i} \geq 0, \sum_i w_{t,i} = 1)$.
- **Transition** P: exogenous market evolution; portfolio memory carried forward.
- **Reward** R: net log-return minus trading costs.
- **Discount** $\gamma = 1$: long-term wealth-growth objective.

Interpretation

- The RL agent observes daily features and prior weights, then rebalances **weekly**.
- Market prices drive transitions; the policy controls allocations only.
- The reward integrates both **return** and **cost sensitivity**.
- The optimal policy $\pi^\star$ maximizes expected cumulative wealth.

Based on frameworks by [2–4].

Design: Environment and State Representation

State at decision epoch t (weekly):

$$s_t = \left(X_{t-L+1:t},\, w_{t-1},\, m_t\right), \qquad X_{t-L+1:t} \in \mathbb{R}^{N \times F \times L}$$

- **Inputs X:** daily features per asset over a rolling window of L days: returns/OHLCV, realized volatility, and technical indicators.
- **Portfolio memory w_{t-1}:** previous weights to inform cost-aware rebalancing.
- **Universe mask m_t:** top-N assets by market cap, **reconstituted monthly**; ensures safe handling of listings/delistings.
- **Lookback L:** e.g., L daily observations; **actions are weekly** (every 7 days), features remain daily.
- **Normalization:** per-asset expanding mean/std (no look-ahead); deterministic preprocessing for reproducibility.

Framework precedents: [1, 3, 4].

Design: Actions and Transitions

Action (long-only weekly rebalance):

$$\hat{w}_t = f_\theta(s_t), \quad \tilde{w}_t = \mathrm{softmax}(\hat{w}_t) \odot m_t, \quad w_t = \frac{\tilde{w}_t}{\mathbf{1}^\top \tilde{w}_t} \in \Delta^N$$

Transition (advance by 7 calendar days):

$$s_{t+1} = \left(X_{t-L+1+7:\, t+7},\, w_t,\, m_{t+1}\right)$$

- **Constraints:** simplex (long-only), weights nonnegative and sum to 1; masking prevents allocation to inactive assets.
- **Costs:** applied at rebalance via $c\,\|w_t - w_{t-1}\|_1$ (Binance spot fees).
- **Optional:** *no-trade band* δ — if $\|w_t - w_{t-1}\|_1 < \delta$, skip trades to reduce micro-turnover; *cash asset* can be included as an additional dimension in X and m_t.

Design follows portfolio-RL practice in [1, 3, 4].

Design: Reward Function (Cost-Aware)

Base reward (weekly):

$$r_{t+1} = \log\left(w_t^\top y_{t+1}\right) \; - \; c \, \|w_t - w_{t-1}\|_1,$$

- y_{t+1}: vector of **7-day price relatives** for each asset (weekly rebalancing).
- c: **transaction cost per trade**, modeled from Binance spot fees: *0.10%* standard (non-VIP) or *0.075%* with BNB discount.

Source: binance.com/en/fee/schedule.
References: [2–4].

Optional risk-adjusted reward:

$$r'_{t+1} = r_{t+1} \; - \; \lambda \, \mathrm{Risk}(r_{t-H:t}),$$

- $\mathrm{Risk}(\cdot)$: rolling downside volatility, drawdown, or **CVaR**; λ tunes risk aversion.
- Encourages smoother policies under heavy tails while preserving growth incentives.

Design: Agent–Environment Interaction Loop

Weekly decision cycle (every 7 calendar days):

1. Observe s_t (features from last L days), current weights w_{t-1}.
2. **Choose action**:
   - *LinUCB*: pick arm/tilt a_t by UCB score; map to weight adjustment.
   - *DQN*: pick discrete rebalancing move $\arg\max_a Q_\psi(s_t, a)$ with ϵ-greedy.
   - *A2C / REINFORCE*: sample weights from softmax policy $\pi_\theta(\cdot|s_t)$.
3. Execute trade; update portfolio w_t; compute reward $r_{t+1} = \log(w_t^\top y_{t+1}) - c\|w_t - w_{t-1}\|_1$.
4. **Update**: LinUCB via ridge updates; DQN via TD loss with replay/target; A2C via advantage actor–critic; REINFORCE via Monte Carlo with baseline.

Aligned with portfolio-RL practice in [2–4].

Algorithm Possibilities: LinUCB, DQN, A2C, and REINFORCE (Baseline) (Choose Three)

Algorithm	Policy Type	Data Regime	Key Idea / Objective
LinUCB [5]	Contextual bandit (linear)	Online, per-decision	Choose arm/tilt a_t maximizing $\hat{\theta}^\top x_{t,a} + \alpha\sqrt{x_{t,a}^\top A^{-1} x_{t,a}}$; map selection to portfolio tilt on simplex (weekly).
DQN [7, 9, 10]	Value-based (discrete)	Off-policy w/ replay	Discretize rebalancing moves (e.g., $\pm 5\%$ shifts); learn $Q(s,a)$ with target network, Double/Dueling variants; argmax action per week.
A2C [8]	Actor–Critic (stochastic)	On-policy (batched)	Policy $\pi_\theta(a\|s)$ outputs weights via softmax; critic $V_\phi(s)$ for baseline; advantage updates with entropy bonus.
REINFORCE + baseline [11]	Policy gradient (stochastic/det.)	On-policy (episodic)	Directly optimize expected return; softmax to enforce simplex; subtract learned/value baseline to reduce variance.

Simplex & Costs: All methods enforce $w_i \geq 0$, $\sum_i w_i = 1$ (softmax or projection). Reward uses cost-aware log-return: $r_{t+1} = \log(w_t^\top y_{t+1}) - c\,\|w_t - w_{t-1}\|_1$ (weekly).

Experimental Setup: Data and Baselines

Data and Sampling

- **Assets:** top-10 cryptocurrencies by market capitalization (excluding stablecoins), reconstituted monthly.
- **Cadence:** daily OHLCV features; **weekly rebalancing** every 7 calendar days.
- **Period:** January 2019 – October 2025.
- **Splits:** Train (2019–2021), Validation (2022), Test (2023–2025).
- **Lookback:** $L = 30$ daily observations per asset; deterministic preprocessing, no look-ahead.
- **Transaction Costs:** $c = 0.0010$ (Binance spot fees).

Baselines (Long-Only, Weekly Rebalance)

- **Equal-Weight (EW):** $w_i = 1/N$; naive diversification benchmark.
- **Market-Cap Weight (CapW):** weights proportional to free-float market capitalization.
- **Mean–Variance (MVO):** classical Markowitz [6]; $\max_{w \geq 0, \mathbf{1}^\top w = 1} w^\top \hat{\mu} - \frac{\lambda}{2} w^\top \hat{\Sigma} w$, estimated from a 60-day rolling window.

Framework references: [1, 2, 4].

Evaluation: Metrics and Reporting Protocol

1. **Profitability**
   - **Cumulative wealth:** $W_T = \prod_{t=1}^{T}(w_t^\top y_t)$.
   - **Annualized return (CAGR):** $(W_T)^{1/(T/52)} - 1$ (weekly cadence).
   - **Excess return:** vs. Equal-Weight (EW) and Market-Cap (CapW) baselines.
2. **Risk**
   - **Volatility (annualized):** $\sigma_{\text{ann}} = \sigma_{\text{weekly}}\sqrt{52}$.
   - **Sharpe ratio:** $(\mu - r_f)/\sigma_{\text{ann}}$.
   - **Maximum Drawdown (MDD):** largest peak-to-trough portfolio loss.
3. **Efficiency**
   - **Turnover:** $\tau = \frac{1}{T}\sum_t \|w_t - w_{t-1}\|_1$.
   - **Fee drag:** realized cost $c\tau$ using Binance fee (0.10% per trade).
   - **Stability:** variance of weekly returns across rebalancing periods.

References: [2–4].

Reporting Protocol

- Model selection via validation Sharpe; final results on untouched test period.
- 3 random seeds per agent; report mean $\pm$ standard deviation.
- Robustness checks: alternate universes (top-5/top-15) and cost levels.

References i

[1] C. Betancourt et al. Deep reinforcement learning for portfolio management of markets with a dynamic number of assets. *Information Sciences*, 557:95–110, 2021. doi: 10.1016/j.ins.2020.12.020.

[2] T. Cui, S. Ding, H. Jin, and Y. Zhang. Portfolio constructions in cryptocurrency market: A cvar-based deep reinforcement learning approach. *Economic Modelling*, 119:106078, 2023. doi: 10.1016/j.econmod.2022.106078.

[3] Z. Jiang and J. Liang. Cryptocurrency portfolio management with deep reinforcement learning. In *2017 International Conference on Systems, Man, and Cybernetics (SMC)*, 2017. URL https://arxiv.org/abs/1612.01277. arXiv:1612.01277.

[4] Z. Jiang, D. Xu, and J. Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*, 2017. URL https://arxiv.org/abs/1706.10059.

[5] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 661–670, 2010. doi: 10.1145/1772690.1772758. URL https://arxiv.org/abs/1003.0146.

[6] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. doi: 10.1111/j.1540-6261.1952.tb01525.x.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

[8] V. Mnih, A. P. Badia, M. Mirza, et al. Asynchronous methods for deep reinforcement learning. In *ICML*, pages 1928–1937, 2016. URL https://arxiv.org/abs/1602.01783.

[9] H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.

[10] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas. Dueling network architectures for deep reinforcement learning. In *ICML*, pages 1995–2003, 2016.

[11] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992. doi: 10.1007/BF00992696.