Portfolio Allocation with Neural Policies: DRL vs Mirrored Evolution Strategies

Teo Pazera
Department of Computer science
Radboud University
Nijmegen
teo.pazera@ru.nl
s1159164

Abstract

We compare two ways to train the same neural portfolio policy: differentiable policy optimization (DRL) and mirrored evolution strategies (ES). The agent reallocates capital across eight liquid ETFs plus cash under transaction costs, using an objective that interpolates between annualized Sharpe ratio and annualized mean return. To reduce unstable high turnover behavior, we restrict rebalancing to every 30 trading days and train with a 90-day holding-period return signal. On a single train–test split, both optimizers can overfit in sample, yet several Sharpe-heavy and mixed settings still outperform benchmark of SP500 index (SPY) on the out of sample test data. DRL changes smoothly as the objective weight varies, while ES is most competitive in the mixed regime but less stable at the pure Sharpe extreme. The code repository used to reproduce and inspect the experimental results is publicly available at `https://github.com/teoPazera/Portfolio_agent`.

1 Introduction

Portfolio allocation is a sequential decision problem meaning that at each decision time the agent must select portfolio weights to balance expected growth against risk, while paying trading frictions. Classical portfolio theory formalizes the risk return trade-off via mean variance optimization (1) and popular risk-adjusted performance measures such as the Sharpe ratio (2). Modern work often casts allocation as reinforcement learning (RL), where a policy maps market state to weights and is trained to maximize a return-based objective. For example Jiang et al. propose deep RL architectures for portfolio management with explicit portfolio-weight outputs and transaction costs (3). In parallel, Evolution Strategies (ES) have been shown to optimize neural policies as a scalable, gradient-free alternative to RL-style policy gradients (4), and ES-based portfolio agents have been explored in finance settings as well (5). Motivated by these threads, this report compares DRL (gradient-based optimization of a differentiable episode objective) against mirrored ES, while holding the environment, policy network, and objective fixed.

2 Data, assets, and state representation

We allocate among eight ETFs that represent distinct asset classes and regions: VNQ (U.S. real estate / REITs), IWM (U.S. small-cap equities), GLD (gold), SPY (U.S. large-cap equities / S&P 500), VGK (developed Europe equities), EFA (developed ex-U.S. equities), AGG (U.S. investment-grade aggregate bonds), and EEM (emerging market equities). In addition, we include a cash asset in the portfolio to enable risk-off allocations without shorting; cash is modeled as an additional weight with zero return.

Preprint. Under review.

From adjusted close prices we compute daily simple returns for the traded ETFs (used for portfolio P&L), and daily log returns (used for features). Let N denote the number of traded ETFs (here $N = 8$). We normalize log returns (per asset) to obtain a matrix of normalized log returns $\tilde{\ell} \in \mathbb{R}^{T\times N}$; the normalization is applied before constructing rolling statistics. For each time index t where a full history window is available, we build the feature vector exactly as in our implementation:

$$\phi_t = \Big[\mathrm{vec}\big(\tilde{\ell}_{t-W+1:t}\big),\ \mu_t,\ \sigma_t\Big],$$

where W is the window length, $\tilde{\ell}_{t-W+1:t} \in \mathbb{R}^{W\times N}$ is the last W days of normalized log returns, $\mathrm{vec}(\cdot)$ denotes flattening into a length WN vector, and $\mu_t, \sigma_t \in \mathbb{R}^N$ are the rolling mean and rolling standard deviation of normalized log returns over the same W-day window (computed independently per ETF). Therefore the feature dimension is

$$F = WN + N + N = N(W + 2).$$

With $N = 8$ and our chosen $W = 15$, this yields $F = 8(15 + 2) = 136$ features per decision time. The resulting dataset has $T_{\text{eff}} = T - W + 1$ effective time steps because the first $W - 1$ days are used to warm up the window.

Although the portfolio is rebalanced only every $k = 30$ days, the state uses only the most recent $W = 15$ trading days (plus rolling mean and standard deviation) when making each decision. This is intentional: the action cadence controls *how often* the portfolio can change, while the lookback window controls *how much recent information* the policy conditions on. In preliminary experiments we tested shorter-horizon, more reactive setups and observed unstable high-turnover behavior and severe in-sample overfitting consistent with "day-trading" dynamics. Restricting the information set to the most recent 15 days reduces noise and limits the policy's ability to exploit very local training-sample idiosyncrasies, while the 30-day rebalancing rule further prevents frequent weight-chasing. Empirically this combination preserved strong out-of-sample performance despite large apparent overfitting in the training wealth curves.

The policy input at decision time t is formed by concatenating the feature vector with the previous portfolio weights:

$$x_t = [\phi_t,\ w_{t-1}] \in \mathbb{R}^{145},$$

where $w_{t-1} \in \mathbb{R}^9$ contains the previous weights over the eight ETFs plus cash. The environment uses the corresponding daily simple returns vector $r_t \in \mathbb{R}^9$ (ETF returns with an appended zero cash return) to evaluate portfolio performance and transaction-cost-adjusted rewards.

3 Methods: policy, objective, and optimization

Our setup is fully differentiable in JAX and trains the same policy network with two different optimizers (DRL vs. mirrored ES). The key detail is that in the sweep experiments we trained with a *horizon-based* objective: each reward sample corresponds to a 90-day holding-period return, not a daily return.

The policy is a small MLP that maps the state $x_t \in \mathbb{R}^{145}$ to portfolio weights $w_t \in \mathbb{R}^9$. The architecture in text form is: *145 inputs (136 features + 9 previous weights) → 64 → 64 → 9 outputs*. Hidden layers use GELU activations. The final layer outputs logits $z_t \in \mathbb{R}^9$, which are converted to a long-only fully-invested portfolio using a softmax.

$$w_t = \mathrm{softmax}(z_t), \tag{1}$$

so that $\sum_{i=1}^{9} w_{t,i} = 1$ and $w_{t,i} \geq 0$. This network has roughly 14,000 trainable parameters.

Let $r_t \in \mathbb{R}^9$ denote the vector of *daily simple returns* at day t for the 8 ETFs plus cash, where the cash return is 0. For a decision made at time index t, we define the H-day compounded simple return of each risky ETF as

$$R_{t,i}^{(H)} = \prod_{h=0}^{H-1} \big(1 + r_{t+1+h,i}\big) - 1, \tag{2}$$

where i indexes assets and $H = 90$ in the sweep. (For cash, $R_{t,\text{cash}}^{(H)} = 0$ under the constant-zero-return cash model.) Given portfolio weights w_t, the portfolio H-day simple return is

$$R_t^{(p,H)} = w_t^\top R_t^{(H)}, \tag{3}$$

where $R_t^{(H)} \in \mathbb{R}^9$ stacks the H-day compounded returns for all assets (including cash).

We penalize turnover using an L1 transaction cost with cost rate $c = 10^{-4}$:

$$\mathrm{cost}_t = c \, \|w_t - w_{t-1}\|_1, \tag{4}$$

where $\|\cdot\|_1$ sums absolute weight changes across assets. In horizon mode, the per-decision reward optimized during training is

$$\tilde{r}_t = \log\bigl(1 + R_t^{(p,H)}\bigr) - \mathrm{cost}_t. \tag{5}$$

This is the critical distinction: $\tilde{r}_t$ is not a daily log-return, it is a *90-day* log-growth reward computed from compounded future returns after the decision time (which is valid in RL because rewards are realized after acting, but it must not cross the train/test boundary).

The policy is only allowed to change weights every $k = 30$ days; between decision points the portfolio is held constant. We tested the same framework without these constraints (more frequent decisions and/or short-horizon objectives) and observed severe train-set overfitting and "day-trading" behavior, visible as unrealistically large training wealth multiples (e.g., $\sim$40×). Using 30-day decision intervals together with a 90-day horizon reward was the practical mitigation we found to reduce high-turnover strategies and improve stability.

Over an episode we collect the sequence of per-decision rewards $\tilde{r}_{1:T_d}$, where T_d counts decision points (not days). We then compute annualized mean and annualized Sharpe using an annualization factor appropriate for H-day samples:

$$\mu_{\mathrm{ann}} = \frac{252}{H}\,\mathrm{mean}(\tilde{r}), \qquad \mathrm{Sharpe}_{\mathrm{ann}} = \sqrt{\frac{252}{H}}\,\frac{\mathrm{mean}(\tilde{r})}{\mathrm{std}(\tilde{r}) + \epsilon}, \tag{6}$$

where ϵ is a small stabilizer to avoid division by zero. The score optimized by both DRL and ES is a convex combination of these terms with an optional prior regularizer:

$$\mathrm{score} = w_{\mathrm{sharpe}} \cdot \mathrm{Sharpe}_{\mathrm{ann}} + w_{\mathrm{return}} \cdot \mu_{\mathrm{ann}} - \lambda_{\mathrm{prior}} \cdot \mathrm{prior\_penalty}, \qquad w_{\mathrm{return}} = 1 - w_{\mathrm{sharpe}}, \tag{7}$$

and the loss is $\mathcal{L}(\theta) = -\mathrm{score}$. We sweep $w_{\mathrm{sharpe}} \in \{1.0, 0.8, 0.6, 0.4, 0.2, 0.0\}$.

Therefore, the plotted "annualized Sharpe" and "total return" are *daily-eval metrics*, while the training loss uses *horizon-mode* samples with annualization factor $252/H$. They are related but not numerically identical, so the report must state which one is being shown in each figure/table.

DRL updates parameters using gradients of $\mathcal{L}(\theta)$ through the differentiable rollout. Mirrored ES estimates a gradient direction using paired perturbations $\pm\sigma\epsilon_i$ and updates parameters without backpropagation. Hyperparameters were chosen by grid search using *training performance only* (DRL: learning rate; ES: learning rate and σ), to avoid test-set leakage. We optimized both for 200 iterations saving the best solution on the train set using $lr = 0.1$ for DRL and for ES we choose $lr = 0.1, \sigma = 0.05$ with population of 64 individuals.

4 Experiments and results

We ran a sweep over objective weights as mentioned above, training one DRL policy and one ES policy per setting.

Figure 1 shows that DRL behaves relatively consistently as the we increase w_{sharpe} we improve test Sharpe and, in our runs, also improves test total return, suggesting that the Sharpe component discourages overly volatile and fragile behavior and improves generalization on the test sample. ES is highly competitive across the mixed regime, it often achieves higher test Sharpe than DRL for intermediate weights and reaches its best overall trade-off around $w_{\mathrm{sharpe}} = 0.6$, where both risk-adjusted performance and total return are strong. However, ES is also noticeably more sensitive to the objective, at the ($w_{\mathrm{sharpe}} = 1$), ES performance drops sharply in both Sharpe and return under daily evaluation, indicating reduced robustness when optimizing the ratio-based Sharpe objective in this setting.

A recurring pattern across the sweep is the presence of severe in-sample overfitting. Many runs generate unrealistically large training wealth multiples (e.g., DRL reaching $\sim$40× and ES reaching $\sim$30× in some configurations), while the test curves remain much more conservative. This gap suggests that the policy can exploit training-specific dynamics even with transaction costs, and that

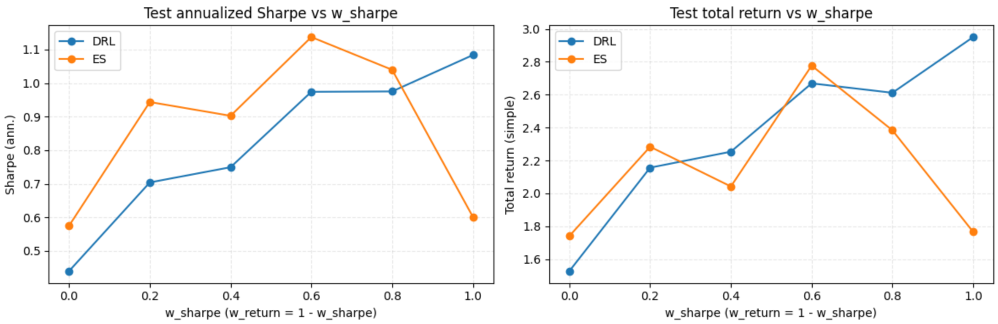

Figure 1: Test annualized Sharpe (left) and test total return (right) for DRL vs ES as the objective weight moves from pure Sharpe ($w_{\text{sharpe}} = 1$) to pure return ($w_{\text{sharpe}} = 0$).

both optimizers can produce powerful but brittle solutions. The key practical point is that although the overfitting signal is visually extreme on the training period, several configurations still generalize well and outperform the SPY benchmark on the test period, indicating that the learned strategies are not purely memorization artifacts.

Looking more closely at settings, the Sharpe-heavy objectives ($w_{\text{sharpe}} = 1.0$ and 0.8) typically show the strongest out-of-sample behavior for DRL as it beats SPY on the test interval with test wealth growing to roughly triple in the pure Sharpe DRL run. ES at $w_{\text{sharpe}} = 1.0$ achieves a strong in-sample Sharpe and can exceed DRL on Sharpe during training, but the out-of-sample return is weaker and the test performance falls behind SPY, consistent with the idea that optimizing a ratio objective can be sensitive to regime shifts. At $w_{\text{sharpe}} = 0.8$, ES still overfits heavily but can slightly beat SPY on the test set, with the learned portfolio often emphasizing defensive assets such as bonds and gold.

The intermediate setting $w_{\text{sharpe}} = 0.6$ appears to be the most robust trade-off in these experiments. DRL at $(0.6, 0.4)$ continues to show the same qualitative overfit pattern (very large train wealth), yet remains strongly positive out-of-sample and beats SPY on the test period, with a test portfolio still largely driven by VNQ and GLD exposure. ES at $(0.6, 0.4)$ is particularly strong, despite also exhibiting large train-set gains, it produces one of the clearest improvements over SPY on the test period and corresponds to the best overall ES trade-off in Figure 1. In this region, ES also shifts away from the bond-heavy allocations seen in Sharpe-dominant settings and becomes more equity and gold-weighted on the test set.

As the objective shifts further toward pure return ($w_{\text{sharpe}} = 0.4$ and 0.2), test performance becomes less consistently above SPY. DRL and ES shows a smoother loss profile as the weights in objective function shifts more towards return. ES at $w_{\text{sharpe}} = 0.4$ test performance is slightly weaker than SPY in our notes, with the learned portfolio remaining more diversified (notably with persistent bond exposure) than DRL. At $w_{\text{sharpe}} = 0.2$, both methods still do not clearly dominate SPY out-of-sample; ES tends to maintain a diversified mix including gold, bonds, and equities, while DRL remains concentrated in VNQ with secondary allocations, and the test curves suggest that emphasizing return alone reduces robustness to the held-out regime.

Finally, the pure return setting ($w_{\text{sharpe}} = 0$) shows the clearest failure mode. Both DRL and ES can still generate large training gains (e.g., DRL reaching $\sim 14\times$ on train), but out-of-sample performance collapses relative to Sharpe-heavy or mixed settings, with test wealth rising only modestly (around $1.5\times$ in the DRL note) and underperforming SPY. This supports the core motivation for including the Sharpe term as without explicit risk adjustment, the policy tends to chase unstable patterns that do not transfer.

Average test-period allocations (Table 1) highlight systematic differences in the solutions found by the two optimizers. DRL converges to a similar structure across the sweep, dominated by VNQ with secondary exposure to GLD and U.S. equities. One plausible reason is that backpropagation follows local gradients of the differentiable objective, and once the softmax output becomes concentrated, moving mass to other assets can require large logit changes. This can lock optimization into a stable basin that looks similar across nearby objective weights. ES, by contrast, evaluates many parameter

Algo	w_s	w_r	VNQ	IWM	GLD	SPY	VGK	EFA	AGG	EEM	Cash
DRL	1.0	0.0	0.475323	0.068518	0.207732	0.122996	0.006629	0.006816	0.011011	0.099742	0.001234
DRL	0.8	0.2	0.503209	0.065989	0.178513	0.110464	0.006880	0.009494	0.019203	0.104417	0.001831
DRL	0.6	0.4	0.523833	0.087158	0.222612	0.086653	0.006511	0.013806	0.022052	0.035089	0.002287
DRL	0.4	0.6	0.586212	0.108593	0.191693	0.070986	0.004147	0.014330	0.012394	0.009397	0.002246
DRL	0.2	0.8	0.595480	0.120232	0.197189	0.041748	0.006380	0.012318	0.013834	0.007732	0.005089
DRL	0.0	1.0	0.614927	0.079704	0.155954	0.038151	0.017638	0.026312	0.023480	0.031864	0.011970
ES	1.0	0.0	0.197759	0.035514	0.150496	0.240157	0.042055	0.000000	0.334016	0.000000	0.000002
ES	0.8	0.2	0.256209	0.044108	0.315006	0.168308	0.007611	0.001486	0.201033	0.004021	0.002219
ES	0.6	0.4	0.267950	0.033432	0.256169	0.270388	0.007419	0.001990	0.159288	0.002870	0.000493
ES	0.4	0.6	0.288520	0.065947	0.193264	0.244494	0.018608	0.007075	0.165958	0.014290	0.001845
ES	0.2	0.8	0.372779	0.107750	0.229007	0.130360	0.009269	0.008022	0.134134	0.005760	0.002922
ES	0.0	1.0	0.554830	0.083201	0.170017	0.043491	0.021327	0.028592	0.045022	0.038892	0.014626

Table 1: Average portfolio weights over the *test period* for each objective setting. $w_s = w_{\text{sharpe}}$ and $w_r = w_{\text{return}} = 1 - w_s$.

perturbations each iteration, which effectively optimizes a smoothed objective in parameter space. That smoothing and broader exploration makes ES more responsive to changes in the Sharpe–return mix, producing more defensive allocations (notably higher AGG weight) when w_{sharpe} is large. At the same time, the Sharpe ratio is a noisy, non-linear objective, therefore fitness variance increases near the pure-Sharpe extreme, so ES updates can become less reliable there despite mirroring.

5 Conclusion

We compared differentiable policy optimization (DRL) and mirrored Evolution Strategies (ES) for training the same MLP allocation policy on an ETF portfolio task with transaction costs. Across a sweep from pure Sharpe to pure return, both methods can overfit in sample, yet several Sharpe-heavy and mixed settings still outperform SPY on the held-out test period. DRL behaves smoothly as the objective weight varies and is most reliable when Sharpe dominates, while ES is particularly strong in the mixed regime around $w_{\text{sharpe}} = 0.6$ but is more sensitive at the extremes.

These results are based on a single train–test split and a small, curated universe of broad ETFs. Outperformance may partly reflect asset-selection bias: the chosen assets are well known diversifiers that performed favorably over the studied period, so beating SPY is not evidence that the method would generalize to an arbitrary set of stocks. The backtest also uses simplified frictions (constant proportional transaction costs, zero cash yield, no slippage or liquidity constraints), and financial time series are non-stationary, so a policy that works on this historical window may degrade in a future regime. Scaling to a large stock universe would increase the action dimension and change the learning problem, consequently the same small MLP and feature set may be insufficient without stronger regularization, constraints, and more robust validation.

A Additional figures

The following plots illustrate allocations and wealth curves for the mixed objective $w_{\text{sharpe}} = 0.6, w_{\text{return}} = 0.4$.

References

[1] H. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.

[2] W. F. Sharpe. Mutual fund performance. *Journal of Business*, 39(1):119–138, 1966.

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

[4] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv:1703.03864*, 2017.

[5] W. Ala-Krekola. *Financial Portfolio Management with Evolution Strategies Based Reinforcement Learning*. Master's Thesis, Aalto University School of Business, Spring 2021.

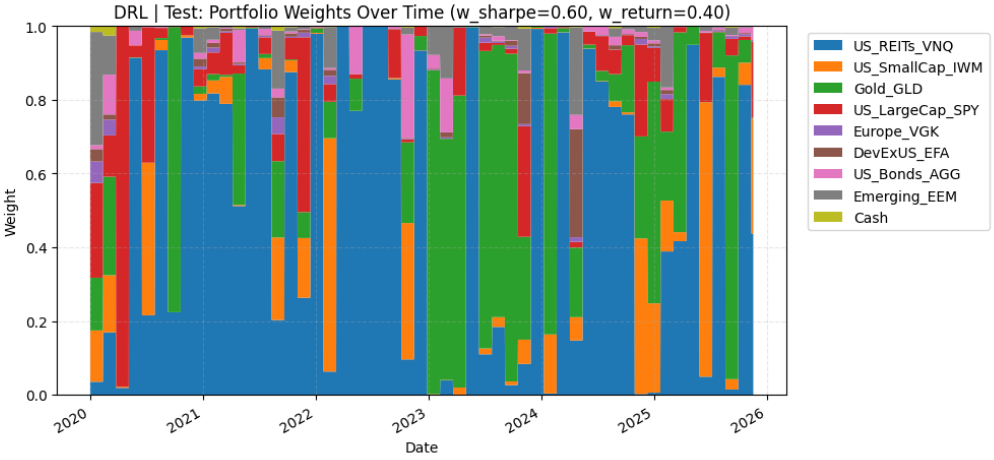

Figure 2: DRL test-period asset weights for $w_{\text{sharpe}} = 0.6, w_{\text{return}} = 0.4$.

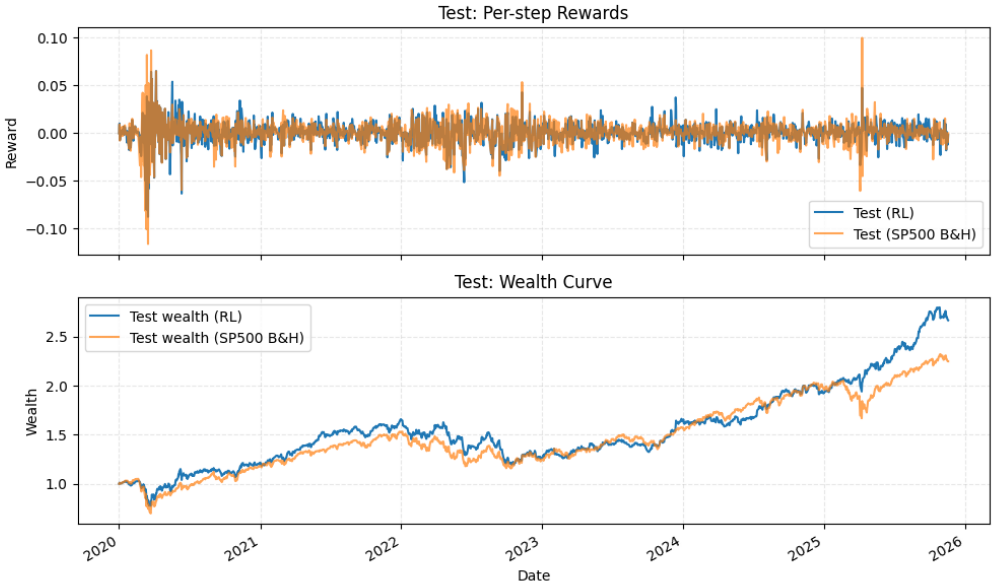

Figure 3: DRL test-period wealth curve for $w_{\text{sharpe}} = 0.6, w_{\text{return}} = 0.4$.

[6] R. Hasani et al. Exploring modern evolution strategies in portfolio construction. Optimization and Machine Learning Workshop (OptML), 2023.

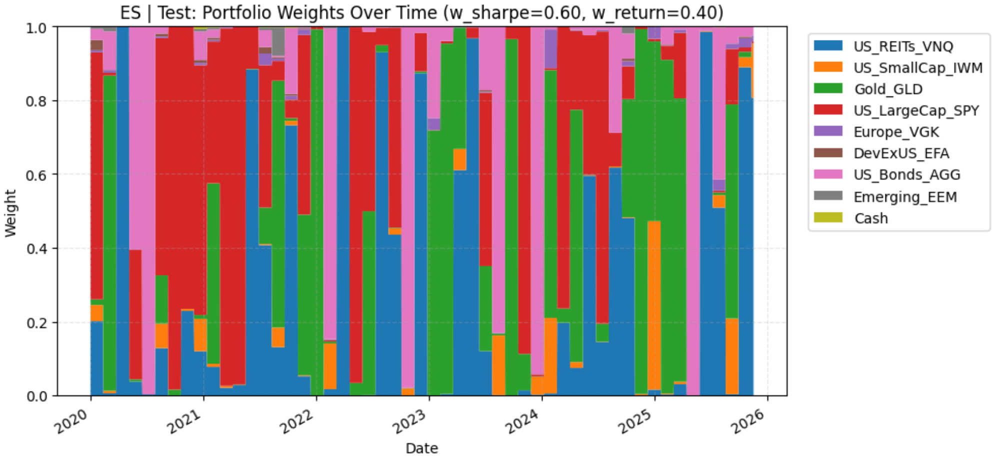

Figure 4: ES test-period asset weights for $w_{\text{sharpe}} = 0.6, w_{\text{return}} = 0.4$.

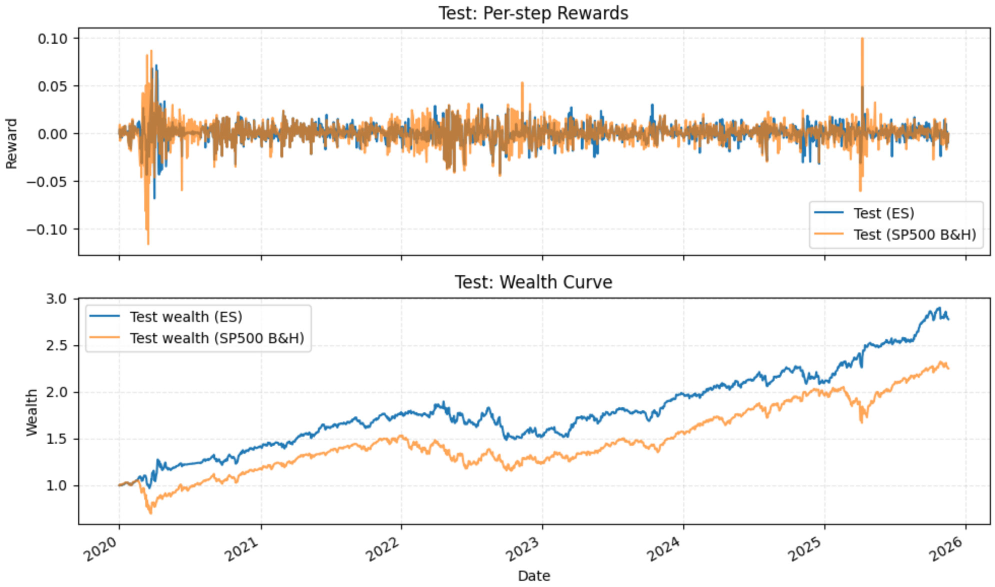

Figure 5: ES test-period wealth curve for $w_{\text{sharpe}} = 0.6, w_{\text{return}} = 0.4$.