Extended Abstract

Motivation Equity markets are a notoriously hard setting for reinforcement learning: returns are non-stationary, regime-driven, and have a very low signal-to-noise ratio—quarterly S&P 500 selection yields on the order of twenty independent decisions per test window, which is barely enough to distinguish skill from luck. The efficient-markets hypothesis [Fama, 1970] suggests that persistent, easily exploited public-information edges should be difficult to maintain, and the empirical record broadly agrees. A recurring failure mode in this space is that a single "lucky" training run is reported as a result, while the underlying policy is in fact dominated by random-seed variance rather than learned skill. Prior work in deep RL for portfolio management [Jiang et al., 2017, Pigorsch and Schäfer, 2021] reports promising empirical results, but reproducibility across random seeds is rarely assessed. I ask a narrower and more honest question: can a PPO agent select an S&P 500 portfolio in a way that is *reproducible* across seeds, and what kind of edge—if any—survives once I report the full distribution rather than the best run?

Method I build a PPO agent [Schulman et al., 2017] with Generalized Advantage Estimation [Schulman et al., 2016] that, each quarter, selects a top-K equal-weight portfolio from the point-in-time S&P 500 universe. A shared, permutation-invariant scoring network [Zaheer et al., 2017] applies a single-hidden-layer MLP to every stock independently, producing a scalar score per name; a mean-pooled value head over the same representations serves as the critic. This set-based architecture handles variable-size universes and is invariant to the arbitrary ordering of stocks in the input. Selection of K stocks without replacement uses the Gumbel-Top-k trick [Kool et al., 2019], with the action log-probability given by the Plackett–Luce model [Plackett, 1975]; the reward is realized quarterly log-return minus a turnover penalty to discourage churn. The policy selects all stocks at equal weight $w_i = 1/K$, which deliberately constrains the action space relative to a free-form allocator and trades expressiveness for stability.

Implementation ALl data are pulled from WRDS (Wharton Research Data Services accessible by any Stanford student) (Compustat–CRSP for fundamentals and returns, I/B/E/S for earnings estimates), with point-in-time S&P 500 membership including delisted and removed names to avoid survivorship and look-ahead bias. The feature set covers $\sim$25 signals per stock: valuation (book-to-market, earnings yield), profitability (gross profit-to-assets [Novy-Marx, 2013], ROE), growth, leverage, momentum [Jegadeesh and Titman, 1993], and standardized-unexpected-earnings features motivated by post-earnings-announcement drift [Bernard and Thomas, 1989]. Features are z-scored *within each GICS sector* each quarter—using the sector's own mean and standard deviation rather than the full universe's—so each feature measures how a stock compares to its same-sector peers. An earlier version of this project used explicit one-hot GICS indicators as input features, where they significantly improved performance; moving that sector information into the normalization step produced more stable results and made the explicit indicators redundant. I train on 1998–2015, validate on 2016–2019, and test on 2020–2024, selecting the checkpoint with the best validation Sharpe. All seeds are trained concurrently on Modal cloud infrastructure.

Results At $K = 60$, over five random seeds, the agent attains a mean test Sharpe of 1.19 ± 0.24, a total return of $+134.3\% \pm 22.3$, and a maximum drawdown of -19.9%, versus Sharpe 0.81 / drawdown -23.0% for the equal-weight S&P and Sharpe 1.11 / drawdown -23.8% for the cap-weighted S&P. Four of the five seeds cluster tightly between Sharpe 1.22 and 1.40; one outlier (seed 4) underperforms at 0.73 with a -33% drawdown, driving the reported standard deviation. The agent beats equal-weight on Sharpe, matches the cap-weighted index on risk-adjusted return on the mean (the two overlap within one standard deviation), and beats both benchmarks on drawdown—the most consistent finding across seeds.

Discussion The dominant source of variation in test performance is the random seed, not any hyperparameter or design choice—the pathology that reliable RL evaluation work warns against [Henderson et al., 2018, Agarwal et al., 2021]. Across a one-factor-at-a-time hyperparameter sweep, the inter-seed spread in test Sharpe exceeded the effect of any individual hyperparameter, making safe test-set model selection impossible. Validation also does not cleanly predict test: the portfolio size K favored on the 2016–2019 validation window differs from the one favored on the 2020–2024 test window, a direct symptom of the COVID-era regime shift reshaping the cross-section of returns. Training curves show reward stabilizing but validation Sharpe oscillating without monotonic improvement—consistent with a low-signal, non-stationary objective where there is limited structure to climb. These observations argue strongly for multi-seed reporting as a default practice in financial RL [Karzanov et al., 2025].

Conclusion Relative to a noisy return-maximizing predecessor whose test Sharpe swung from $+0.86$ to -0.32 across seeds, the top-K equal-weight redesign dramatically reduces seed variance and yields a portfolio that is competitive with passive benchmarks on a risk-adjusted basis, with a consistent drawdown advantage. The primary contribution is methodological: a more reproducible PPO pipeline for quarterly portfolio selection, built on within-sector feature normalization, a set-based permutation-invariant architecture, and a constrained equal-weight action space. Future work should move toward richer policies with variable position counts and continuous per-stock weights, while carrying forward the reproducibility practices established here.

Reproducible Top-K PPO for S&P 500 Portfolio Selection: Risk-Adjusted Gains, Seed Variance, and the Limits of Return-Maximizing RL

Andres Restrepo
Department of Computer Science
Stanford University
andresfr@stanford.edu

Abstract

I study proximal policy optimization (PPO) for quarterly S&P 500 portfolio selection and focus on a question that is often skipped in financial RL: reproducibility. I design a permutation-invariant, set-based policy that selects a top-K equal-weight portfolio each quarter via the Gumbel-Top-k trick, trained on point-in-time data (1998–2015) with $\sim$25 fundamental, momentum, and earnings-surprise features, and evaluated out-of-sample on 2020–2024. Over five seeds the agent attains a mean test Sharpe of 1.19 ± 0.24 with a -19.9% maximum drawdown, beating the equal-weight S&P on Sharpe, remaining competitive with the cap-weighted index on risk-adjusted return, and beating both on drawdown. I find that test performance is dominated by random-seed variance rather than by any hyperparameter or architectural choice, and that validation does not reliably predict test performance under the 2020–2024 regime shift. Relative to a return-maximizing predecessor that was effectively irreproducible, my design substantially reduces seed variance. I frame the result as consistent with the efficient-markets hypothesis and argue that distribution-level, multi-seed reporting should be the default in this setting.

1 Introduction

Applying reinforcement learning to portfolio management is appealing: the problem is naturally sequential, the agent receives a scalar reward (return), and large universes of assets provide a rich, high-dimensional state space [Jiang et al., 2017, Pigorsch and Schäfer, 2021]. It is also unusually treacherous. Equity returns are non-stationary and regime-driven—a strategy that thrives in one macroeconomic environment can collapse in the next—and the efficient-markets hypothesis [Fama, 1970] suggests that simple, persistent public-information edges should be difficult to maintain. Compounding this, quarterly decision-making yields very few independent decisions per test window (on the order of twenty), so the signal-to-noise ratio available to the learner is extremely low.

These conditions make financial RL especially vulnerable to a subtle but serious evaluation failure. Deep RL algorithms are known to exhibit high run-to-run variance, and reporting a single training run—or, worse, the best of several—can produce conclusions that do not survive replication [Henderson et al., 2018, Agarwal et al., 2021]. In finance this is acute: a policy whose test Sharpe is governed by the random seed can be made to look skillful simply by choosing a fortunate run. An earlier iteration of this project exhibited exactly this pathology. A return-maximizing PPO agent, evaluated across otherwise-identical seeds, produced test Sharpe ratios ranging from $+0.86$ down to -0.32; the "best" configuration was not reproducible, and validation performance did not predict which seed would do well on test.

In this work I take reproducibility as the central object of study rather than an afterthought. I redesign the agent around a top-K equal-weight action space: each quarter the policy scores every eligible stock with a shared, permutation-invariant network [Zaheer et al., 2017], selects the top K via the Gumbel-Top-k trick [Kool et al., 2019], and holds them in equal weight. This constrains the policy's degrees of freedom relative to a free-form allocator, trading expressiveness for stability. I train with PPO [Schulman et al., 2017] and Generalized Advantage Estimation [Schulman et al., 2016] on point-in-time S&P 500 data spanning 1998–2024, and I report all results over multiple seeds.

My contributions are: (i) a permutation-invariant, set-based PPO formulation for variable-membership portfolio selection with a stochastic top-K action with tractable Plackett–Luce log-probabilities; (ii) an empirical demonstration that this design is substantially more reproducible than a return-maximizing predecessor, reducing seed-to-seed Sharpe

variance from a $+0.86 \to -0.32$ spread to a standard deviation of 0.24; (iii) an honest, distribution-level evaluation showing that the agent matches a cap-weighted benchmark on risk-adjusted return and beats both passive benchmarks on drawdown, while making explicit that the dominant driver of test performance remains the random seed. These results are consistent with the EMH: the agent's reliable edge is in risk management, not in beating the market on returns.

2 Related Work

Deep RL for portfolio management. Jiang et al. [2017] introduced an end-to-end deep RL framework for portfolio allocation, and Pigorsch and Schäfer [2021] proposed a Deep Q-learning approach for high-dimensional stock portfolio trading, both reporting promising empirical performance. Most relevant here, Karzanov et al. [2025] use PPO to enhance a fixed benchmark via dynamic rebalancing, and train many independent agents per period and report *average* performance precisely because of the high stochasticity of financial RL. My work shares their emphasis on distribution-level evaluation but differs in objective: rather than enhancing a known-good strategy, I ask how reproducible a from-scratch selection policy can be made, and I treat seed variance as a primary measurement rather than something to be averaged away silently.

Reliable evaluation in RL. Henderson et al. [2018] documented how sensitive deep RL results are to seeds, implementation details, and hyperparameters, and Agarwal et al. [2021] showed that point estimates over a handful of runs routinely mislead, advocating interval estimates and stratified reporting. These findings are the lens through which I interpret my own results: the failure of my earlier return-maximizing agent was not a bug to be patched but the expected behavior of an under-constrained policy in a low-signal environment.

Differentiable selection and set-based policies. My action space relies on the Gumbel-Top-k trick for sampling k items without replacement [Kool et al., 2019], with selection probabilities described by the Plackett–Luce model [Plackett, 1975]. To handle a universe whose membership changes every quarter I use a permutation-invariant, shared-encoder architecture following Deep Sets [Zaheer et al., 2017].

Cross-sectional return signals. My features draw on the empirical asset-pricing literature: the gross-profitability premium [Novy-Marx, 2013], price momentum [Jegadeesh and Titman, 1993], and post-earnings-announcement drift [Bernard and Thomas, 1989], the last of which motivates my standardized-unexpected-earnings (SUE) features.

3 Method

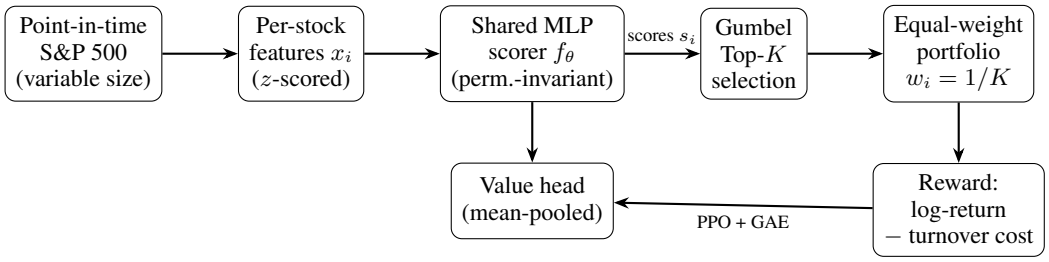

Figure 1: Method overview. Each quarter, a shared permutation-invariant network scores every eligible stock; the policy selects K stocks via the Gumbel-Top-k trick and holds them in equal weight. The critic mean-pools a value head over the universe. The agent is trained with PPO and GAE on the realized-log-return-minus-turnover reward.

Problem setup. I model quarterly portfolio selection as a finite-horizon MDP. At decision time t the state is the set of feature vectors $\{x_{i,t}\}_{i \in \mathcal{U}_t}$ for the eligible universe $\mathcal{U}_t$ (point-in-time S&P 500 membership), where each $x_{i,t} \in \mathbb{R}^F$ contains $\sim$25 standardized cross-sectional features. The action selects a subset $S_t \subseteq \mathcal{U}_t$ with $|S_t| = K$, the portfolio is held at equal weight $w_{i,t} = 1/K$ for $i \in S_t$, and the agent then observes the realized quarterly returns and forms the next state.

Permutation-invariant scoring policy. The universe is not a fixed-length vector—names enter and leave the index every quarter—so the policy must be invariant to ordering and robust to changing cardinality. I therefore use a shared encoder applied independently to each stock [Zaheer et al., 2017]. A single MLP maps each feature vector to a hidden representation $h_i = \tanh(Wx_i + b)$ with $h_i \in \mathbb{R}^{32}$, from which a linear *score head* produces a scalar score

$$s_i = f_\theta(x_i) = w_s^\top h_i + b_s, \tag{1}$$

and a linear *value head* produces a per-stock value that is mean-pooled across the universe to form a single, permutation-invariant critic estimate $V_\theta = \frac{1}{|\mathcal{U}_t|} \sum_i (w_v^\top h_i + b_v)$.

Top-K action via the Gumbel-Top-k trick. Given scores $\{s_i\}$, sampling K stocks without replacement in proportion to $\exp(s_i)$ is exactly the Gumbel-Top-k procedure [Kool et al., 2019]: I perturb each score with i.i.d. Gumbel noise, $\tilde{s}_i = s_i + g_i$, $g_i \sim \mathrm{Gumbel}(0, 1)$, and take the K largest $\tilde{s}_i$. This yields a stochastic top-K selection that provides exploration during training; at evaluation I take the deterministic top-K by score. The probability of drawing the ordered selection $(\sigma_1, \ldots, \sigma_K)$ follows the Plackett–Luce model [Plackett, 1975],

$$\log \pi_\theta(S_t \mid s) = \sum_{j=1}^{K} \left[s_{\sigma_j} - \log \sum_{i \in \mathcal{U}_t \setminus \{\sigma_1, \ldots, \sigma_{j-1}\}} \exp(s_i) \right], \tag{2}$$

which I use as the action log-probability inside the PPO objective. Equal weighting $w_i = 1/K$ is applied to the selected set; I treat conviction-based weighting as an ablation (Appendix A).

Reward. The per-step reward is the realized portfolio log-return net of transaction costs, charged on turnover:

$$r_t = \log\Big(1 + \sum_{i \in S_t} w_{i,t}\, R_{i,t+1}\Big) \;-\; c \sum_i \big| w_{i,t} - w_{i,t-1} \big|, \tag{3}$$

where $R_{i,t+1}$ is the realized quarterly return of stock i and c is a per-unit cost coefficient. Log-return makes the objective additive across quarters and penalizes large drawdowns implicitly; the turnover term discourages churn.

Optimization. I train with PPO [Schulman et al., 2017] using the clipped surrogate objective

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\big(\rho_t(\theta)\,\hat{A}_t,\ \mathrm{clip}(\rho_t(\theta),\, 1-\epsilon,\, 1+\epsilon)\,\hat{A}_t\big)\Big], \quad \rho_t(\theta) = \frac{\pi_\theta(S_t \mid s_t)}{\pi_{\theta_{\text{old}}}(S_t \mid s_t)}, \tag{4}$$

with advantages $\hat{A}_t$ computed by Generalized Advantage Estimation [Schulman et al., 2016]. The critic is the mean-pooled value head above, and I add an entropy bonus on the selection distribution to maintain exploration.

Sector information. GICS sector membership enters the model through the normalization itself rather than as a separate feature. Each quarter I z-score every feature *within each GICS sector*, so a stock's standardized features measure how it ranks against same-sector peers rather than against the universe at large. This is motivated by the v1/v2 history of this project (see Appendix A): in the earlier return-maximizing agent, appending one-hot GICS indicators significantly improved performance, indicating the model was using sector identity to interpret the other features. In the redesigned agent I move that information into the normalization step, which makes explicit sector indicators redundant and removes them from the headline configuration.

4 Experimental Setup

Data and universe. I use point-in-time S&P 500 membership over 1998–2024, including delisted and removed names, to eliminate survivorship and look-ahead bias. Quarterly fundamentals and returns come from WRDS (Compustat–CRSP merged); analyst estimates for the SUE features come from I/B/E/S, using unadjusted estimates with the most recent statistical period strictly preceding each announcement date to avoid look-ahead.

Features. Each stock carries $\sim$25 cross-sectional features spanning valuation (book-to-market, earnings yield), profitability (gross profit-to-assets [Novy-Marx, 2013], ROE, margins), growth (asset and earnings growth), leverage, and momentum (trailing returns and volatility [Jegadeesh and Titman, 1993]), plus standardized-unexpected-earnings signals (SUE, SUE momentum, and 2–12 month momentum) motivated by post-earnings-announcement drift [Bernard and Thomas, 1989]. Features are z-scored *within each GICS sector* each quarter, using that sector's own mean and standard deviation rather than the universe's, so each feature measures how a stock compares to its same-sector peers. Values are winsorized to tame outliers and missing values are imputed by sector median; a single row with an undefined price is dropped (a stock cannot be traded at an undefined price).

Splits and selection. Train 1998–2015, validation 2016–2019, test 2020–2024. Within each run, I select checkpoints using validation Sharpe. Hyperparameter sweeps are reported diagnostically rather than used to claim an unbiased test-selected optimum, so the main conclusions emphasize seed distributions and validation-to-test instability. I report all metrics over five random seeds with identical hyperparameters.

Infrastructure. Training runs are executed in parallel on Modal cloud infrastructure, which lets me launch the full multi-seed sweep concurrently and collect per-seed equity curves, sector allocations, and training diagnostics.

Benchmarks and metrics. I compare against the equal-weight and cap-weighted S&P 500. I report annualized Sharpe ratio, total return, and maximum drawdown over the test window.

Table 1: Out-of-sample performance, 2020–2024 test window. Agent figures are mean ± standard deviation over five seeds for the reported $K = 60$ configuration (within-sector normalized features, equal-weight portfolio). Sharpe is annualized.

Method	Test Sharpe	Total Return	Max Drawdown
Equal-weight S&P 500	0.81	$+101.6\%$	-23.0%
Cap-weighted S&P 500	1.11	$+127.6\%$	-23.8%
PPO agent (ours)	$\mathbf{1.19 \pm 0.24}$	$\mathbf{+134.3\% \pm 22.3}$	$\mathbf{-19.9\%}$

5 Results

In the most stable large-portfolio configuration examined, $K = 60$, the agent attains a mean test Sharpe of 1.19 ± 0.24, a total return of $+134.3\% \pm 22.3$, and a maximum drawdown of -19.9% (Table 1). This beats the equal-weight S&P on Sharpe (0.81), is competitive with the cap-weighted index (1.11) on risk-adjusted return, and improves on both benchmarks' drawdowns (-23.0% and -23.8%). The equity curves (Figure 3) show the agent tracking and modestly leading both benchmarks over the test window, with the seed band widening in the post-2022 recovery.

5.1 Quantitative Evaluation

Risk-adjusted return and drawdown. On the mean, the agent's Sharpe exceeds both benchmarks, but the cap-weighted index (1.11) lies inside the agent's one-standard-deviation band (1.19 ± 0.24); I therefore claim a *match* with cap-weighted on risk-adjusted return rather than a clean win, and a clear win over equal-weight. The drawdown advantage is more robust: it holds for four of the five seeds and is visible across the entire 2020–2024 stress period in Figure 2, where the agent's drawdown path sits above both benchmarks through the 2022 trough.

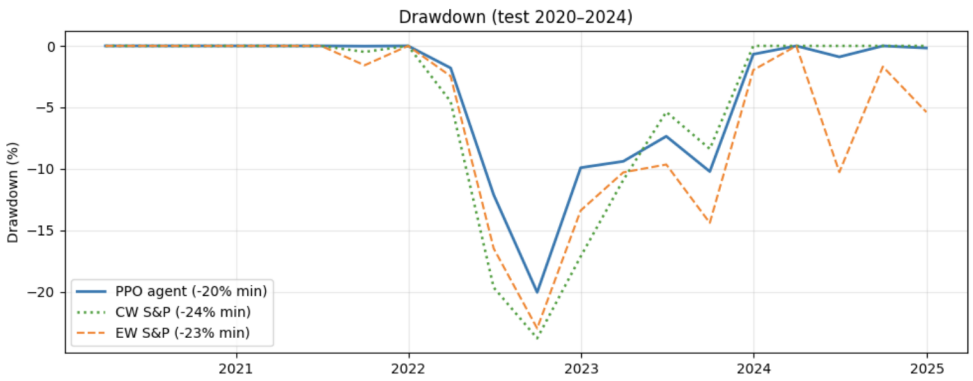

Figure 2: Drawdown over the 2020–2024 test window. The agent's drawdown path stays above both benchmarks through the 2022 trough, the most consistent edge I observe (holding for four of five seeds).

Reproducibility. The per-seed results are $\{+1.38, +1.23, +1.22, +1.40, +0.73\}$ in Sharpe, i.e. four tightly-clustered seeds in $[1.22, 1.40]$ and one outlier (seed 4) at $+0.73$ with a -33% drawdown. This yields the reported standard deviation of 0.24. For comparison, the earlier return-maximizing design produced test Sharpe ranging from $+0.86$ to -0.32 under the same seed-only perturbation. The redesign thus removes the sign-flipping instability and shrinks the spread substantially, while not eliminating residual seed sensitivity—the kind of partial, honestly-reported improvement that Agarwal et al. [2021] argue should be the norm.

Seed dominates design. Across my one-factor-at-a-time hyperparameter exploration, the variation in test Sharpe across random seeds was comparable to, and often larger than, the variation from individual hyperparameter changes. The practical consequence is that test-set model selection is unsafe in this regime: choosing the configuration with the best test Sharpe would largely be selecting a lucky seed.

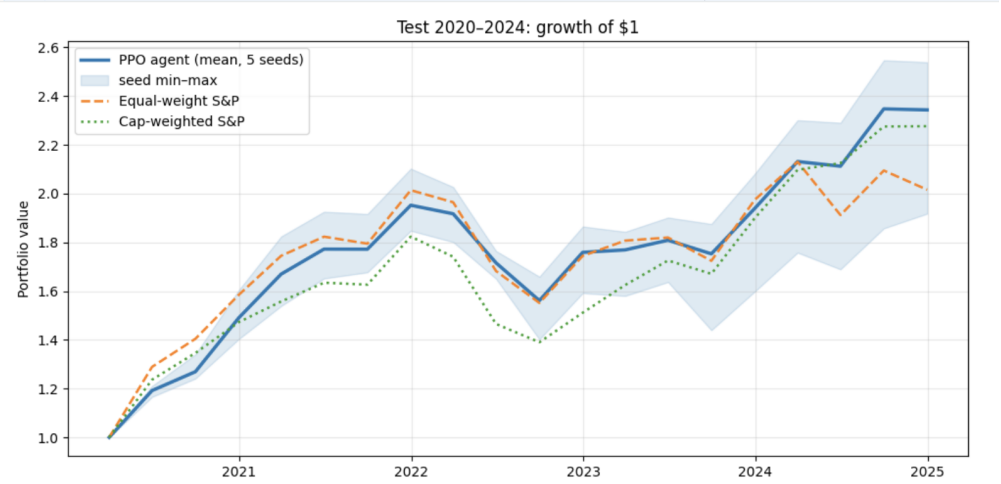

Figure 3: Growth of $1 over the 2020–2024 test window. The solid line is the mean over five seeds; the shaded band is the seed min–max. The agent tracks and modestly leads both the equal-weight and cap-weighted S&P 500.

5.2 Qualitative Analysis

Allocation behavior. Figure 4 shows the agent's sector allocation across the test quarters for a representative seed. The portfolio is broadly diversified across GICS sectors for most of the window, with occasional sharp concentration into a single sector in specific quarters. This concentration is a candidate target for future constraints (Section 6).

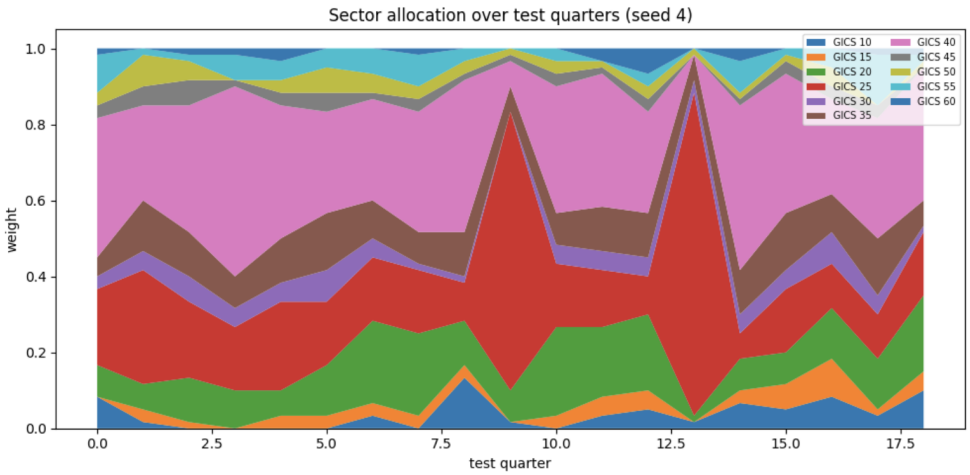

Figure 4: Sector allocation across test quarters for a representative seed. The policy is broadly diversified across GICS sectors but occasionally concentrates heavily in a single sector in specific quarters.

Training dynamics. Figure 5 plots the training reward and validation Sharpe for a representative high-performing seed. Reward improves and stabilizes, but the validation Sharpe oscillates without monotonic improvement. I read this not as a training failure but as a direct fingerprint of a low-signal, non-stationary objective: there is limited learnable structure to climb, so validation performance fluctuates around a noisy plateau.

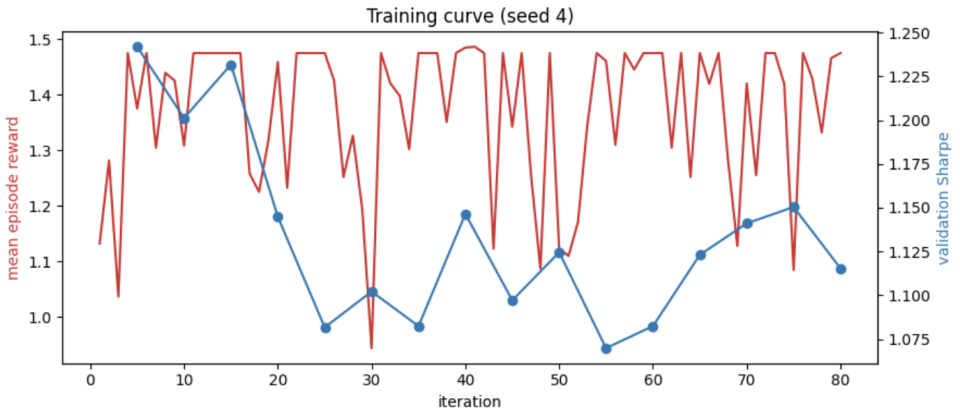

Figure 5: Training reward (left axis) and validation Sharpe (right axis) for a representative high-performing seed. Reward stabilizes while validation Sharpe oscillates around a noisy plateau, consistent with a low-signal, non-stationary objective.

6 Discussion

The headline takeaway is that, once I report the full seed distribution rather than a single run, the agent's reliable contribution is *risk management*, not return generation. It matches a cap-weighted benchmark on risk-adjusted return and beats both passive benchmarks on drawdown, but it does not deliver a clean, reproducible excess return—and the EMH does not predict such an edge in the first place [Fama, 1970]. This is also why my earlier return-maximizing agent was so fragile: with very few effective decisions per test window and a non-stationary objective, an under-constrained policy has ample freedom to overfit idiosyncratic seed dynamics, producing Sharpe ratios that flip sign across seeds. Constraining the action space to top-K equal-weight trades expressiveness for stability and is what makes the redesigned agent reproducible enough to discuss honestly.

Two findings deserve emphasis. First, *validation does not predict test*. The portfolio size and configuration favored on the 2016–2019 validation window are not those favored on the 2020–2024 test window—a direct consequence of the COVID shock and 2022 drawdown reshaping the cross-section of returns. Honest model selection on validation therefore cannot reliably climb toward the best test outcome; reporting one favorable run as "the result" would be misleading. Second, *seed variance exceeds design effects*, echoing the broader reliability literature [Henderson et al., 2018, Agarwal et al., 2021] and the multi-seed averaging adopted in contemporary financial RL [Karzanov et al., 2025]. The appropriate unit of analysis is the distribution over seeds, not a point estimate.

Limitations. My test window is a single, unusually turbulent five-year period with $\sim$20 quarterly decisions, so absolute performance numbers should be read as indicative rather than definitive. The occasional single-sector concentration in Figure 4 indicates the equal-weight constraint controls per-name but not per-sector risk.

Future work. The natural next step is a substantially richer policy: rather than selecting a fixed-K equal-weight basket, allow a variable number of positions with continuous, per-stock weights—in effect letting the agent choose both *which* stocks to hold and *how much* of each, with explicit sector- and concentration-level risk constraints. This is considerably harder to train stably, and the reproducibility lessons here—constrain the action space, report distributions, and avoid test-set model selection—are exactly the guardrails such an extension will need. Moving to monthly data would increase the number of decisions and improve statistical power, and seed-ensembled policies [Karzanov et al., 2025] would directly address the variance I document.

7 Conclusion

I presented a reproducible top-K PPO agent for S&P 500 portfolio selection built on a permutation-invariant, set-based policy with a Gumbel-Top-k action space. Over five seeds the agent matches a cap-weighted benchmark on risk-adjusted return and beats both passive benchmarks on drawdown, while substantially reducing the seed-to-seed instability that made a return-maximizing predecessor irreproducible. My central message is methodological and EMH-consistent: in a low-signal, non-stationary market, the honest result is a distribution over seeds, the reliable edge is in risk rather than return, and reproducibility deserves to be a priority objective.

8 Changes from Proposal

The proposal and poster framed the project around a sector-aware vs. sector-blind ablation under a return-maximizing objective and concluded that no configuration reliably beat the passive benchmarks. After diagnosing that those results were dominated by random-seed variance, I redesigned the agent around a top-K equal-weight action space and moved the sector information out of the feature vector (where one-hot indicators had been a major contributor in v1) and into the feature normalization itself (within-sector z-scoring in v2). Together with the constrained action space, that change is what produces the reproducible, risk-adjusted-competitive results reported here; the v1/v2 sector handling is documented in Appendix A.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 29304–29320, 2021.

Victor L. Bernard and Jacob K. Thomas. Post-earnings-announcement drift: Delayed price response or risk premium? *Journal of Accounting Research*, 27:1–36, 1989.

Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2): 383–417, 1970.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. arXiv:1709.06560.

Narasimhan Jegadeesh and Sheridan Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91, 1993.

Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*, 2017.

Daniil Karzanov, Rubén Garzón, Mikhail Terekhov, Caglar Gulcehre, Thomas Raffinot, and Marcin Detyniecki. Regret-optimized portfolio enhancement through deep reinforcement learning and future looking rewards. In *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF)*, pages 890–897, 2025. arXiv:2502.02619.

Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *PMLR*, pages 3499–3508, 2019.

Robert Novy-Marx. The other side of value: The gross profitability premium. *Journal of Financial Economics*, 108(1): 1–28, 2013.

Uta Pigorsch and Sebastian Schäfer. High-dimensional stock portfolio trading with deep reinforcement learning. *arXiv preprint arXiv:2112.04755*, 2021.

Robin L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016. arXiv:1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1703.06114.

A Additional Experiments

Sector information: features vs. normalization. In the earlier return-maximizing agent (v1), appending one-hot GICS sector indicators to the feature vector significantly improved test Sharpe over a version without them, indicating that the model was using sector identity as an extra signal to interpret the other features. The redesigned agent (v2) instead carries that information in the normalization step itself, z-scoring each feature against same-sector peers rather than the universe at large. Under this within-sector normalization the explicit one-hot sector indicators are no longer informative

on the margin—adding them back did not improve, and slightly trailed, the headline configuration. The natural reading is that the relevant sector content ("how does this stock look relative to other tech / energy / financial names?") has been absorbed into the standardized features, leaving the explicit indicators redundant.

Conviction weighting. I replaced equal weighting with a temperature-controlled softmax over the selected stocks' scores (conviction weighting). Across temperatures this did not beat equal weighting on test Sharpe and tended to increase turnover; I therefore keep equal weighting in the headline model.

Earnings-surprise features. Adding SUE-based features (SUE, SUE momentum, and 2–12 month momentum) motivated by post-earnings-announcement drift [Bernard and Thomas, 1989] did not produce a clear edge over the equal-weight baseline on the test window.

Validation/test K inversion. Sweeping the portfolio size K revealed that the value of K that maximizes validation Sharpe differs from the one that maximizes test Sharpe (validation favoring smaller, more concentrated portfolios and test favoring larger, more diversified ones), a concrete instance of the regime shift discussed in Section 6.

B Implementation Details

The scoring network is a shared single-hidden-layer MLP (hidden width 32, $\tanh$ activation) applied per stock, with separate linear score and value heads; the critic mean-pools the value head over the eligible universe. Selection uses the Gumbel-Top-k trick at training time (deterministic top-K at evaluation) with Plackett–Luce action log-probabilities. I optimize with PPO (clipped surrogate, ϵ clip, entropy bonus) and GAE-estimated advantages. Features are cross-sectionally z-scored and winsorized per quarter; missing values are imputed by sector median. Training was run in parallel across seeds on Modal cloud infrastructure, and the checkpoint with the best validation Sharpe was selected for test-time evaluation.

C Implementation Details

Team Contributions: This was a solo project by Andres Restrepo and did every part of this project.