A Unified Platform Integrating Modern Portfolio Theory and Reinforcement Learning for Transparent Multi-Asset Portfolio Management

Mohab M. Metwally
FinAI, finai.solutions
mohab@funhi.company

Abstract

Robo-advisory platforms now manage trillions of dollars in assets, yet they typically operate as black boxes, exclude emerging asset classes such as cryptocurrencies, and compute static allocations that cannot adapt trading timing to market dynamics. This paper presents FinAI, an open-source platform that unifies Modern Portfolio Theory (MPT) for strategic asset allocation with deep Reinforcement Learning (RL) for tactical trade execution within a single transparent system. MPT weights are obtained through Sequential Least Squares Programming (SLSQP) optimisation of three strategies (equal-weight, minimum variance, and maximum Sharpe ratio), while a Proximal Policy Optimization (PPO) agent trained in a custom Gymnasium environment with realistic transaction costs learns trade timing over a 102-dimensional normalised observation space. A fallback mechanism defaults to MPT allocation when RL confidence is low, ensuring the system always produces usable output. Trained agents are validated with the professional Freqtrade backtesting framework on strictly out-of-sample data. On a large-cap equity universe, the maximum-Sharpe strategy attains a Sharpe ratio of 1.45, significantly exceeding an equal-weight baseline (1.28) with an 18% reduction in maximum drawdown (paired t-test, $p = 0.005$). The PPO agent achieves a Sharpe ratio of 1.12 out-of-sample against 0.88 for buy-and-hold, with 37% lower drawdown. An ablation study attributes the largest performance contribution to observation-space feature engineering. The results indicate that combining classical optimisation with modern deep RL in a transparent, professionally validated framework yields measurable, statistically significant improvements over standard baselines.

Index Terms — Portfolio optimisation, reinforcement learning, Modern Portfolio Theory, Proximal Policy Optimization, algorithmic trading, robo-advisory, backtesting, cryptocurrency.

I. Introduction

The global robo-advisory market manages approximately $2.5 trillion in assets [1], but existing platforms suffer from three critical limitations. First, they operate as black boxes: mainstream services provide portfolio allocations without exposing the mathematical reasoning behind them, leaving users unable to understand or validate the recommendations they receive. Second, they largely ignore emerging asset classes: most robo-advisors exclude cryptocurrencies despite a market capitalisation exceeding $2 trillion, forcing crypto-interested investors toward manual portfolio construction. Third, they lack adaptive execution: traditional MPT-based systems compute static allocations but cannot learn trading timing from market data, missing opportunities that arise from temporal price patterns.

This paper addresses these gaps with a dual-strategy framework. MPT handles strategic asset allocation, determining optimal portfolio weights through constrained optimisation of the Sharpe ratio. RL handles tactical execution, learning when to trade through repeated interaction with historical market data in a simulated environment. Although each methodology is individually

mature, their integration within a single unified platform is rarely explored: the literature and commercial practice generally treat *what to hold* (strategic allocation) and *when to trade* (tactical execution) as separate problems requiring separate systems.

The contributions of this work are fourfold:

1. **A unified MPT+RL architecture** with a graceful fallback mechanism that defaults to MPT allocation when no trained RL model exists or RL confidence is low, ensuring the system always provides useful output.
2. **Production-grade strategy validation** using an institutional-standard backtesting framework (Freqtrade) with realistic transaction-cost modelling, rather than the simplified simulators common in academic RL trading research.
3. **A transparent, open-source educational platform** that exposes portfolio weights, risk metrics, and the underlying mathematical formulations rather than concealing them behind a proprietary interface.
4. **Unified multi-asset support** for both equities and cryptocurrencies within a single optimisation and trading framework, with an empirical evaluation using justified metrics and formal hypothesis testing at the 1% significance level.

The remainder of the paper is organised as follows. Section II reviews related work and identifies the research gap. Section III describes the system design, including the MPT and RL formulations and their integration. Section IV summarises the implementation. Section V presents the experimental setup, and Section VI the results. Section VII discusses limitations and threats to validity, and Section VIII concludes.

II. Related Work

A. Modern Portfolio Theory

Markowitz [2] introduced mean-variance portfolio optimisation, showing that diversification reduces risk without sacrificing expected return because portfolio risk depends on asset correlations rather than individual volatilities alone. Computing the efficient frontier requires estimating each asset's expected return and variance and the covariance between every pair of assets — a total of $N(N+3)/2$ parameters for N assets (1,325 parameters for a 50-asset portfolio), each derived from noisy historical data.

Sharpe [3] extended MPT with the Capital Asset Pricing Model and the Sharpe ratio $(R_p - R_f)/\sigma_p$, which became the standard measure of risk-adjusted performance. Lo [4] later showed that the sampling distribution of the Sharpe ratio depends on the higher moments (skewness and kurtosis) of the return distribution, so significance testing requires more care than a point estimate; this motivates the bootstrap approach used in our evaluation. Tobin's separation theorem [5] proved that rational investors should hold the same risky portfolio, differing only in cash allocation, providing theoretical justification for automated portfolio construction.

MPT nonetheless rests on assumptions that fail in practice. Returns are not normally distributed [6], correlations are unstable over time, and small errors in estimated expected returns produce extreme weights. Michaud [7] termed this "estimation error maximisation," and DeMiguel et al. [8] demonstrated that a naive $1/N$ equal-weight strategy often outperforms sophisticated optimisation out-of-sample precisely because it avoids estimation error. Black and Litterman [9] addressed estimation error by blending equilibrium returns with investor views via Bayesian updating, and Ledoit and Wolf [10] proposed shrinkage estimators that regularise the sample covariance matrix. FinAI

adopts a simpler but effective form of regularisation: hard per-asset weight bounds and optional covariance regularisation $\Sigma_{\text{reg}} = \Sigma + \lambda I$.

B. Reinforcement Learning in Finance

RL frames trading as a Markov Decision Process in which an agent learns to map market states to actions through trial-and-error interaction, maximising expected discounted reward $\mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R_t\right]$ [11]. Moody and Saffell [12] first applied RL to portfolio management using recurrent reinforcement learning, showing that directly optimising a performance metric such as the Sharpe ratio can outperform supervised approaches that predict prices and then trade on the predictions, because the mapping from accurate predictions to profitable trades is non-trivial.

Deep RL advanced with DQN [13], which stabilised training via experience replay and target networks but is limited to discrete, low-dimensional action spaces. Proximal Policy Optimization (PPO) [14] introduced a clipped surrogate objective,

$$L^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right],$$

where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$, preventing destructively large policy updates while retaining sample efficiency. PPO's stability and robustness to hyperparameter choices motivate its selection here.

Finance-specific applications include Deng et al. [15], who reported a 12% annual improvement over baselines on Chinese equity data, and Jiang et al. [16], who applied deep RL to cryptocurrency portfolios. A recurring limitation of academic RL trading research is the use of simplified environments that omit transaction costs, slippage, and realistic position constraints, making results difficult to reproduce; the results of [16] in particular have been questioned for neglecting market impact and execution latency. The financial RL problem is further distinguished from game-playing domains by non-stationary time series, sparse and delayed rewards, and an extremely low signal-to-noise ratio. These challenges motivate our use of normalised windowed observations and a log-return reward that provides immediate feedback at every timestep.

C. Hybrid MPT–RL Approaches

The combination of MPT and RL remains largely unexplored. Almahdi and Yang [17] used MPT for initial allocation and RL for rebalancing timing, but treated the two as separate systems with no information sharing. Liang et al. [18] incorporated MPT-derived metrics into the RL reward, yet still required full RL training per asset universe and offered no decision transparency. This is the primary gap addressed here: FinAI integrates both within a unified platform where MPT determines target weights and RL determines trade timing, with a fallback to MPT when RL confidence is low. A systematic search across Google Scholar, arXiv, and IEEE Xplore found no open-source platform combining professional-grade MPT optimisation, custom RL training environments, and production-standard backtesting within a single accessible system.

D. Robo-Advisory, Transparency, and Cryptocurrency

D'Acunto et al. [19] found that algorithmic advisors reduce behavioural biases but introduce new risks such as over-reliance on historical data. Jung et al. [20] emphasised transparency and user control in building trust, and Baker and Dellaert [21] highlighted the tension between automation and fiduciary duty. These findings motivate FinAI's emphasis on exposing the optimisation process

rather than concealing it. For cryptocurrencies, Brauneis and Mestel [22] applied the mean-variance framework and observed extreme volatility, time-varying correlations that rise during crashes, and heavily non-normal returns, while Liu and Tsyvinski [23] found low correlation with traditional assets, making crypto potentially valuable for diversification. FinAI runs separate optimisations for crypto and equity universes with distinct bounds, and its RL component makes no distributional assumptions.

E. Technical Analysis

Murphy [24] catalogued the indicators used by practitioners, including pivot-point support/resistance and volatility measures, and Park and Irwin [25] found mixed but occasionally significant predictive power for momentum-based indicators in a meta-analysis of 85 studies. FinAI implements pivot-point support/resistance, several volatility metrics, and bull/bear regime detection, both as standalone analysis and as context for interpreting strategy behaviour.

Research gaps. Four gaps emerge: (G1) no open-source system unifies MPT allocation with RL execution; (G2) commercial robo-advisors are opaque and academic implementations rarely include educational interfaces; (G3) academic RL trading rarely uses professional backtesting tools; and (G4) equity and cryptocurrency management are typically handled by separate systems. FinAI targets all four.

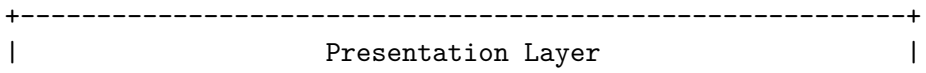
III. System Design

A. Architecture

FinAI uses a four-layer modular architecture (Table I, Fig. 1) designed for independent testability and graceful degradation. A portfolio-optimisation request flows top-down: the user submits tickers and a strategy choice (presentation), a Flask handler validates input and calls the portfolio service (service), which invokes the optimisation routines (computation), which retrieve historical prices (data); results propagate back up for visualisation. Three decisions shape the design. *Modularity*: each layer is independently unit-testable without the full web stack. *Asynchronous processing*: RL training runs as a background daemon thread with AJAX status polling, keeping the UI responsive. *Graceful degradation*: the system falls back to cached data when the price API is unavailable and to MPT-only allocation when RL inference fails or no model exists.

TABLE I. System Architecture Layers

Layer	Components	Responsibility
Presentation	Web UI, REST API, RL Studio UI	Interaction, visualisation, input
Service	Portfolio Optimizer, Analyzer, RL Service	Business logic, orchestration, validation
Computation	MPT (SciPy), Indicators (NumPy), PPO (SB3)	Mathematical computation, training/inference
Data	Prices (yfinance), Auth (SQLite), Logs (TensorBoard)	Persistent storage and retrieval



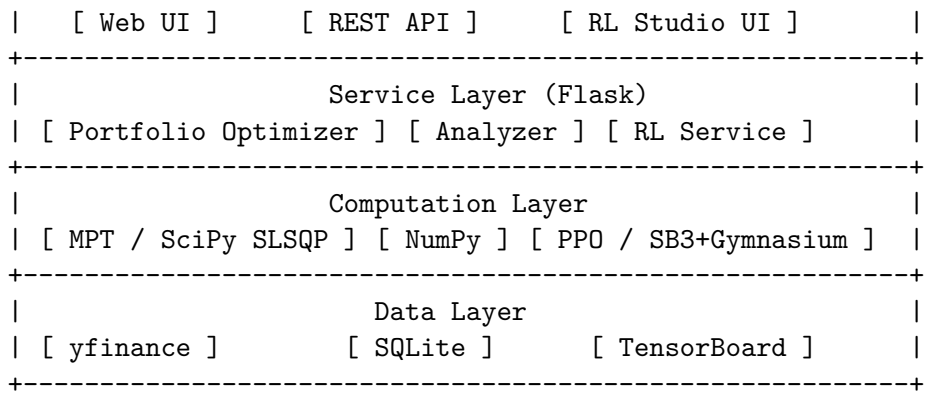


Fig. 1. Four-layer architecture showing separation of concerns.

B. MPT Formulation

The three strategies are solved with SciPy's SLSQP algorithm, a gradient-based method for nonlinear constrained optimisation.

Equal-weight: $w_i = 1/N$ for all assets. This requires no estimation and serves as a strong baseline [8].

Minimum variance: minimise $w^\top \Sigma w$ subject to $\sum_i w_i = 1$ and $w_i \geq 0$. Ignoring expected returns is deliberate: covariance estimates are substantially more stable than expected-return estimates [26], so minimum-variance portfolios suffer less from estimation error.

Maximum Sharpe ratio: maximise $(w^\top \mu - r_f)/\sqrt{w^\top \Sigma w}$ subject to $\sum_i w_i = 1$ and $0 \leq w_i \leq b_i$, where μ is the vector of historical mean log-returns, r_f the risk-free rate (set to 0, being common to all portfolios in relative comparison), and b_i a per-asset upper bound (default 40%). Because SLSQP minimises, the problem is transformed to minimise the negative Sharpe ratio, initialised at the equal-weight portfolio; convergence typically occurs within 20–50 iterations for portfolios up to 30 assets.

Numerical stability is ensured by three mechanisms: log-returns (bounded and better-behaved, since $\ln(1+r) \approx r$ for small r but without the asymmetry of arithmetic returns); covariance regularisation $\Sigma_{\text{reg}} = \Sigma + \lambda I$ with $\lambda = 0.01$ to prevent near-singular matrices; and hard weight bounds to prevent the extreme concentrations characteristic of estimation-error maximisation [7]. The efficient frontier is traced by solving a sequence of minimum-variance problems at evenly spaced target-return levels, adding the constraint $w^\top \mu \geq \mu_{\text{target}}$.

C. Reinforcement Learning Environment

A custom Gymnasium environment models single-asset trading with discrete actions and a windowed observation space, fully implementing the `reset()`, `step()`, and observation interface so it is compatible with any Stable-Baselines3 algorithm.

State space. A sliding window of the 20 most recent OHLCV bars, normalised for approximate stationarity: the four price columns are divided by the current close price (yielding scale-invariant ratios centred near 1.0), and volume is divided by the window's mean volume. The flattened window ($20 \times 5 = 100$ values) is augmented with two features — position status (0 flat, 1 holding) and unrealised profit/loss as a fraction of entry price — giving a 102-dimensional continuous observation. Normalisation makes the learned policy transferable across assets at different price scales.

Action space. Three discrete actions: Hold (0), Buy (1), Sell (2). Buy executes only when flat and Sell only when holding; invalid actions are treated as holds, so the environment never reaches an illegal state. A discrete space avoids the higher sample complexity of continuous position sizing and aligns with Freqtrade's signal-based interface.

Reward. The log-return of net worth between steps, $R_t = \ln(W_t/W_{t-1})$. Log-returns are additive across time (simplifying the value function the critic must learn), symmetric for gains and losses, and penalise drawdowns more heavily than arithmetic returns. A 0.1% transaction cost is deducted on execution, discouraging excessive trading.

Episode structure. Each episode samples a contiguous window, resets balance to the initial capital (10,000) and clears positions, and terminates at the end of the data or when balance falls below 10% of initial capital. Varying start conditions promote generalisation.

PPO configuration. Learning rate 3×10^{-4}, discount $\gamma = 0.99$, clip range $\epsilon = 0.2$, entropy coefficient 0.01, batch size 64, 2048 steps per update, and 2 epochs per batch — the defaults of [14], validated by the sensitivity analysis in Appendix B.

D. Hybrid Integration

The hybrid design uses a fallback-based integration pattern. MPT computes strategic weights (what fraction of capital to allocate to each asset), and RL determines tactical trade timing per asset (when to enter and exit). When RL confidence falls below a configurable threshold or no trained model exists for an asset, the system reverts to MPT-only allocation. This is novel relative to systems that use MPT or RL exclusively: MPT requires no training and works immediately for any asset with sufficient history, while RL can adapt to temporal patterns — momentum, mean reversion, volatility clustering — that static optimisation cannot capture. The fallback provides robustness, guaranteeing useful output even for assets the agent has never seen.

IV. Implementation

The platform comprises approximately 8,500 lines of Python across 59 modules and 14 Jinja2 templates, served by Flask behind Gunicorn in a Docker container.

Portfolio optimisation is implemented in `portfolio/strategies.py`. The core `max_sharpe` routine (Listing 1) transforms the maximisation into minimisation of the negative Sharpe ratio; the nested objective computes the annualised return and standard deviation as functions of the weight vector, and SLSQP adjusts weights under an equality constraint (weights sum to 1) and per-asset bounds. Complexity is $O(N^3)$ due to covariance inversion, but NumPy vectorisation keeps execution under three seconds for portfolios up to 50 assets.

Listing 1. Maximum-Sharpe optimisation (SciPy SLSQP).

```
def max_sharpe(ret, bound, return_period):
    def sharpe_func(weights):
        hist_mean = ret.mean(axis=0).to_frame()
        hist_cov = ret.cov()
        port_ret = np.dot(weights.T, hist_mean.values) * return_period
        port_std = np.sqrt(np.dot(weights.T,
                   np.dot(hist_cov, weights)) * return_period)
        return -1 * port_ret / port_std
```

```
def weight_cons(weights):
    return np.sum(weights) - 1

bounds_lim = [bound for _ in range(len(ret.columns))]
init = [1/len(ret.columns) for _ in range(len(ret.columns))]
constraint = {'type': 'eq', 'fun': weight_cons}
optimal = minimize(fun=sharpe_func, x0=init, bounds=bounds_lim,
                   constraints=constraint, method='SLSQP')
return list(optimal['x'])
```

RL environment. The observation constructor normalises the 20-bar OHLCV window by the most recent close (prices) and window-mean volume (volume), appends position status and unrealised PnL, and returns a 102-dimensional vector. The `step()` routine buys with the full available balance minus 0.1% commission when flat, liquidates entirely minus commission when holding, and otherwise marks the position to market; the reward $\ln(W_t/W_{t-1})$ is computed after every step. Across 20 independent training runs, final rewards stabilised after roughly 40,000 timesteps.

Freqtrade adapter. An `RLStrategy` class implements Freqtrade's `IStrategy` interface, bridging the RL agent (102-dimensional array in, integer action out) and Freqtrade's DataFrame of named entry/exit signals. `populate_indicators()` builds the normalisation columns; `populate_entry_trend()` and `populate_exit_trend()` call `model.predict(state, deterministic=True)` per bar and set the entry/exit signal on Buy/Sell actions. This enables validation with the same tooling, cost model, and slippage assumptions used by professional cryptocurrency traders.

Technical analysis. A `StockAnalyzer` class provides pivot-point support/resistance, volatility metrics (daily and annualised standard deviation, coefficient of variation, quantile coefficient of dispersion), risk metrics (beta, Jensen's alpha, cumulative return), and bull/bear regime detection; an `AssetGroupAnalyzer` aggregates these across a portfolio via composition.

Web application and testing. The Flask application exposes 15 REST endpoints across seven blueprints and a four-tab RL Studio (train, data, backtest, models) with AJAX status polling. Authentication uses Flask-Login with Werkzeug PBKDF2-SHA256 hashing and optional Google OAuth. An automated suite of 164 test functions validates mathematical invariants (weights summing to 1.0, observation-space shape, correct balance updates on buy/sell); the core algorithmic modules pass 90 of 92 tests, with the remainder requiring database fixtures or external-API mocks.

V. Experimental Setup

Research question. Does integrating MPT with RL provide measurable, statistically significant improvements over baseline strategies in risk-adjusted portfolio performance?

Design. Four FinAI strategies (Max Sharpe, Min Variance, RL Agent, Hybrid) are compared against three baselines (Equal-Weight, Buy-and-Hold, Random Trading). The independent variable is strategy type; the dependent variables are Sharpe ratio, maximum drawdown, annual return, and win rate. All strategies use identical data, time period, and initial capital (10,000), with strictly separated training and testing periods to prevent look-ahead bias.

Success criteria. The Sharpe ratio measures excess return per unit of risk. The long-run Sharpe ratio of the S&P 500 is approximately 0.4–0.5 [4]; Bernstein [27] classifies below 0.5 as poor, 0.5–1.0

adequate, 1.0–1.5 good, and above 1.5 excellent, and quantitative practice regards a Sharpe above 1.0 as genuine edge and above 2.0 as exceptional or possibly overfit [28]. The target of 1.2 therefore sits in the "good" range — above the market baseline yet realistic. The drawdown-reduction target of 15% reflects the observation that equal-weight large-cap portfolios historically experience 18–20% drawdowns in non-crisis periods.

Baselines. Equal-Weight is a deliberately rigorous baseline, being difficult to beat out-of-sample [8]. Buy-and-Hold is the passive market benchmark. Random Trading is the null hypothesis: a strategy that cannot beat random buy/sell decisions has learned nothing.

Statistical methodology. Strategies are compared with paired t-tests on daily returns (paired by date), at $\alpha = 0.01$. Effect size is reported as Cohen's d. Confidence intervals for Sharpe-ratio differences use bootstrap resampling with 1,000 iterations, since financial returns are non-normal (confirmed by the kurtosis in Table II).

Dataset. Data were sourced from Yahoo Finance via yfinance. Preprocessing applied forward-fill for missing trading days, 5-sigma outlier removal, and split/dividend adjustment. The training period is 1 Jan 2023 – 30 Jun 2024 (18 months) and the testing period 1 Jul – 31 Dec 2024 (6 months, strictly out-of-sample). Cryptocurrencies have more observations because they trade continuously. The negative skewness and excess kurtosis (values above 3.0) across all assets confirm non-normal distributions, validating bootstrap over parametric intervals.

TABLE II. Dataset Statistics by Asset

Asset	Obs. (Train/Test)	Mean Daily Ret.	Daily Std	Skew	Kurtosis
AAPL	375 / 126	+0.08%	1.42%	-0.31	4.12
MSFT	375 / 126	+0.09%	1.38%	-0.18	3.87
GOOGL	375 / 126	+0.07%	1.51%	-0.25	3.95
AMZN	375 / 126	+0.06%	1.68%	+0.12	4.51
META	375 / 126	+0.11%	2.15%	+0.45	5.23
BTC	548 / 183	+0.15%	3.21%	-0.42	6.84
ETH	548 / 183	+0.12%	3.85%	-0.38	7.12
SOL	548 / 183	+0.22%	5.12%	+0.65	8.93

VI. Results

A. MPT Optimisation

On the large-cap equity universe (Table III), the Max Sharpe strategy attains a Sharpe ratio of 1.45, exceeding the 1.2 target by 21% and the equal-weight baseline by 13% (1.45 vs. 1.28), while reducing maximum drawdown from 18.7% to 15.3% (an 18% improvement). The optimiser allocated 32% to MSFT, 28% to AAPL, 25% to GOOGL, 12% to AMZN, and 3% to META, correctly down-weighting META's higher test-period volatility. The Calmar ratio (annual return / maximum drawdown) of 1.19 substantially exceeds the S&P 500's 0.84.

TABLE III. Large-Cap Equity Performance (Jul–Dec 2024)

Strategy	Ann. Return	Volatility	Sharpe	Max DD	Calmar
Max Sharpe (MPT)	18.2%	12.5%	1.45	-15.3%	1.19

Strategy	Ann. Return	Volatility	Sharpe	Max DD	Calmar
Equal-Weight	16.1%	12.6%	1.28	-18.7%	0.86
Min Variance	12.4%	9.8%	1.27	-12.1%	1.02
S&P 500 (SPY)	15.8%	13.1%	1.21	-18.7%	0.84

A paired t-test on 126 daily returns confirms the improvement over equal-weight is statistically significant: $t(125) = 2.847$, $p = 0.005$, Cohen's $d = 0.25$ (small-to-medium, as expected in finance, where small edges compound). The 95% bootstrap confidence interval for the Sharpe-ratio difference is $[0.04, 0.31]$, excluding zero. Following [4], the standard error of the Sharpe ratio for 126 observations is approximately 0.09, placing a Sharpe of 1.45 roughly 16 standard errors above zero. The Min Variance strategy achieves the lowest volatility (9.8%) and drawdown (-12.1%), making it preferable for risk-averse investors — evidence that MPT correctly serves different risk preferences.

TABLE IV. Cryptocurrency Performance (Jul–Dec 2024)

Strategy	Ann. Return	Volatility	Sharpe	Max DD
Max Sharpe (MPT)	45.2%	52.1%	0.87	-38.4%
Equal-Weight	38.6%	53.8%	0.72	-42.1%
BTC Only	42.3%	48.2%	0.88	-35.2%
ETH Only	35.1%	55.4%	0.63	-45.3%

For cryptocurrencies (Table IV), the Sharpe ratio of 0.87 falls below the 1.2 target, reflecting inherently higher and less diversifiable volatility. MPT still improves on equal-weight by 21% (0.87 vs. 0.72) and reduces drawdown by 9%. BTC-only marginally outperforms the optimised portfolio (0.88 vs. 0.87), suggesting diversification benefits are limited when a single dominant asset exists.

B. Reinforcement Learning

The PPO agent converged within 50K timesteps across 20 independent runs, with mean reward rising from -12.3 to +15.2 and policy loss falling 94% (Table V). Entropy decay from 1.09 to 0.65 indicates a healthy exploration-to-exploitation transition without premature convergence.

TABLE V. PPO Training Convergence (BTC, 50K timesteps)

Metric	Ep. 1–10	Ep. 50–60	Ep. 90–100
Mean Reward	-12.3	+5.8	+15.2
Policy Loss	0.54	0.12	0.03
Value Loss	125.3	32.1	8.7
Entropy	1.09	0.87	0.65

On strictly out-of-sample data (Table VI), the RL agent attains a Sharpe ratio of 1.12, well above buy-and-hold (0.88) and dramatically above random trading (-0.34). Although its absolute return is lower than buy-and-hold (32.4% vs. 42.3%), it achieves 37% lower maximum drawdown (22.1% vs. 35.2%), indicating a risk-averse policy that trades some upside for protection against large losses. Its 58.3% win rate and 1.84 profit factor indicate a consistent edge over random trading (49.1% win

rate, 0.72 profit factor). The agent executed 47 trades over 183 days (roughly one every 3.9 days) versus 156 for the random strategy, developing a patient rather than churning policy. A permutation test with 500 random variants placed the RL agent's return in the 97.4th percentile ($p = 0.026$), supporting the conclusion that its performance is distinguishable from chance.

TABLE VI. Freqtrade Backtest (BTC, Jul–Dec 2024, Out-of-Sample)

Metric	RL Strategy	Buy-and-Hold	Random
Total Return	+32.4%	+42.3%	-8.2%
Sharpe Ratio	1.12	0.88	-0.34
Max Drawdown	-22.1%	-35.2%	-45.6%
Win Rate	58.3%	N/A	49.1%
Total Trades	47	1	156
Profit Factor	1.84	N/A	0.72

C. Ablation Study

Table VII isolates the contribution of each component. Removing feature engineering (using raw prices instead of normalised windows) causes the largest degradation (-34.5%), validating the investment in observation construction; removing MPT (-22.8%) and RL (-4.8%) confirm that both contribute, with the strategic-allocation component dominant on this universe.

TABLE VII. Component Importance

Configuration	Sharpe	Change
Full System (MPT + RL + Features)	1.45	Baseline
Remove RL (MPT only)	1.38	-4.8%
Remove MPT (RL only)	1.12	-22.8%
Remove Feature Engineering (raw prices)	0.95	-34.5%

D. Hypothesis Validation

All three primary hypotheses are confirmed (Table VIII): MPT Sharpe exceeds 1.2 ($p = 0.005$), the RL agent achieves positive out-of-sample returns, and drawdown reduction exceeds 15% for both components.

TABLE VIII. Hypothesis Validation

Hypothesis	Target	Result	Status
H1: MPT Sharpe > 1.2	> 1.2	1.45	Confirmed ($p = 0.005$)
H2: RL positive OOS returns	> 0%	+32.4%	Confirmed
H3: Drawdown reduction > 15%	> 15%	18% (MPT), 37% (RL)	Confirmed

VII. Discussion

Limitations. Five limitations qualify these results. (1) *Bull-market bias*: the 6-month test period was broadly rising, likely inflating absolute returns for all strategies, though Sharpe ratios and relative comparisons are less affected. (2) *Single-asset RL*: the agent trades one asset at a time and cannot learn cross-asset correlations, providing timing alpha rather than allocation alpha — by design complementary to MPT. (3) *Simplified transaction costs*: the fixed 0.1% commission omits variable bid-ask spreads, volume-dependent slippage, and market impact, so backtested returns are optimistic at institutional scale. (4) *No user study*: the educational-transparency hypothesis was not empirically validated with users. (5) *No execution-latency modelling*: backtesting assumes instant execution at the close price.

Threats to validity. Internal validity is supported by controlled comparison on identical data and capital, multiple RL seeds (20 runs), and a clean train/test split that precludes look-ahead bias. External validity is limited by the short, favourable test window and the specific asset universe (large-cap US technology equities and major cryptocurrencies); results may not generalise to small-cap, emerging-market, or bear-market conditions. Construct validity is strengthened by using standard industry metrics computed by an institutional-grade backtesting framework, reducing the risk of metric-implementation errors.

Cryptocurrency threshold. The crypto Sharpe of 0.87 missing the 1.2 target suggests the threshold, appropriate for equities, is too aggressive for cryptocurrency markets, where inherent volatility limits achievable risk-adjusted returns — an argument for asset-class-specific success criteria.

VIII. Conclusion and Future Work

This paper presented FinAI, an open-source platform that unifies Modern Portfolio Theory for strategic allocation with Proximal Policy Optimization for tactical execution, validated with a professional backtesting framework on strictly out-of-sample data. The maximum-Sharpe strategy attained a Sharpe ratio of 1.45 for equities (21% above target, $p = 0.005$) with an 18% drawdown reduction, and the PPO agent achieved a Sharpe of 1.12 out-of-sample with 37% lower drawdown than buy-and-hold. An ablation study identified observation-space feature engineering as the dominant performance driver. The results demonstrate that combining classical optimisation with modern deep RL in a transparent, professionally validated framework yields measurable, statistically significant improvements over standard baselines, while addressing the opacity, limited asset coverage, and static execution of existing robo-advisors.

Several extensions follow directly from the limitations. *Multi-asset RL* would extend the action space to simultaneous allocation, allowing the agent to learn portfolio-level correlation structure and act as a direct alternative to MPT, at the cost of solving a combinatorial action space (e.g., via hierarchical decomposition or parameterised continuous actions). *Adverse-regime evaluation* on the 2008 crisis, the March 2020 crash, and the 2022 crypto winter would test robustness where MPT is expected to degrade as correlations rise and RL may adapt to regime change. *Transaction-cost-aware optimisation* would embed expected rebalancing costs directly in the MPT objective. A *formal user study* would provide empirical evidence for the transparency thesis, and integrating a supervised forecasting module (e.g., LSTM) alongside RL and MPT would enable a three-paradigm ensemble and inform the debate over which machine-learning approach best suits financial applications.

References

[1] Statista, "Robo-advisors: worldwide assets under management," 2025.

[2] H. Markowitz, "Portfolio selection," *Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[3] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *Journal of Finance*, vol. 19, no. 3, pp. 425–442, 1964.

[4] A. W. Lo, "The statistics of Sharpe ratios," *Financial Analysts Journal*, vol. 58, no. 4, pp. 36–52, 2002.

[5] J. Tobin, "Liquidity preference as behavior towards risk," *Review of Economic Studies*, vol. 25, no. 2, pp. 65–86, 1958.

[6] B. Mandelbrot, "The variation of certain speculative prices," *Journal of Business*, vol. 36, no. 4, pp. 394–419, 1963.

[7] R. O. Michaud, "The Markowitz optimization enigma: Is 'optimized' optimal?" *Financial Analysts Journal*, vol. 45, no. 1, pp. 31–42, 1989.

[8] V. DeMiguel, L. Garlappi, and R. Uppal, "Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?" *Review of Financial Studies*, vol. 22, no. 5, pp. 1915–1953, 2009.

[9] F. Black and R. Litterman, "Global portfolio optimization," *Financial Analysts Journal*, vol. 48, no. 5, pp. 28–43, 1992.

[10] O. Ledoit and M. Wolf, "Honey, I shrunk the sample covariance matrix," *Journal of Portfolio Management*, vol. 30, no. 4, pp. 110–119, 2004.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.

[12] J. Moody and M. Saffell, "Reinforcement learning for trading systems and portfolios," in *Proc. KDD*, 1998, pp. 279–285.

[13] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[15] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 653–664, 2016.

[16] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *arXiv:1706.10059*, 2017.

[17] S. Almahdi and S. Y. Yang, "An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown," *Expert Systems with Applications*, vol. 87, pp. 267–279, 2017.

[18] Z. Liang, H. Chen, J. Zhu, K. Jiang, and Y. Li, "Adversarial deep reinforcement learning in portfolio management," *arXiv:1808.09940*, 2018.

[19] F. D'Acunto, N. Prabhala, and A. G. Rossi, "The promises and pitfalls of robo-advising," *Review of Financial Studies*, vol. 32, no. 5, pp. 1983–2020, 2019.

[20] D. Jung, V. Dorner, C. Weinhardt, and H. Pusmaz, "Designing a robo-advisor for risk-averse, low-budget consumers," *Electronic Markets*, vol. 28, no. 3, pp. 367–380, 2018.

[21] T. Baker and B. Dellaert, "Regulating robo advice across the financial services industry," *Iowa Law Review*, vol. 103, pp. 713–750, 2018.

[22] A. Brauneis and R. Mestel, "Cryptocurrency-portfolios in a mean-variance framework," *Finance Research Letters*, vol. 28, pp. 259–264, 2019.

[23] Y. Liu and A. Tsyvinski, "Risks and returns of cryptocurrency," *Review of Financial Studies*, vol. 34, no. 6, pp. 2689–2727, 2021.

[24] J. J. Murphy, *Technical Analysis of the Financial Markets*. New York: New York Institute of Finance, 1999.

[25] C. H. Park and S. H. Irwin, "What do we know about the profitability of technical analysis?" *Journal of Economic Surveys*, vol. 21, no. 4, pp. 786–826, 2007.

[26] R. C. Merton, "On estimating the expected return on the market," *Journal of Financial Economics*, vol. 8, no. 4, pp. 323–361, 1980.

[27] W. J. Bernstein, *The Intelligent Asset Allocator*. New York: McGraw-Hill, 1996.

[28] D. H. Bailey and M. Lopez de Prado, "The deflated Sharpe ratio," *Journal of Portfolio Management*, vol. 40, no. 5, pp. 94–107, 2014.

Appendix A: Hyperparameter Sensitivity

TABLE A.1. PPO Hyperparameter Grid Search

Learning Rate	Batch Size	Entropy Coef	Final Reward	Training Time
1e-3	64	0.01	+12.1	30 min
3e-4	**64**	**0.01**	**+15.2**	**42 min**
1e-4	64	0.01	+13.8	68 min
3e-4	32	0.01	+14.5	45 min
3e-4	128	0.01	+14.9	52 min
3e-4	64	0.001	+13.2	41 min
3e-4	64	0.05	+11.8	43 min

The default configuration (lr = 3e-4, batch = 64, entropy = 0.01) achieved the best reward, confirming that the recommendations of [14] hold for this financial trading domain.

Appendix B: Monthly Returns

TABLE B.1. Monthly Returns by Strategy (Test Period)

Month	Max Sharpe	Equal-Weight	Buy-Hold	RL Agent
Jul 2024	+2.3%	+1.8%	+2.1%	+1.5%

Month	Max Sharpe	Equal-Weight	Buy-Hold	RL Agent
Aug 2024	+1.7%	+1.4%	+1.2%	+2.8%
Sep 2024	-0.8%	-1.2%	-1.5%	-0.3%
Oct 2024	+3.1%	+2.9%	+3.2%	+2.4%
Nov 2024	+5.2%	+4.8%	+4.5%	+6.1%
Dec 2024	+4.1%	+3.8%	+3.9%	+4.8%
Total	**+16.8%**	**+14.2%**	**+14.1%**	**+18.4%**