International Journal of
Financial Studies

MDPI

Article

A Hierarchical Signal-to-Policy Learning Framework for Risk-Aware Portfolio Optimization

Jiayang Yu * and Kuo-Chu Chang

Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030, USA; kchang@gmu.edu
* Correspondence: jyu23@gmu.edu

Abstract

This study proposes a hierarchical signal-to-policy learning framework for risk-aware portfolio optimization that integrates model-based return forecasting, explainable machine learning, and deep reinforcement learning (DRL) within a unified architecture. In the first stage, next-period returns are estimated using gradient-boosted tree models, and SHAP-based feature attributions are extracted to provide transparent, factor-level explanations of the predictive signals. In the second stage, a Proximal Policy Optimization (PPO) agent incorporates both predictive forecasts and explanatory signals into its state representation and learns dynamic allocation policies under a mean–CVaR reward function that explicitly penalizes tail risk while controlling trading frictions. By separating signal extraction from policy learning, the proposed architecture allows the use of economically interpretable predictive signals to incorporate into the policy's state representation while preserving the flexibility and adaptability of reinforcement learning. Empirical evaluations on U.S. sector ETFs and Dow Jones Industrial Average constituents show that the hierarchical framework delivers higher and stable out-of-sample risk-adjusted returns relative to both a single-layer DRL agent trained solely on technical indicators, a mean–CVaR optimized portfolio using the same parameters used in the proposed hierarchical model and standard equal weight as well as index-based benchmarks. These results demonstrate that integrating explainable predictive signals with risk-sensitive reinforcement learning improves the robustness and stability of data-driven portfolio strategies.

Keywords: hierarchical reinforcement learning; portfolio optimization; Conditional Value-at-Risk (CVaR); XGBoost; explainable artificial intelligence (XAI); SHAP values; Proximal Policy Optimization (PPO)

Check for updates

Academic Editor: Kuan-Min Wang

Received: 21 January 2026
Revised: 27 February 2026
Accepted: 6 March 2026
Published: 13 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Dynamic portfolio allocation requires balancing return generation with effective risk control in the presence of changing market conditions. Classical mean–variance optimization (Markowitz, 1952) provides a foundational framework for this trade-off, but its reliance on variance as the sole risk measure and its sensitivity to estimation error can lead to poor downside protection and unstable portfolio allocations. These limitations have motivated the use of downside-oriented risk measures, such as Conditional Value-at-Risk (CVaR), which better capture extreme losses and tail risk.

More recently, deep reinforcement learning (DRL) has been applied to portfolio management as a flexible framework for sequential decision-making. By interacting with features directly from market environments, DRL agents can learn adaptive rebalancing

strategies and have shown promising empirical results. However, most existing DRL-based portfolio methods adopt end-to-end architectures that map raw market features directly to portfolio weights. Despite their empirical effectiveness, these black box designs entangle return prediction and allocation decisions, making portfolio actions difficult to interpret and limiting explicit control over risk and trading behavior.

This study proposes a hierarchical signal-to-policy learning framework that separates return prediction from portfolio allocation. In the first stage, supervised learning models are used to generate short-horizon return forecasts together with interpretable, factor-level explanations of those predictions. In the second stage, a reinforcement learning agent uses these model-driven signals to learn dynamic portfolio allocation policies under a risk-aware objective that accounts for downside risk and trading frictions. By decoupling signal extraction from policy learning, the framework separates predictive modeling from portfolio decision-making while preserving the flexibility and adaptability of reinforcement learning.

The proposed framework is evaluated on two representative equity universes, i.e., U.S. sector exchange-traded funds and constituents of the Dow Jones Industrial Average, based on a rolling out-of-sample back-testing design. The empirical results show that the hierarchical approach delivers stronger and more stable risk-adjusted performance than both end-to-end DRL benchmarks and traditional allocation strategies. The proposed model also provides more consistent downside risk control across market conditions.

The primary contribution of this work is to show that a hierarchical signal-to-policy architecture delivers more stable and robust out-of-sample performance under a mean–CVaR objective. By separating predictive signal extraction from risk-aware policy learning, the framework improves the robustness of dynamic portfolio construction under estimation uncertainty and tail risk.

The remainder of this work is organized as follows. Section 2 reviews the related literature on portfolio optimization, predictive modeling, reinforcement learning in finance, and explainable AI. Section 3 introduces the proposed signal-to-policy framework, detailing both the Stage-1 return-forecasting module and the Stage-2 PPO-based portfolio allocation agent. Section 4 describes empirical design, including data preparation, feature engineering, the back-testing protocol, and statistical evaluation, followed by a comprehensive analysis of the results. Section 5 concludes with the main findings, practical implications, and directions for future research.

2. Literature Review

2.1. Portfolio Optimization and Risk Management

Modern Portfolio Theory (MPT), first introduced by Markowitz (1952), established the mean–variance framework that remains the foundation of portfolio optimization theory. By defining the efficient frontier, MPT formalized the trade-off between expected return and risk and highlighted the role of diversification in reducing portfolio variance. Related early work by Roy (1952) further emphasized the importance of controlling downside outcomes. Subsequent theoretical extension includes that Sharpe (1964) proposed the Capital Asset Pricing Model (CAPM), which links expected returns to systematic market risk, while Ross (1976) introduced the Arbitrage Pricing Theory (APT), allowing multiple risk factors to explain asset returns. Empirical studies by Fama and French (1992, 1993) showed that firm characteristics such as size and value explain a significant portion of cross-sectional return variation, motivating the widespread use of multifactor asset-pricing models.

Despite its wide influence, the classical mean–variance framework has several well-known limitations. First, it relies on second-moment risk measures and implicitly assumes that returns are approximately normally distributed, which leads to symmetric treatment of upside and downside volatility (Harvey et al., 2010). Second, mean–variance optimization

is highly sensitive to estimation error, i.e., small changes in expected returns or covariance estimates can lead to large shifts in optimal portfolio weights (Hurley & Brimberg, 2015). As a result, portfolios optimized using historical estimates often perform poorly when evaluated out of sample, as shown by Welch and Goyal (2008).

To address these issues, a large body of research has focused on alternative risk measures and robust optimization approaches. Roy (1952) proposed the safety-first criterion, which emphasizes controlling the probability of extreme losses. Downside-focused measures, such as semivariance, were later introduced to better reflect investor preferences toward losses rather than total volatility (Harvey et al., 2010). A major advance was made by Rockafellar and Uryasev (2000, 2002), who introduced Conditional Value-at-Risk (CVaR) as a coherent risk measure that explicitly focuses on tail losses. CVaR-based optimization is computationally efficient and has been shown to provide stronger protection against large drawdowns than variance-based methods.

Other approaches aim to improve robustness with respect to parameter uncertainty. The Black–Litterman model (Black & Litterman, 1992) combines a market equilibrium prior with investor views to generate more stable expected return estimates. Risk-parity strategies allocate capital so that each asset contributes equally to total portfolio risk and have been shown to deliver more stable performance across market regimes than traditional heuristic allocations, such as the 60/40 portfolio (Chaves et al., 2011). Overall, this literature highlights the importance of accounting for estimation uncertainty, asymmetric return distributions, and tail risk in portfolio construction, providing strong motivation for frameworks that incorporate downside-risk penalties such as CVaR and adopt robust optimization principles. Extending CVaR-based portfolio construction to data-driven forecasting settings, Yu and Chang (2020) integrate neural network return prediction with simulation-based mean–CVaR optimization. Their results suggest that nonlinear predictive signals can enhance downside-risk-adjusted performance when embedded within coherent risk-aware portfolio frameworks.

Dynamic allocation under non-stationary market conditions has also been examined in adaptive rebalancing frameworks. Chang et al. (2017) emphasizes that portfolio weights must adjust to evolving market structure, characterizing asset allocation as a moving target problem rather than a static optimization exercise. This perspective highlights the importance of flexibility and responsiveness in multi-period portfolio design.

Evidence from empirical portfolio studies further highlights the trade-off between theoretical optimality and practical robustness. DeMiguel et al. (2009) show that the naïve 1/N allocation often outperforms mean–variance optimized portfolios out of sample due to estimation error and parameter instability. Related work from Kehinde et al. (2025) on volatility-targeting portfolios demonstrates that constraint-aware allocation rules can deliver stable risk profiles without relying on precise return forecasts. Together, these findings suggest that robust portfolio construction benefits from simplicity, explicit risk control, and resistance to estimation noise, which motivates approaches that prioritize stability over pure optimality.

Recent studies emphasize that implementation frictions materially affect long-only factor performance. Y. Chen and Israelov (2024a) show that frequent rebalancing erodes after-tax alpha and that buy-and-hold portfolios tilted toward value, quality, and defensive characteristics retain strong performance. Related work from Y. Chen and Israelov (2024b) on personalized indexing finds that portfolios are highly robust to stock and industry exclusions, suggesting that realistic constraints do not necessarily imply large efficiency losses. These findings motivate portfolio frameworks that explicitly control turnover and operate within practical constraints. These observations suggest that portfolio construction

methods must jointly address risk, estimation uncertainty, and implementation frictions within a dynamic decision-making framework.

2.2. Deep Reinforcement Learning and Predictive Modeling in Finance

Machine learning (ML) methods have become increasingly important in empirical asset pricing and return prediction. Traditional linear factor models often show weak out-of-sample forecasting performance (Welch & Goyal, 2008), partly because linear specifications cannot capture nonlinear interactions and structural patterns in financial data. Recent studies show that nonlinear ML models can significantly improve predictive accuracy. Gu et al. (2020) demonstrate that tree-based ensembles and neural networks outperform ordinary least squares when predicting U.S. equity returns, roughly doubling out-of-sample predictive R^2 by exploiting nonlinear relationships among firm characteristics. Fischer and Krauss (2018) similarly find that Long Short-Term Memory (LSTM) networks improve short-horizon return direction prediction and lead to higher risk-adjusted performance in trading applications.

These results suggest that ML models can extract weak but persistent signals from high-dimensional and noisy financial datasets. In particular, methods such as random forests and gradient-boosted trees perform well because they can identify a small number of influential predictors from a large pool of candidate variables, effectively performing automatic feature selection (L. Chen et al., 2024; Kelly et al., 2019; Krauss et al., 2017).

Building on advances in predictive modeling, reinforcement learning (RL) provides a framework for learning dynamic trading and portfolio allocation strategies in a sequential decision-making context. Unlike supervised learning, which focuses on forecasting returns and then relies on external rules to form portfolios, RL directly learns trading policies that optimize the long-horizon investment reward function. Early work by Moody et al. (1998) showed that RL systems could directly learn to maximize reward functions such as the Sharpe ratio. Almahdi and Yang (2017) further demonstrated that incorporating risk-adjusted objectives in recurrent RL frameworks leads to more stable performance than return-only optimization.

The development of deep reinforcement learning (DRL) has enabled agents to process high-dimensional states involving multiple assets and technical indicators. Jiang et al. (2017) proposed a convolutional–recurrent DRL architecture for cryptocurrency portfolio management and reported strong performance relative to heuristic strategies. Liu et al. (2022) applied an actor–critic DRL approach using DDPG to equity trading and observed higher cumulative returns and Sharpe ratios. The introduction of open-source platforms such as FinRL (Liu et al., 2020) has further standardized experimental environments and facilitated systematic evaluation of DRL algorithms, including DQN, DDPG, and PPO. Among these, Proximal Policy Optimization (PPO) (Schulman et al., 2017) has gained particular attention due to its training stability in noisy and non-stationary financial environments. More recent studies combine DRL with transformers, ensemble learning, and regime-switching mechanisms to improve robustness and tail-risk control (Choudhary et al., 2025; Gao et al., 2023). More recent work applies deep reinforcement learning to dynamic portfolio construction. Yu and Chang (2025) develop a DRL-based tangency portfolio framework in which the agent learns time-varying risk aversion and rebalancing frequency under changing market conditions. While effective in improving risk-adjusted performance, such approaches primarily rely on end-to-end policy learning and do not explicitly separate predictive signal extraction from allocation decisions.

A parallel strand of research explores how predictive signals can be translated directly into portfolio weights. Brandt et al. (2009) propose parametric portfolio policies that map firm characteristics to portfolio allocations, providing an early bridge between cross-sectional prediction and portfolio construction. More recent studies from Oyewola et al. (2026) apply

deep learning architectures, such as LSTM-based hybrid models, to learn nonlinear mappings from price dynamics to portfolio weights. While these approaches improve flexibility and predictive capacity, they typically embed prediction and allocation in a single layer, limiting interpretability and explicit risk control.

Despite recent progress, most DRL methods rely on end-to-end mappings from raw market features to portfolio weights. This structure combines prediction and decision-making in a single step, making policies difficult to interpret and hard to constrain using economic or risk-based principles. In addition, many end-to-end models do not explicitly account for tail risk, even though ML-based forecasts contain informative signals. These limitations motivate hybrid frameworks that separate signal generation from policy learning, as implemented in the hierarchical architecture proposed in this study.

2.3. Explainable AI and Interpretable Decision-Making

As ML and DRL models become more complex, interpretability has become an increasingly important concern, especially in financial applications where decisions have material economic and regulatory consequences. Black-box models provide limited insight into why specific portfolio actions are taken, which complicates risk oversight, model validation, and practical deployment. Explainable AI (XAI) methods address this issue by offering structured ways to decompose model outputs into interpretable components.

SHAP (Shapley Additive Explanations), proposed by Lundberg and Lee (2017), has become one of the most widely used XAI tools in finance. Grounded in cooperative game theory, SHAP provides locally accurate and consistent feature attributions, quantifying the marginal contribution of each input variable to a model prediction. This framework aligns naturally with factor-based thinking in asset pricing. Applying SHAP to ML return prediction models allows practitioners to identify whether predicted returns are driven by momentum, valuation, quality, macroeconomic factors, or other economic variables.

Recent studies have explored the direct integration of XAI methods into trading and portfolio management systems. de-la-Rica-Escudero et al. (2025) develop an explainable DRL framework in which SHAP and LIME are used to interpret the agent's actions in real time, helping to verify that portfolio decisions remain economically reasonable under changing market conditions. Sun et al. (2024) incorporate SHAP-based feature selection into a GraphSAGE–PPO combined trading model and show that filtering low contribution inputs improves both training efficiency and out-of-sample performance. Beyond prediction-level explanations, Shapley value concepts have also been applied to portfolio-level analysis to decompose risk and return contributions across assets, providing transparent measures of diversification and marginal risk exposure (Shalit, 2021). Industry practitioners have adopted similar approaches; for example, Man Group (2023) reports using SHAP to analyze proprietary ML trading strategies and finds that their behavior aligns with intuitive factor exposures. Evidence from dividend-based strategies by Y. Chen and Israelov (2024c) further highlights the importance of understanding underlying return drivers. While high-dividend stocks appear to outperform historically, their performance is fully explained by exposures to value, quality, and defensive factors, and dividend tilts reduce after-tax returns. This reinforces the need for interpretable, factor-level signal decomposition rather than reliance on surface-level characteristics, aligning naturally with explainable machine learning approaches.

The proposed framework builds on these developments by embedding interpretability directly into the learning architecture. Return forecasts from gradient-boosted tree models are paired with SHAP-based factor attributions and passed explicitly to the DRL agent, ensuring that portfolio decisions are based on transparent and economically meaningful signals. By integrating explainability into both prediction and decision-making, the hierar-

chical framework supports risk-aware portfolio construction, strengthens risk management, and facilitates practical deployment.

2.4. Literature Synthesis and Research Gap

The literature reviews above show clear progress in portfolio optimization, machine learning-based return prediction, deep reinforcement learning, and explainable AI. However, these areas of research have largely developed in isolation and have not yet been integrated into a unified framework for dynamic, risk-aware, and interpretable portfolio allocation.

Classical portfolio models provide strong theoretical foundations and effective tools for controlling downside risk and estimation uncertainty, but they typically rely on static parameters and fixed rebalancing rules. Machine learning models improve return prediction by capturing nonlinear and regime-dependent patterns, yet most studies focus on forecasting accuracy or apply simple trading rules, leaving the portfolio construction problem only partially addressed. Deep reinforcement learning offers a flexible framework for sequential decision-making, but most existing applications adopt end-to-end designs that are difficult to interpret and provide limited control over tail risk and trading frictions. At the same time, explainable AI methods such as SHAP have improved transparency, but are mainly used for post hoc analysis rather than as inputs to policy learning.

Overall, the existing literature lacks an approach that jointly combines nonlinear predictive modeling, explicit downside-risk-sensitive optimization, and interpretable economic signals within a single dynamic portfolio framework. This gap motivates the hierarchical signal-to-policy framework proposed in this study, which integrates machine learning, risk-aware reinforcement learning, and explainable AI as a trading signal in a coherent and practically implementable way.

3. Problem Setup and Proposed Methodology

Return forecasts play a central role in investment decision-making and portfolio construction. Building on this insight, this section introduces a hierarchical signal-to-policy learning framework designed for dynamic and risk-aware portfolio allocation. The framework begins with a supervised learning stage that estimates next-period asset returns and computes model-based explanatory indicators. These predictions and explanations form structured signals that are passed to a downstream reinforcement learning policy, which translates them into allocation decisions. The following subsections formalize the problem setting, describe each modeling component, and present the theoretical foundations for integrating these elements into a coherent signal-to-policy mapping. Figure 1 provides a high-level process flow overview of the proposed hierarchical signal-to-policy (HSTP) framework. This framework explicitly separates signal extraction from policy learning, i.e., predictive modeling and interpretability are handled upstream, while allocation decisions are optimized downstream under an explicit risk–return trade-off and real-world trading assumptions.

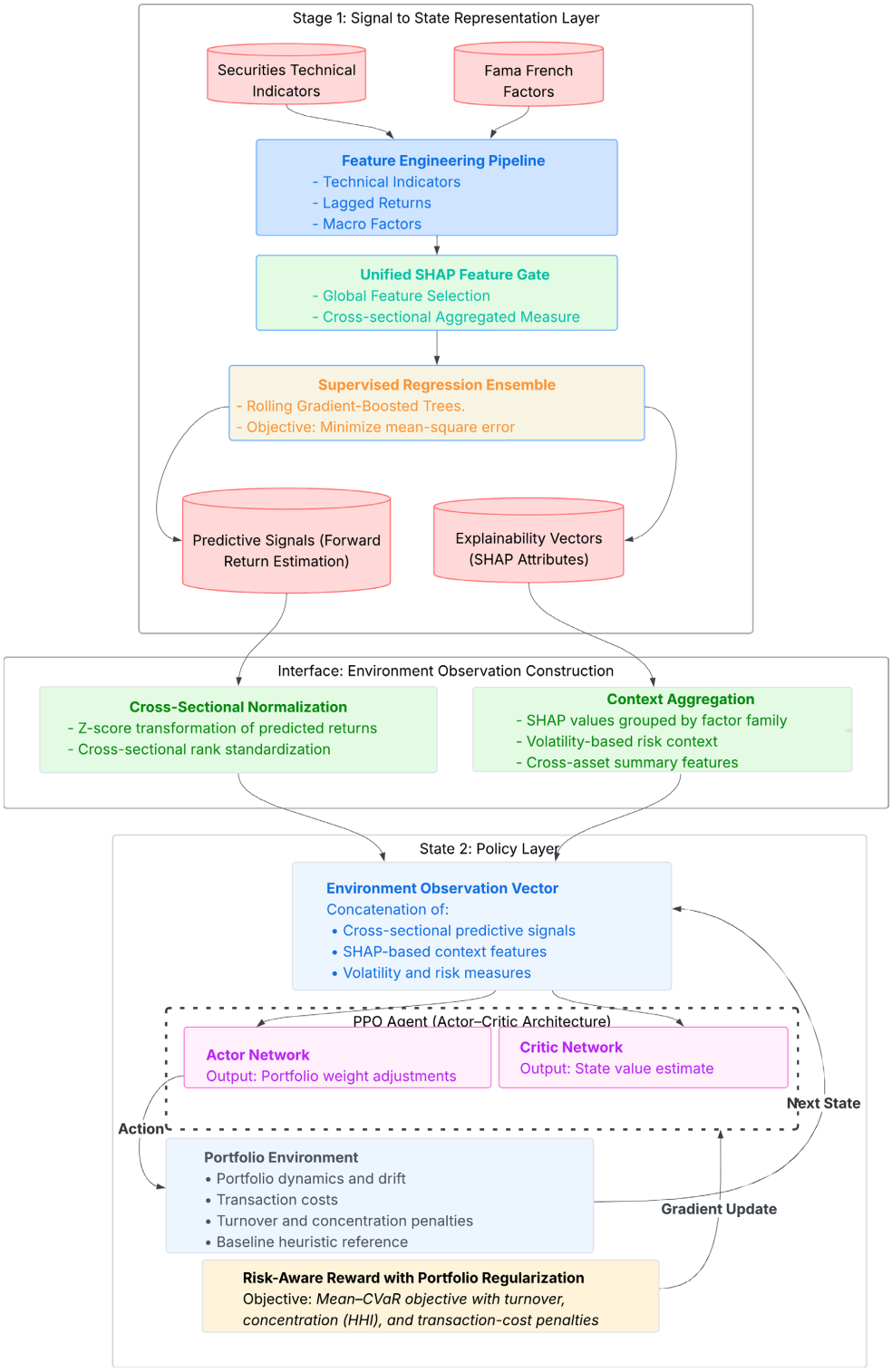


Figure 1. Hierarchical signal-to-policy (HSTP) framework for risk-aware portfolio optimization. Note: The figure summarizes the proposed two-stage hierarchical signal-to-policy (HSTP) framework. In Stage 1, asset-level technical indicators and macroeconomic factors are processed through a deterministic feature engineering pipeline and filtered using a unified SHAP-based feature gate, yielding predictive return signals and interpretable SHAP attributions. These outputs are transformed into the agent's observation space via cross-sectional normalization and context aggregation. In Stage 2, the resulting observation vector is consumed by a PPO actor–critic agent, which produces portfolio weights under realistic trading constraints and is trained using a risk-aware reward that balances expected return, downside risk (CVaR), and portfolio regularization penalties.

3.1. Multifactor Return Decomposition and Mean-Risk Portfolio Optimization

Let $r_{i,t+1}$ denote the simple return on asset i from time t to $t+1$, and let $r_{f,t+1}$ be the contemporaneous risk-free rate. In a general K-factor setting, the excess return of an asset can be written as

$$r_{i,t+1} - r_{f,t+1} = \alpha_i + \beta_i^\top f_{t+1} + \varepsilon_{i,t+1}, \tag{1}$$

where $f_{t+1} \in \mathbb{R}^K$ represents realizations of systematic factors, β_i denotes factor loadings, and $\varepsilon_{i,t+1}$ is an idiosyncratic risk with zero conditional mean. Under standard no-arbitrage conditions and efficient market assumptions, the pricing errors satisfy $\alpha_i \approx 0$, and expected excess returns reduce to

$$\mathbb{E}[r_{i,t+1} - r_{f,t+1}] = \beta_i^\top \lambda, \tag{2}$$

with λ representing factor risk premia, as proposed by Ross (1976).

A canonical empirical specification is the Fama–French three-factor model (Fama & French, 1993), which adds factors like the size (SMB) and value (HML) on top of market excess returns. Subsequent work (Fama & French, 2015) extends this to the well-known five-factor model, further adding profitability (RMW) and asset growth factor (CMA) of using conservative minus aggressive.

Therefore the five factors included in this setting are $f_{t+1} = (\mathrm{MKT}_{t+1}, \mathrm{SMB}_{t+1}, \mathrm{HML}_{t+1}, \mathrm{RMW}_{t+1}, \mathrm{CMA}_{t+1})$. For a portfolio with weight vector $w \in \mathbb{R}^N$, the excess portfolio return can be written as

$$r_{p,t+1} - r_{f,t+1} = w^\top (r_{t+1} - r_{f,t+1}\mathbf{1}) = \alpha_p + \beta_p^\top f_{t+1} + \varepsilon_{p,t+1}, \tag{3}$$

where the portfolio's factor exposure is $\beta_p = \sum_i w_i \beta_i$. This expression highlights the role of systematic factors in shaping both expected performance and risk. Classical portfolio theory (Markowitz, 1952) relies on the mean–variance structure of asset returns, a structure that is naturally shaped by underlying factor exposures as captured in multifactor asset-pricing models. In our framework, this factor decomposition provides the economic foundation for the predictive modelling stage. Fama–French factors and related macro-financial variables are selected as the Stage-1 feature set, ensuring that the supervised learning model is informed by exposures that have well-documented explanatory power in asset pricing. By incorporating factor-based signals into the prediction stage, the model relies on empirically validated drivers of return variation, ensuring that its forecasts reflect economically meaningful sources of systematic risk.

3.2. Stage 1: Supervised Return Prediction with Gradient-Boosted Trees

3.2.1. Overview of the XGBoost Prediction Model

Stage 1 uses supervised learning to predict next-period returns using gradient-boosted regression trees (Friedman, 2001; T. Chen & Guestrin, 2016). This approach was selected as a practical and well-established tree-based learner for noisy financial return prediction, supported by mature and stable SHAP implementations. Tree-based boosting methods are widely used in financial machine learning settings characterized by low signal-to-noise ratios and complex nonlinear interactions (Feng et al., 2020), and TreeSHAP provides a consistent and theoretically grounded feature attribution framework for boosted tree models. Importantly, the role of Stage 1 is not to maximize standalone predictive accuracy, but to extract a stable and structured signal representation that can be reliably consumed by the downstream reinforcement learning policy. Each investment asset is assigned to its own predictive model, but all models share a unified feature set consisting of investment instrument specific technical indicators and Fama–French-style factor returns. Gradient boosting (Friedman, 2001) builds an ensemble of predictive models by sequentially adding

decision trees, where each new tree is trained to correct the errors made by the previous ones under a chosen loss function. Let $f^{(M)}(x)$ denote the ensemble prediction after m boosting iterations. The model takes the form

$$f^{(M)}(x) = \sum_{m=1}^{M} \nu \, h^{(m)}(x), \tag{4}$$

where $h^{(m)}(\cdot)$ is a regression tree and $\nu \in (0, 1]$ is a shrinkage parameter controlling the learning rate. At each iteration, the newly added tree is trained to approximate the negative functional gradient of the loss evaluated at the current ensemble prediction. This stage-wise construction enables the model to flexibly capture nonlinearities, interaction effects, and regime-dependent patterns that are difficult to encode using linear or parametric specifications.

For each asset i, the final predictor takes the form

$$\hat{y}_{i,t} = f_i(X_{i,t}) = \sum_{k=1}^{K_i} f_{i,k}(X_{i,t}), \tag{5}$$

where each $f_{i,k}$ is a regression tree. A tree partitions the feature space into leaf regions and assigns a constant weight to each leaf. XGBoost estimates the ensemble by minimizing a regularized objective based on a second-order Taylor expansion of the loss function. The objective is:

$$\min_{\theta_i} \left\{ \sum_{t} \mathcal{L}(y_{i,t}, f_i(X_{i,t}; \theta_i)) + \sum_{k=1}^{K_i} \Omega(f_{i,k}) \right\}, \tag{6}$$

where the loss is the squared error,

$$\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2, \tag{7}$$

and the regularization penalty for tree $f_{i,k}$ is

$$\Omega(f_{i,k}) = \alpha \, T_k + \frac{\lambda}{2} \sum_{j=1}^{T_k} w_j^2, \tag{8}$$

Here T_k is the number of leaves in the tree, while α and λ control tree size and weight shrinkage. The use of both first-order and second-order derivatives allows XGBoost to evaluate split gains in an efficient and numerically stable manner, making it well-suited for noisy financial data with complex nonlinear patterns.

3.2.2. Model Training and Forward-Return Prediction

Let $r_{i,t+1}$ denote the simple return of asset i from day t to day $t+1$. For a forecasting horizon of H trading days, the supervised learning target is the cumulative simple return:

$$y_{i,t} = \prod_{h=1}^{H} (1 + r_{i,t+h}) - 1, \tag{9}$$

This label reflects the realized forward performance of the asset over the next H days and is used without risk adjustment.

For each date t, the feature vector $X_{i,t}$ contains only information available at time t, ensuring the no-look-ahead conditions. Prior to model training and prediction, all features are standardized to ensure features are placed on comparable scales.

Model hyperparameters, including tree depth, learning rate, subsampling, and regularization parameters (α, λ), are tuned by maximizing the Information Coefficient (IC):

$$\text{IC} = \text{Spearman}(\hat{y}_{i,t}, y_{i,t}), \tag{10}$$

This correlation measures whether the model correctly ranks assets by expected return. After tuning, the model is re-estimated using the combined training data and subsequently applied to generate out-of-sample predictions.

For each asset and date t, Stage 1 produces both the point forecast $\hat{y}_{i,t}$ and a cross-sectional percentile score that normalizes predictions to the unit interval. This score is computed as:

$$\text{Score}_{i,t} = \frac{\text{rank}(\hat{y}_{i,t}) - 1}{N - 1}, \tag{11}$$

where N is the number of assets. This rank (or percentile score) represents the cross-sectional ordering of one-day-ahead predicted returns across assets on each date, and is used as a relative, scale-free signal. The realized forward return $y_{i,t}$ is retained for out-of-sample evaluation of the Stage 1 predictive models. Throughout the paper, the predicted return produced by the Stage-1 model refers to the one-day-ahead return forecast for each asset, computed using information available up to the prediction date.

3.3. SHAP-Based Explainability and Signal Construction

While Stage 1 focuses on predicting returns based on relevant features for investment assets, it is important to extract structured information from the prediction model that can be used to construct informative and economically useful signals. This applies both for interpreting the economic meaning and for transforming model outputs into informative signals that can be used by the downstream allocation policy. Because our predictive models are generated by tree-based XGBoost ensembles, the associated feature attributions can be computed exactly using the TreeSHAP algorithm (Lundberg & Lee, 2017; Lundberg et al., 2020).

Shapley values were developed in cooperative game theory (Shapley, 1953), and provide a useful way to allocate a model's prediction across its input features. SHAP is adopted as the explainability mechanism for the Stage 1 prediction models due to its theoretical grounding and suitability for integration into a downstream learning pipeline. Unlike local surrogate approaches, such as LIME, which can exhibit instability across sampling realizations, SHAP provides additive, model-consistent feature attributions derived from cooperative game theory. Compared to permutation importance and partial dependence methods, SHAP produces observation-level attributions that capture nonlinear feature interactions and can be directly used as structured numerical inputs. These properties are essential in our framework, where explainability outputs are not used solely for interpretation, but serve as state representations for the Stage 2 reinforcement learning agent.

Beyond interpretability, these attributions can also be used as structured, economically meaningful signals, such as grouped technical and factor-level contributions and cross-sectionally demeaned "relative Shapley" score. These attribution-based signals are subsequently passed to the Stage 2 reinforcement learning agent, where they inform portfolio allocation decisions using economically grounded predictive signals derived from the return prediction models.

3.3.1. Shapley Value Decomposition and Group Attribution

For each investment asset i, prediction model f_i, and feature vector $X_{i,t}$, TreeSHAP yields the Shapley decomposition

$$\hat{y}_{i,t} = f_i(X_{i,t}) = \phi_{i,0} + \sum_{j=1}^{d} \phi_{i,j,t}, \tag{12}$$

where $\phi_{i,0}$ is the base value (the expected model output under a reference distribution), and $\phi_{i,j,t}$ is the Shapley value for feature j at time t. Each $\phi_{i,j,t}$ measures the marginal contribution of feature j to the deviation in the prediction from the base value. Because SHAP originates from cooperative game theory, this decomposition satisfies local accuracy, symmetry, and consistency, properties that are particularly important in financial applications where explanations must be numerically consistent to model outputs.

To obtain economically meaningful explanations, we aggregate SHAP values within feature families. Let $G^{\text{tech}}(s)$ denote the set of technical features associated with suffix s (e.g., all combinations of a 5-day simple moving average) and let $G^{\text{ff}}(f)$ denote the features associated with a Fama–French factor family f. We define group-level attributions as

$$\Phi^{\text{tech}}_{i,s,t} = \sum_{j \in G^{\text{tech}}(s)} \phi_{i,j,t}, \Phi^{\text{ff}}_{i,f,t} = \sum_{j \in G^{\text{ff}}(f)} \phi_{i,j,t}, \tag{13}$$

These quantities measure the total contribution of an entire technical indicator class or factor family to the predicted forward return at time t. Because of additivity, the grouped attributions will reconstruct the prediction:

$$\phi_{i,0} + \sum_{s} \Phi^{\text{tech}}_{i,s,t} + \sum_{f} \Phi^{\text{ff}}_{i,f,t} = \hat{y}_{i,t}, \tag{14}$$

This ensures that grouped attributions preserve the explanatory structure of the model.

3.3.2. Unified SHAP Feature Gate

Stage-1 models are trained on a broad set of candidate features, including technical indicators and Fama–French-style factors. Because not all features contribute equally to forecasting next-period returns, we apply a Unified SHAP Feature Gate on an initial base period T_{base} for Stage 1 models to identify a coherent and global feature set for subsequent training and prediction. This gate is estimated once and then fixed, and thus, no re-estimation in rolling windows is required.

On the base window, a lightweight XGBoost model g_i is fitted for each asset i using shallow trees to emphasize interpretability. TreeSHAP produces feature-level attributions $\phi^{\text{gate}}_{i,j,t}$. The mean absolute SHAP importance of feature j for asset i is defined as

$$I_i(j) = \mathbb{E}_{t \in T_{\text{base}}} \left[\left| \phi^{\text{gate}}_{i,j,t} \right| \right], \tag{15}$$

Feature importances are then aggregated across assets and feature families. To identify features that are robust across assets, we aggregate SHAP importances at the group level by taking the mean per feature set.

For a technical indicator group s, the cross-asset importance is

$$I(s) = \frac{1}{|U|} \sum_{i \in U} I_i(j(i,s)), \tag{16}$$

where U denotes the investment universe and $j(i,s)$ identifies the features for suffix s and asset i. For a Fama–French factor family f with feature set $F(f)$, importance is defined as:

$$I(f) = \frac{1}{|U_f|} \sum_{i \in U_f} \sum_{j \in F(f)} I_i(j), \tag{17}$$

where U_f denotes the subset of assets for which at least one feature in family f is present.

These group-level measures provide a data-driven ranking of technical and factor families. A small number of top-ranked groups is selected to form the final unified feature set; technical indicators are retained across all assets, whereas factor families are represented by the single lag with the highest mean importance. The broader asset-pricing literature shows that mechanically expanding the set of predictors without careful controls can lead to issues, as many proposed factors lack incremental explanatory power and may reflect false discoveries arising from data mining (Feng et al., 2020). Therefore, the proposed framework does not treat factor inclusion as a source of alpha by itself. Instead, SHAP-based feature gating is used to restrict the predictive model to a small, stable set of economically interpretable features that demonstrate consistent contributions across assets and over time. By fixing this feature set prior to policy learning, the framework mitigates overfitting of signals in the downstream allocation policy.

3.3.3. SHAP-Derived Metrics for Policy Learning

Although individual SHAP attributions provide detailed, asset-specific explanations for the Stage 1 return predictions, the reinforcement learning agent in Stage 2 requires a compact and stable set of explanatory signals, which operate on the cross-asset levels. To this end, we aggregate the SHAP attributions across the investment universe on each date and construct cross-sectional SHAP summary metrics that describe how important each feature group is on that day in an aggregated fashion.

Let $\Phi_{i,g,t}$ denote the group-level SHAP attribution (defined in Section 3.3.1) for asset i, feature group g, and date t. For each date, we compute the cross-sectional mean SHAP contribution of group g:

$$\text{Avg_SHAP}_{g,t} = \frac{1}{N_t} \sum_{i=1}^{N_t} \Phi_{i,g,t}, \tag{18}$$

where N_t is the number of assets with valid observations on date t. This statistic captures the directional contribution of group g to predicted returns at time t. A positive value indicates that, on average, the predictive model contributes to group g positively with expected returns across assets on that day, while a negative value indicates the opposite.

To quantify the degree of volatility in this directional contribution of group g across assets, we compute the cross-sectional standard deviation of its contributions:

$$\text{Std_SHAP}_{g,t} = \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} \left(\Phi_{i,g,t} - \text{Avg_SHAP}_{g,t} \right)^2}, \tag{19}$$

This measure reflects how fluctuated the feature group g is across universe. Larger values suggest that the SHAP values vary differently on that date, whereas smaller values indicate a more uniform impact.

These aggregated SHAP metrics serve two purposes. First, they summarize the global explanatory structure of the predictive model on each day, indicating which technical or Fama–French factor groups the model currently has higher contributions. Second, by compressing per-asset SHAP values into market-level attributes, the reinforcement learning

agent receives this as an informative signal about predictions for asset returns on the market level to avoid the instability and dimensionality issues using per-asset metrics.

In addition to these SHAP summaries, we construct complementary cross-sectional measures derived from Stage-1 predictions and realized volatility. For predicted returns $\hat{y}_{i,t}$, we compute their daily cross-sectional mean and standard deviation,

$$\text{CrossSec_Mean_PredRet}_t = \frac{1}{N_t}\sum_{i=1}^{N_t}\hat{y}_{i,t}, \tag{20}$$

$$\text{CrossSec_Std_PredRet}_t = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}(\hat{y}_{i,t} - \text{CrossSec_Mean_PredRet}_t)^2}, \tag{21}$$

Together, the SHAP-derived summary metrics and cross-sectional prediction statistics form a market-level contextual layer within the Stage-2 observation space. By incorporating both the prediction and interpretability signals from Stage 1, the reinforcement learning agent is endowed with a transparent and economically interpretable representation of the environment in which it makes allocation decisions.

3.4. Hierarchical Deep Neural Network Framework

The second stage of the proposed signal-to-policy architecture transforms the predictive outputs from Stage 1 into dynamic, risk-aware portfolio decisions. In this stage, a reinforcement learning (RL) agent, implemented using Proximal Policy Optimization (PPO), learns how to map multi-asset predicted returns, SHAP-derived explanatory signals, technical indicators, and volatility measures into actionable portfolio weights. The agent operates within a formally defined Markov Decision Process (MDP) in which actions specify weights tilt relative to a baseline portfolio. Baseline portfolios are generated from predicted signals while learning of weights tilt from baseline is guided by a mean–CVaR reward with turnover and concentration penalties to reflect practical risk-management considerations. Through iterative policy improvement, the PPO agent learns a stable mapping from interpretable model signals to feasible portfolio weights, completing the hierarchical pipeline from prediction to policy.

The overall hierarchy consists of two tightly coupled layers: Stage 1 (Signal-to-State) produces model-based predictive signals and explanations, including per-asset forward-return forecasts, SHAP attributions, and cross-sectional SHAP summary metrics, together with any normalized predictive scores. These outputs, along with the technical indicators and global factor inputs used to train Stage 1, are used as a unified time-stacked observations state for the next stage. The resulting state summarizes both what the predictive models expect (returns and their drivers) and how current market conditions have evolved (momentum, volatility, and macro-factor exposures). Stage 2 (Policy Learning) is a deep reinforcement learning agent implemented using Proximal Policy Optimization (PPO) that maps the state into portfolio tilts relative to a model-based baseline allocation. These tilts are then projected into feasible portfolio weights that satisfy long-only and fully invested constraints. The agent is trained to optimize a monthly mean–CVaR reward augmented with turnover and concentration penalties, yielding an explicitly risk-aware policy suitable for real-world implementations.

3.4.1. MDP Specification and Observation Design

The portfolio allocation problem is formulated as an MDP evolving over time. Let $\mathcal{U} = \{1, \ldots, N\}$ denote the investment universe and $r_{i,t+1}$ denote the simple return of asset i between t and $t+1$. The portfolio at time t is represented by a weight vector w_t satisfying the standard fully invested constraint $\mathbf{1}^\top \mathrm{w}_t = 1$.

The state observed by the agent consolidates the relevant information produced in Stage 1, together with recent asset-price-related metrics. To avoid the look-ahead bias, all components of the state are based only on data available up to time t. The state is constructed as a rolling window

$$s_t = [\phi_{t-L+1}, \ldots, \phi_t], \tag{22}$$

where each cross-sectional snapshot ϕ_τ includes the Stage-1 predicted returns $\hat{y}_{i,\tau}$, their rank-normalized scores $\hat{s}_{i,\tau}$, grouped SHAP attributions $g_{i,\tau}^{(k)}$, and cross-sectional SHAP summary measures that depict the explanatory structure of the predictive model on that date. These quantities are complemented by historical volatility, technical indicators and Fama–French factors selected through the unified feature gate. Together, these features form a compact yet meaningful representation of prevailing market conditions to guide the agent's decisions.

Given the observed state s_t, the agent selects an action $a_t \in [-1, 1]^N$, interpreted as a vector of tilts relative to a baseline portfolio defined in the following Section 3.4.2. After the action is applied, the portfolio is then invested under a buy-and-hold strategy passively between rebalancing dates.

$$\tilde{\mathbf{w}}_{t+1} = \frac{\mathbf{w}_t \odot (1 + \mathbf{r}_{t+1})}{1 + \mathbf{w}_t^\top \mathbf{r}_{t+1}}, \tag{23}$$

where $\tilde{\mathbf{w}}_{t+1}$ is the next period instrument weight for all investments growing with return of $\mathbf{r}_{t+1}$ and normalized based on total portfolio return $1 + \mathbf{w}_t^\top \mathbf{r}_{t+1}$. Reward feedback follows after the next period, reflecting the cumulative effect of the portfolio including that period. A standard discount factor $\gamma \in (0, 1)$ is applied at the decision frequency.

3.4.2. Baseline Portfolio and Action Parameterization

Rather than learning absolute portfolio weights directly, the reinforcement learning agent operates relative to a signal-driven baseline portfolio constructed from Stage-1 predictive information. This formulation anchors the policy to an economically interpretable reference allocation and constrains learning to relative deviations, improving feasibility and stability.

For each asset i at rebalance date t, a predictive signal is formed by combining short- and medium-horizon return forecasts generated in Stage 1. When daily predictions are available, the composite signal is defined as

$$s_{i,t} = 0.7\, \hat{r}_{i,t}^{(1)} + 0.3\, \hat{r}_{i,t}^{(21)}, \tag{24}$$

where $\hat{r}_{i,t}^{(1)}$ denotes the next-day predicted return and $\hat{r}_{i,t}^{(21)}$ denotes historical rolling 21-day averages of predicted returns. This helps balance short-term responsiveness with smoother trend information, reducing sensitivity to one-off forecast noise.

To account for cross-asset risk heterogeneity, the predictive signal is standardized cross-sectionally for a relative signal strength within the universe,

$$z_{i,t} = \frac{s_{i,t} - \bar{s}_t}{\mathrm{std}(s_t)}, \tag{25}$$

where $\bar{s}_t$ and $\mathrm{std}(s_t)$ denote the signal mean and standard deviation across all assets. Therefore, the baseline portfolio is constructed as a linear deviation from the Equal-Weight allocation,

$$\tilde{w}_{i,t}^{\text{base}} = \frac{1}{N} + \beta\, z_{i,t}, \tag{26}$$

where N denotes the number of assets and β controls the strength of this signal tilt. The raw baseline weights are then applied to the transformation $\Pi_{\mathcal{W}}(\cdot)$, which makes the baseline weights long-only, with no shorting, no leverage, and satisfying asset-level constraints.

$$w_t^{\text{base}} = \Pi_{\mathcal{W}}(\widetilde{w}_t^{\text{base}}), \tag{27}$$

At each rebalance date, the PPO agent outputs an action vector $u_t \in \mathbb{R}^N$, which is mapped to a bounded tilt through a temperature-scaled hyperbolic tangent transformation.

$$a_{i,t} = \tanh\left(\frac{u_{i,t}}{T_a}\right), \tag{28}$$

where T_a is the temperature for the effective strength of this action. The action is interpreted as a relative adjustment to the baseline portfolio, and the resulting tilted weights are projected back onto the investable set through $\Pi_{\mathcal{W}}(\cdot)$:

$$w_t = \Pi_{\mathcal{W}}\left(w_{i,t}^{\text{base}} \odot (1 + a_t)\right), \tag{29}$$

Overall, this baseline-relative structure captures the signal from Stage 1 predictive modeling from Stage 1 and then leaves the residual signal from Stage 1 result as well as factors in the observation space to the policy-based PPO agent for tilts around the baseline weights. By learning adjustments rather than absolute weights, the agent can focus on tilting the portfolio to better manage risk–return trade-offs via reward function design.

3.4.3. Risk-Aware Reward Function Design

The reward is designed to align the agent's learning objective with the classical mean-risk portfolio allocation target. Let $R_{t:t'}$ denote the compounded portfolio return realized between two consecutive decision points t and t'. Using historical losses, we compute empirical Value-at-Risk and Conditional Value-at-Risk at confidence level α and loss level L,

$$\text{VaR}_\alpha = \inf\{\updownarrow : \mathbb{P}(L \leq \updownarrow) \geq 1 - \alpha\}, \tag{30}$$

$$\text{CVaR}_\alpha = \mathbb{E}[L \mid L \geq \text{VaR}_\alpha]., \tag{31}$$

The mean–CVaR trade-off is captured by the baseline risk-adjusted reward,

$$\widetilde{r}_t = R_{t:t'} - \lambda_{\text{cvar}}\,\text{CVaR}_\alpha, \tag{32}$$

To ensure that the resulting allocation policy remains practical, we include additional penalties for portfolio concentration, turnover, and transaction costs. Turnover is defined as the absolute change in portfolio weights between rebalancing dates (accounting for both buys and sells). Concentration is quantified using the Herfindahl–Hirschman Index (HHI), defined as the sum of squared portfolio weights. Each penalty term has its own coefficient, i.e., λ_{hhi} for concentration, λ_{to} for turnover, and c_{tc} for transaction costs, which are incorporated into the reward function defined below.

Incorporating downside risk, turnover, and concentration penalties directly into portfolio optimization objectives are well established in the literature and widely used in

regulated investment settings (Mugerman et al., 2019). In this study, these elements are adopted as standard control mechanisms to ensure realistic and stable policy learning.

$$\begin{aligned} \mathrm{TO}_t &= \sum_{i=1}^{N} \left| w_{i,t} - w_{i,t^-} \right| \\ \mathrm{HHI}_t &= \sum_{i=1}^{N} w_{i,t}^2 \\ r_t &= \widetilde{r}_t - \lambda_{\mathrm{hhi}} \mathrm{HHI}_t - (\lambda_{\mathrm{to}} + c_{\mathrm{tc}}) \mathrm{TO}_t \end{aligned} \tag{33}$$

3.4.4. Policy Network Architecture and PPO Training

The Stage-2 policy is implemented using a deep neural network trained with the Proximal Policy Optimization (PPO) algorithm. The policy network maps the high-dimensional state s_t to the parameters of a stochastic action distribution, and a separate value function network, sharing the same architecture, estimates the state value $V_\psi(s_t)$ used for advantage estimation.

The PPO update is done based on the clipped surrogate objective, which constrains the magnitude of policy updates to enhance stability. Let $\rho_t(\theta) = \frac{\pi_\theta(u_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(u_t|s_t)}$ denote the likelihood ratio of the new and old policies. The objective is defined as below:

$$\mathcal{L}^{\mathrm{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(\rho_t(\theta)\,\hat{A}_t,\ \mathrm{clip}(\rho_t(\theta),\,1-\epsilon,\,1+\epsilon)\,\hat{A}_t\right)\right], \tag{34}$$

where $\hat{A}_t$ is the generalized advantage estimate. This formulation prevents large step updates within policy space, and this structure would fit well in investment environments as return series have been seen as noisy and non-stationary data.

The overall learning objective extends the clipped surrogate with a squared-error value-function loss and an entropy regularization term, which encourages exploration. Training is performed on rolling windows of data, similar to the Stage-1 protocol. Each window contains a training set used for gradient updates, followed by a validation segment used for hyperparameter tuning. The resulting policy is evaluated out of sample. Because PPO training is stochastic, we repeat the Stage 2 training with multiple random initializations and summarize performance across runs to assess robustness.

4. Empirical Experiments

This section presents the empirical evaluation of the proposed hierarchical signal-to-policy learning framework. The objective is to assess the model's performance in realistic portfolio allocation settings, focusing on both prediction effectiveness and risk-adjusted investment outcomes generated by the learned allocation policies. We apply the framework to two distinct investment universes that differ in asset composition and market structures. For each universe, we use a consistent model configuration to train the framework, generate portfolio action weights, and construct an investable portfolio over subsequent periods. Corresponding benchmark strategies are defined under the same rebalancing frequency and portfolio constraints. Out-of-sample performance is evaluated using a rolling-window protocol to emulate realistic deployment conditions. To assess the robustness and stability of the learned policy network, the entire training–prediction–evaluation procedure is repeated multiple times, producing an ensemble of action–weight trajectories and performance outcomes.

4.1. Data Collection

We train and evaluate the hierarchical signal-to-policy learning framework on two equity universes commonly used in real-world portfolio management. The first universe consists of nine Select Sector SPDR ETFs, which include Materials Select Sector SPDR Fund (XLB), Energy Select Sector SPDR Fund (XLE), Financial Select Sector SPDR Fund (XLF),

Industrial Select Sector SPDR Fund (XLI), Technology Select Sector SPDR Fund (XLK), Consumer Staples Select Sector SPDR Fund (XLP), Consumer Discretionary Select Sector SPDR Fund (XLY), Health Care Select Sector SPDR Fund (XLV), and Utilities Select Sector SPDR Fund (XLU). Those ETFs collectively cover the major U.S. stock market sectors. The second universe contains twenty-six large-cap U.S. equities drawn from the Dow Jones Industrial Average index (DJIA). We use the passive index tracking ETF (ticker DIA) to represent this index for the benchmark. These stocks broadly represent the cross-industry composition of the Dow Jones Industrial Average index over the sample period, while excluding a small number of names with incomplete or inconsistent price histories.

Daily close prices and returns are collected from January 1999 to November 2024 to ensure coverage across multiple market cycles. All asset returns used in Stage 1 are trained and predicted based on the next one-day forward-looking period, while Stage 2 operates at a monthly rebalancing frequency, with portfolio rebalancing and reward evaluation every 21 trading days. This design allows short-horizon predictive signals to be aggregated and acted upon at a lower-frequency portfolio decision level.

To prevent lookahead bias, all features are computed using only information available up to each observation date. Forward-return labels that cannot be fully observed due to sample boundaries are removed through label-availability filtering, ensuring that neither training nor evaluation has any information leakage. Rolling training, validation, and out-of-sample test partitions are then formed using a rolling-forward procedure, with strict sequential ordering.

Each asset is represented by a rich feature set combining asset-specific technical indicators. Those include moving-average signals, momentum oscillators, lagged returns, and volatility measures, as well as shared Fama–French factors with different lags, such as market beta, size, value, profitability and investment. All features are computed at a daily frequency and aligned across assets. Appendix A summarizes the feature definition for Stage 1. Different factors selected by the Stage 1 SHAP gate will be included in the next stage policy learning.

4.2. Model Architecture and Configuration

The empirical evaluation is conducted using the proposed hierarchical signal-to-policy learning framework, which explicitly decouples return prediction (Stage 1) from portfolio optimization (Stage 2). This separation introduces a modular structure that enables independent analysis of the predictive signal layer and contributes to more stable learning dynamics using a reinforcement learning framework. In short, Stage 1 produces predictive signals at a daily frequency, whereas Stage 2 aggregates these signals to inform portfolio allocation decisions made on a monthly basis under explicit risk controls.

4.2.1. Stage 1: Predictive Modeling and SHAP-Based Signal Construction

Stage 1 trains a set of asset-specific XGBoost models with one model per ETF or stock from the investment universe. Each model predicts the one-day forward simple return:

$$\hat{r}_{t,i}^{(1)} = f_{\theta_i}(X_{t,i}), \tag{35}$$

where $X_{t,i}$ is the feature vector for asset i on trading day t, and f_{θ_i} is the corresponding gradient-boosted tree.

The feature set includes two components, i.e., asset-specific technical indicators including moving-average measures, momentum signals, lagged returns, and volatility, as well as Fama–French factors with daily lags shared across assets. To ensure feature selection stability, this list of features is first filtered using a Unified SHAP Feature Gate, which fits preliminary XGBoost models on a designated base window, computes TreeSHAP impor-

tances, and aggregates them across assets. We selected the top 10 feature groups using the first period data, which had consistently strong predictive relevance at the universe level for subsequent rolling training and prediction.

Models are trained using the top 10 selected features from Unified SHAP Feature Gate with rolling-forward splits to generate predicted forward-return estimations. Because the target uses information at $t+1$, the pipeline excludes observations whose forward labels are not fully realized.

Hyperparameters are selected to maximize the validation information coefficient (IC):

$$\mathrm{IC} = \rho\left(\hat{r}_{t,i}^{(1)},\, r_{t,i}^{(1)}\right), \tag{36}$$

where $r_{t,i}^{(1)}$ denotes the realized one-day forward return and $\rho(\cdot,\cdot)$ is the cross-sectional correlation operator. For each trading day, Stage 1 produces several outputs that are passed to Stage 2. In addition to the point forecast $\hat{r}_{t,i}^{(1)}$, the model generates a cross-sectional percentile score $s_{t,i} = \frac{\mathrm{rank}(\hat{r}_{t,i}^{(1)})}{N}$, where N is the number of assets, as well as SHAP attributions aggregated at the feature-group level according to $\hat{r}_{t,i}^{(1)} = \phi_{t,i}^{\mathrm{base}} + \sum_{g\in G} \phi_{t,i}^{(g)}$.

Although the supervised learning target is one-day-ahead, portfolio decisions in Stage 2 are made at a 21-trading-day rebalancing frequency. To bridge this horizon mismatch and improve signal robustness, the daily Stage-1 predictions are further transformed into a medium-horizon signal by blending the one-day forecast with its rolling 21-day average, as detailed in Section 3.4.2. This aggregation reduces high-frequency noise while preserving responsiveness to new information. The signals passed from Stage 1 to the policy-learning stage consist of the smoothed predictive signal, daily return forecasts, cross-sectional rankings, and SHAP-based group attributions.

4.2.2. Stage 2: PPO-Based Portfolio Allocation

Stage 2 converts the daily predictive signals produced in Stage 1 into portfolio allocations using a Proximal Policy Optimization (PPO) agent. While the environment evolves at a daily frequency, portfolio rebalancing occurs every 21 trading days, corresponding to a monthly decision horizon. Rewards are issued only at rebalance dates and summarize portfolio performance over the preceding period.

- State Representation

At each rebalance time t, the agent observes a state vector constructed from the Stage-2 observation feature set described in Appendix B. The state combines asset-level predictive signals with cross-sectionally aggregated explanatory and market-context features, all computed using information available up to time t. The observation state can be written as,

$$\mathbf{s}_t = \left[\underbrace{\left\{\hat{r}_{t,i}^{(1)},\ \phi\left(r_{t,i}^{(1)}\right), \mathrm{rank}\left(\hat{r}_{t,i}^{(1)}\right)\right\}_{i=1}^{N}}_{\text{asset-level Stage 1 outputs}},\ \underbrace{\overline{\phi}_t,\ \widetilde{\phi}_t,\ \mu_t^{\hat{r}},\ sd_t^{\hat{r}}, \mu_t^{\sigma}}_{\text{Stage 1 cross-sectional aggregates}},\ \underbrace{F_t, \left\{\mathrm{X}_{t,i}^{\mathrm{tech}},\ \mathrm{X}_{t,i}^{\mathrm{mom}}\right\}_{i=1}^{N}}_{\text{Stage 1 inputs}} \right], \tag{37}$$

where N denotes the number of assets in the investment universe.

The first block consists of the asset-level outputs of Stage 1: the one-day-ahead return forecasts $\hat{r}_{t,i}^{(1)}$, their historical volatilities and their cross-sectional ranks. These quantities preserve prediction and their relative ordering of expected returns across assets.

The second block contains cross-sectionally aggregated quantities already produced in Stage 1, including SHAP feature-group summaries and cross-sectional context variables. Specifically, $\overline{\phi}_t$ denotes the vector of cross-sectionally normalized (relative) SHAP attributions across the feature groups selected by the Unified SHAP Feature Gate, corresponding

to the variables *_REL in Appendix B. The quantity $\widetilde{\overline{\phi}}_t$ represents the dispersion of SHAP attributions across assets for each feature group, corresponding to the variables Std_SHAP_*. These quantities summarize the explanatory structure of the Stage-1 predictive models at the universe level by indicating which feature groups drive forecasts and how heterogeneous their influence is across assets. In addition, the observation includes cross-sectional context variables derived from Stage-1 outputs, namely the mean and standard deviation of predicted returns across assets $\left(\mu_t^{\hat{r}},\ sd_t^{\hat{r}}\right)$ and the cross-sectional mean of asset-level volatility $\left(\mu_t^{\sigma}\right)$. Together, these aggregated variables describe the overall strength and dispersion of predictive signals and prevailing risk conditions across the universe at time t.

The final block reuses a subset of the Stage-1 input features themselves, i.e., Fama–French factors F_t and asset-level technical and momentum indicators $\left\{X_{t,i}^{\text{tech}}, X_{t,i}^{\text{mom}}\right\}_{i=1}^{N}$ selected by the Unified SHAP Gate. These features provide additional context on market conditions and recent price dynamics to be observed by the Stage 2 agents.

In addition, the observation includes cross-sectional context variables, such as the mean and standard deviation of predicted returns and volatility across assets. The complete state vector is stacked over a fixed lookback window prior to each rebalance, allowing the policy to condition decisions on recent temporal dynamics at both the asset and market levels. Given this representation, the PPO policy learns to generate portfolio tilts relative to a model-based baseline allocation, which are subsequently mapped into feasible portfolio weights subject to long-only, no shorting and other diversification constraints.

- Baseline Portfolio Construction

Portfolio actions are optimal weights defined as tilted weights from a static baseline allocation constructed from the predictive signals generated in Stage 1. The purpose of the baseline is to provide a stable, model-driven reference portfolio, allowing the reinforcement learning agent to focus on relative tilts rather than learning absolute allocations from scratch.

At each rebalance date t, similar to Section 3.4.2, predictions are further transformed into a medium-horizon signal by blending the one-day forecast with its rolling 21-day average. Let $\bar{r}_t$ and $std(\hat{r}_t)$ denote the cross-sectional average and standard deviation of the predicted returns at time t. These standardized scores $z_{t,i}$ are then mapped to preliminary baseline weights through a linear tilt around the equal weight portfolio with fixed intensity $\beta = 0.15$,

$$\tilde{w}_{i,t}^{\text{base}} = \frac{1}{N} + 0.15\, z_{t,i}, \tag{38}$$

The above raw weights are then mapped onto a long-only, fully invested feasible set using transformation of $\Pi_{\mathcal{W}}(\cdot)$, with individual asset bounds $w_{t,i} \in [0.01, \frac{1}{3}]$ and summing to 100% $\sum_{i=1}^{N} w_i = 1$. This projection yields the baseline portfolio w_t^{base}.

- Action Mapping and Portfolio Dynamics

At each rebalance date, the PPO policy outputs an action vector u_t. To control the magnitude of portfolio adjustments and improve numerical stability, the raw action is transformed using a temperature-scaled hyperbolic tangent function,

$$a_t = \tanh\left(\frac{u_t}{\tau}\right), \quad \tau = 0.4, \tag{39}$$

The transformed action weights are interpreted as relative tilts around the baseline portfolio generated from the predictive signals in Stage 1, and the resulting tilted weights are projected back onto the feasible set to enforce long-only and budget constraints,

$$w_t = \Pi_{\mathcal{W}}\left(w_t^{\text{base}} \odot (1 + a_t)\right), \tag{40}$$

Between rebalance dates, the portfolio evolves passively according to standard self-financing dynamics,

$$w_{t+1} = \frac{w_t \odot (1 + r_{t+1})}{1 + w_t^\top r_{t+1}}, \tag{41}$$

where r_{t+1} denotes the vector of realized asset returns. This is the portfolio-drifted weight during non-rebalancing dates.

- Reward Function

At each rebalance date, the agent receives a reward designed to balance return generation, downside risk control, trading costs, and diversification. The reward is specified as

$$R_t = r_t^{(21)} - \lambda_{\text{CVaR}} \, \text{CVaR}_{0.05}(L_{1:t}) - \lambda_{\text{HHI}} \sum_{i=1}^{N} w_{t,i}^2 - (\lambda_{\text{to}} + c_{\text{tc}})\text{TO}_t, \tag{42}$$

where $r_t^{(21)}$ denotes the realized 21-day portfolio return and $L_{1:t}$ represents the historical portfolio loss sequence used to compute Conditional Value-at-Risk (CvaR). CVaR is computed from empirical loss distribution after a 12-month warm-up period. The penalty terms reduce the likelihood of excessive trading, portfolio concentration (measured by the Herfindahl–Hirschman Index), and proportional transaction costs.

Together, this reward structure encourages exploitation of predictive signals while explicitly controlling tail risk and implementation frictions, yielding a policy that is both risk-aware and practically implementable.

4.2.3. Rolling-Window Training, Evaluation, and Prediction

We adopt a rolling forward training, evaluation, and prediction protocol to emulate realistic model deployment and to ensure a strictly out-of-sample process across both stages of the hierarchical framework. The full dataset spans January 1999 to November 2024. Stage-1 predictive models are trained using rolling windows consisting of a 12-year training period, followed by a 1-year validation period and a 1-year prediction period. Stage-2 policy learning is conducted using rolling windows consisting of 7 years of training data, followed by 1 year of validation and 1 year of out-of-sample testing. Portfolio performance reported in Sections 4.4–4.6 corresponds exclusively to the aggregated out-of-sample test periods from these rolling evaluations. All cumulative return figures display only the out-of-sample performance intervals.

In Stage 1, asset-specific supervised learning models are used to generate one-day-ahead return forecasts. For each rolling window, the data are divided into a 12-year training period, a 1-year validation period, and a 1-year prediction period. Within each window, several candidate models are trained using only the training data under different hyperparameter settings. Early stopping is applied during training, and model selection is based on performance on the validation set.

After the best-performing model is identified, it is fixed and applied directly to the prediction period without further adjustment. This model produces next-day return forecasts together with the corresponding SHAP attributions. The above procedure is implemented sequentially over the entire sample, resulting in a single, consistent time series of predicted returns and SHAP-based signals. These outputs are then treated as fixed inputs for the downstream policy-learning stage.

In Stage 2, portfolio allocation is learned using a Proximal Policy Optimization (PPO) agent. To evaluate robustness and sensitivity to stochastic learning dynamics, the full rolling training–validation–testing procedure is repeated over 30 iterations over the entire data period. In each iteration, the environment is constructed using trading-day windows

consisting of 7 years (252 × 7 trading days) of training, followed by a 1-year (252 trading days) validation window and a 1-year (252 trading days) out-of-sample testing window.

The agent observes a rolling lookback window of 21 trading days and rebalances the portfolio every 21 trading days, corresponding to a monthly decision frequency. The PPO agent is trained to observe daily state information and take actions and receive rewards at monthly intervals. This design reduces reward noise and aligns the learning problem with the portfolio rebalancing horizon.

PPO hyperparameters, including the learning rate, clipping range, CVaR risk-aversion coefficient, turnover penalty, and concentration (HHI) penalty, are selected via short PPO training runs fitted on the training window and evaluated on the validation window. Hyperparameter selection is performed only once using the first rolling period, and the chosen configuration is then held fixed throughout the entire reallocation process. To assess whether fixing PPO hyperparameters introduces regime dependence or temporal bias, Section 4.5.3 conducts a structured sensitivity analysis over key parameters while holding the PPO architecture and optimization settings fixed. Candidate configurations are ranked by their average validation reward,

$$\overline{R} = \frac{1}{T_{\text{val}}} \sum_{t \in \text{val}} R_t, \tag{43}$$

Using the selected hyperparameters, the PPO agent is trained on the training window through incremental updates. During training, performance is evaluated repeatedly on the validation window, and early stopping is applied to avoid overfitting. After training, the PPO policy is fixed and applied to the out-of-sample test window without further parameter updates. The portfolio is rebalanced every 21 trading days and remains drifted between rebalancing dates. The corresponding wealth W_{t+1} evolves according to

$$W_{t+1} = W_t(1 + r_t^{(21)}), \tag{44}$$

where $r_t^{(21)}$ denotes the realized portfolio return over the 21-day holding period.

To evaluate the performance of this model, we compute standard performance metrics, including compound annual growth rate (CAGR), annualized volatility, return over volatility ratio, Sortino ratio, maximum drawdown, and CVaR at the 5% level. This evaluation is repeated over 30 rolling PPO iterations, each corresponding to a distinct training–validation–testing window and an independent policy training run. We report both aggregate statistics (e.g., mean or median across runs) and run-level win ratios to quantify the stability of the learned allocation policy. The collection of out-of-sample results across random initializations provides evidence on robustness and parameter sensitivity under realistic stochastic training dynamics.

Figure 2 provides a workflow overview of the rolling training, validation, and out-of-sample testing design, which is applied consistently across both stages.

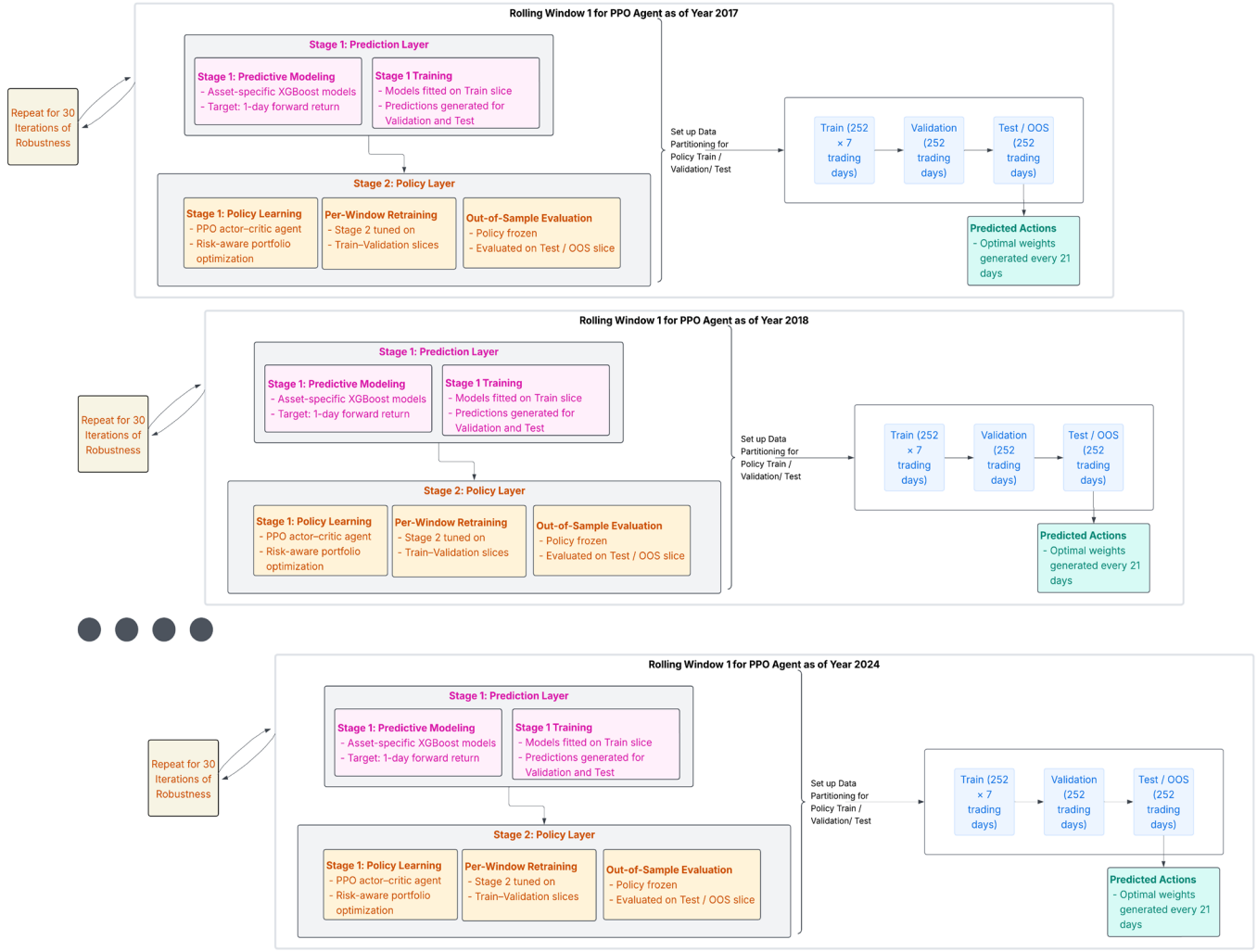


Figure 2. Rolling out-of-sample evaluation protocol. Note: This figure illustrates the rolling walk-forward evaluation and information flow of the proposed framework. At each rebalance time t, Stage 1 generates asset-level predictive outputs $\{\hat{r}_{t,i}^{(1)},\ \phi\left(r_{t,i}^{(1)}\right), \mathrm{rank}(\hat{r}_{t,i}^{(1)})\}_{i=1}^{N}$. and cross-sectional aggregates $(\bar{\phi}_t,\ \tilde{\phi}_t,\ \mu_t^{\hat{r}},\ sd_t^{\hat{r}}, \mu_t^{\sigma})$ using only data available up to time t. These quantities, together with the corresponding Stage-1 input features $\{F_t, X_{t,i}^{\mathrm{tech}}, X_{t,i}^{\mathrm{mom}}\}_{i=1}^{N}$., form the observation state s_t in Equation (37). Stage 2 trains, validates, and evaluates the PPO policy on rolling windows using only $\{s_\tau : \tau \le t\}$. Stage-1 models are not updated during Stage-2 learning, ensuring strictly out-of-sample evaluation and no information leakage.

4.3. Model Benchmark Definition

To evaluate the efficacy of the proposed hierarchical signal-to-policy framework, we compare out-of-sample performance with three different benchmarks. All benchmarks are evaluated under the same experimental setting as the HSTP model, including identical rolling windows, a 21-trading-day rebalancing frequency, and long-only, fully invested portfolio constraints.

The Equal-Weight benchmark allocates an identical fraction of capital to each asset at every rebalance date,

$$w_i^{\mathrm{EW}} = \frac{1}{N}, \quad i = 1, \ldots, N, \tag{45}$$

where N denotes the number of assets in the investment universe. This strategy rebalances the portfolio back to Equal Weight on the same day when the PPO agent makes the allocation changes.

We additionally include a mean–CVaR optimization benchmark to isolate the contribution of hierarchical policy learning from explicit risk-aware portfolio optimization. This benchmark constructs portfolios by directly solving a static optimization problem that mirrors the risk–return trade-off embedded in the PPO reward function, but without reinforcement learning. At each rebalance date t, portfolio weights are solved from a mean–CVaR optimization problem using the previous 252 trading days of realized daily returns. To ensure horizon consistency with the proposed HSTP model, the mean–CVaR portfolio is rebalanced every 21 trading days. Expected returns are calculated as historical means over this rolling 252-day window, while downside risk is measured using 5% Conditional Value-at-Risk (CVaR) at the same confidence level as in the Stage-2 reward function. The benchmark objective is defined to be consistent with the PPO reward formulation in Equation (42).

Specifically, at each rebalance date t, the mean–CVaR benchmark solves:

$$\begin{gathered}\mathbb{E}\left(r_t^{(21)}\right) - \lambda_{\text{CVaR}}\,\text{CVaR}_{0.05}(L_{1:t}) - \lambda_{\text{HHI}}\sum_{i=1}^{N} w_{t,i}^2 - (\lambda_{\text{to}} + c_{\text{tc}})\text{TO}_t \\ \textit{subject to}\ \sum_{\text{i}=1}^{\text{N}} \text{w}_{\text{t,i}} = 1, \text{w}_{\text{t,i}} \geq 0, \text{w}_{\text{t,i}} \in [\text{w}_{\text{min}}, \text{w}_{\text{max}}]\end{gathered} \tag{46}$$

where $r_{p,t}^{(21)}$ denotes the realized 21-trading-day portfolio return, $L_{1:t}$ represents the historical portfolio loss sequence used to compute CVaR, and TO_t denotes portfolio turnover at rebalance date t. The HHI term penalizes portfolio concentration, while the turnover and transaction cost terms penalize excessive trading, exactly as specified in the PPO reward function.

Importantly, the penalty coefficients λ_{CVaR}, λ_{HHI}, λ_{to}, and c_{tc} are fixed at the optimal values learned by the PPO agent during Stage-2 training. This ensures that the mean–CVaR benchmark reflects the same risk preferences and implementation frictions as the HSTP framework, while replacing the sequential policy-learning component for allocation weights with an optimization approach.

To ensure a fair comparison with learning-based strategies, the mean–CVaR optimization is repeated across the same 30 rolling out-of-sample iterations used for HSTP and momentum-based PPO portfolios, producing a distribution of run-level performance outcomes that can be directly compared under identical market realizations.

The primary active benchmark is a momentum-based PPO strategy, denoted PPO–MOM. This benchmark removes Stage 1 entirely and replaces model-based return forecasts with a momentum-only signal constructed from historical price dynamics. All Stage-2 components, including the PPO algorithm, reward formulation, portfolio constraints, rebalancing schedule, and rolling evaluation protocol, are kept identical to those of the proposed model. As a result, differences in performance can be attributed directly to including or excluding hierarchical signal generation.

For each asset i at time t, momentum signals are computed over multiple horizons,

$$\text{Mom}_{t,i}(d) = \frac{P_t - P_{t-d}}{P_{t-d}}, d \in \{5, 10, 20\}, \tag{47}$$

together with a 20-day rolling volatility measure,

$$\text{Vol}_{t,i}(20) = \frac{1}{20}\sum_{j=1}^{20}(r_{t-j,i} - \bar{r}_{t,i})^2, \tag{48}$$

Each component is standardized cross-sectionally across assets on each date. A daily composite momentum score is then constructed by averaging the standardized momentum signals and penalizing volatility,

$$S_{t,i}^{\text{daily}} = \tfrac{1}{4}(Z_t(\text{Mom}_{t,i}(5)) + Z_t(\text{Mom}_{t,i}(10)) + Z_t(\text{Mom}_{t,i}(20)) - Z_t(\text{Vol}_{t,i}(20))), \tag{49}$$

where $Z_t(\cdot)$ denotes cross-sectional z-scoring. To incorporate both short- and medium-horizon information, the daily score is also blended with its 21-day rolling average, similar to the HSTP model in Section 3.4.2,

$$S_{t,i} = 0.7\, S_{t,i}^{\text{daily}} + 0.3\, \frac{1}{21} \sum_{k=0}^{20} S_{t-k,i}^{\text{daily}}, \tag{50}$$

The blended score is further scaled by inverse volatility and standardized again cross-sectionally to obtain the final momentum signal $z_{t,i}$. The momentum baseline portfolio is constructed through a linear tilt around the Equal-Weight allocation, like the model above $\tilde{w}_{i,t}^{\text{base}} = \frac{1}{N} + \beta\, z_{t,i}$, where β controls the aggressiveness of the tilt. The raw weights are projected onto a long-only, fully invested feasible set by clipping negative values and renormalizing them, followed by per-asset floor and cap constraints to limit extreme allocations. In the reported configuration, we set $\beta = 0.15$, impose a minimum weight of 1% and a maximum weight of 1/3.

The PPO–MOM agent learns deviations around this momentum-only baseline, similar to HSTP, and generates action weights via $w_t = \Pi_{\mathcal{W}}\big(w_t^{\text{base}} \odot (1 + a_t)\big)$ using the same PPO architecture, mean–CVaR reward, turnover and concentration penalties, and training procedure. Hyperparameters are selected via short PPO training runs on the training window and evaluated on the validation window. The selected policy is then trained on the training window with early stopping guided by validation performance and frozen for out-of-sample evaluation. This benchmark represents a learning policy generated from the technical indicator-based observation space and serves as a clean ablation for assessing the marginal contribution of Stage-1 predictive modeling.

A passive market index benchmark is also included to represent the opportunity cost of active allocation, and we selected two exchange-traded funds to represent the benchmarks. For the sector ETF universe, we use SPDR S&P 500 ETF, while for the Dow Jones stock universe, we use SPDR Dow Jones Industrial Average ETF—both index ETFs are the best proxy as passively invested based on securities in both universes. The benchmarks are using buy-and-hold strategies so that the wealth is accumulated from their returns,

$$W_{t+1}^{\text{IDX}} = W_t^{\text{IDX}}(1 + r_t^{\text{IDX}}), \tag{51}$$

4.4. Out-of-Sample Performance and Model Robustness Analysis

This section reports the out-of-sample performance of the proposed hierarchical signal-to-policy (HSTP) framework. Performance is evaluated using daily portfolio returns over the out-of-sample test periods, with metrics annualized where appropriate. To account for the inherent stochasticity of policy gradient training, both HSTP and the momentum-based PPO benchmark (PPO–MOM) are trained and evaluated over 30 independent runs, corresponding to different random seeds but identical data splits and identical Stage-1 inputs. The mean–CVaR optimization benchmark, which solves a static mean–CVaR portfolio optimization problem using rolling historical returns and fixed risk-penalty parameters, is also evaluated over the same 30 rolling iterations to ensure a fair run-level comparison. In contrast, the Equal Weight (EW) strategy and the market index benchmarks, namely, SPY

ETF for the Sector ETF universe and DIA ETF for the DJIA stock universe, are deterministic and, therefore, produce a single realized performance path.

HSTP is evaluated against three benchmarks—(i) PPO–MOM, which removes the hierarchical Stage-1 signal layer; (ii) the mean–CVaR optimization benchmark, which incorporates explicit tail risk, turnover, and concentration penalties calibrated from the PPO agent but excludes reinforcement learning; (iii) the Equal Weight portfolio; and (iv) the relevant passive index ETFs—and all strategies are assessed under the same rolling out-of-sample protocol.

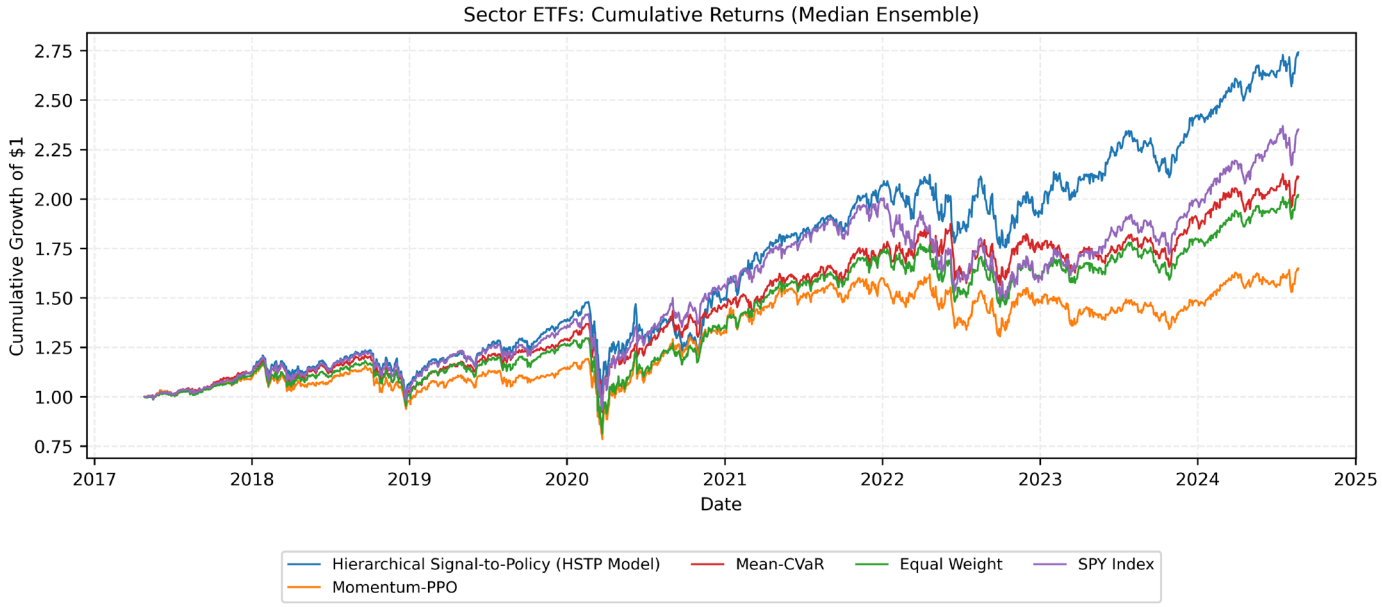
4.4.1. Aggregate Return and Risk Metrics

Figure 3 illustrates the out-of-sample cumulative wealth trajectories. For learning-based strategies (HSTP, PPO–MOM, and mean–CVaR), we report median ensemble paths across the 30 iteration runs, while deterministic benchmarks of index ETFs and Equal Weight are shown as single paths. Across both investment universes, HSTP exhibits the strongest cumulative growth profile, indicating that the hierarchical signal layer improves both the quality and consistency of the learned allocation policy in out-of-sample settings, delivering gains beyond those achievable through either end-to-end reinforcement learning or static risk-aware optimization alone.

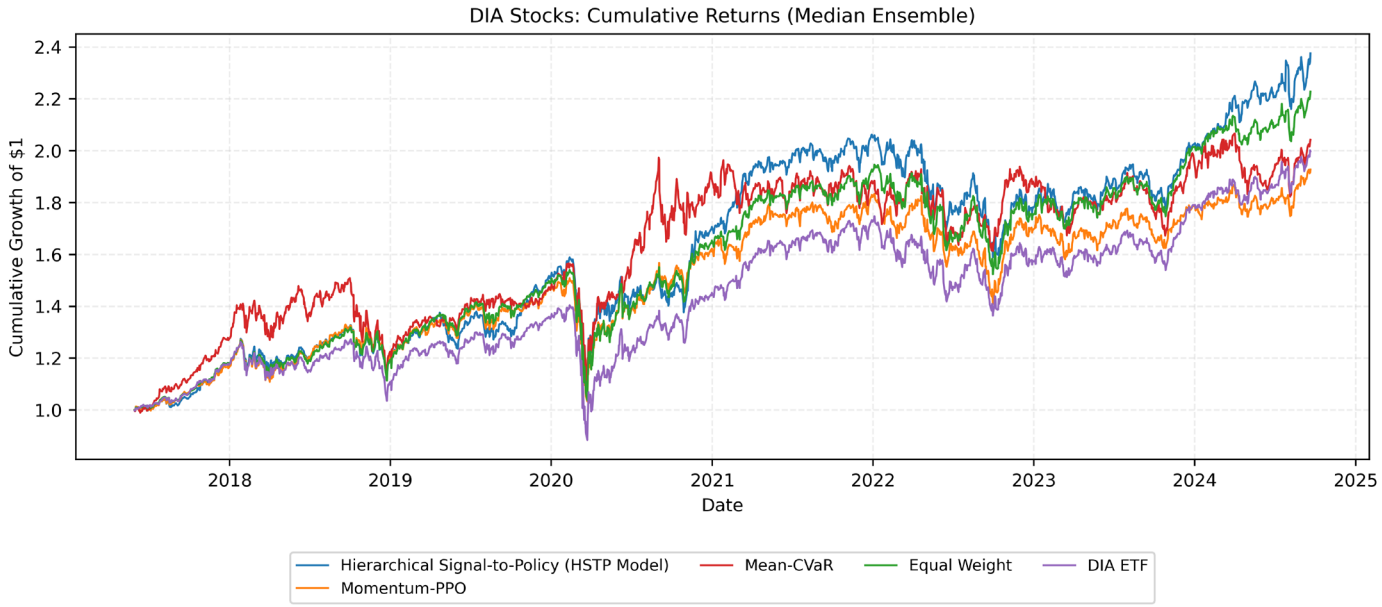
For the Sector ETF universe, the median terminal wealth (growth of \$1) is approximately 2.72× for HSTP, compared with 1.66× for PPO–MOM, 2.02× for EW, 2.1× for mean–CVaR, and 2.35× for SPY. For the DJIA stock universe, HSTP achieves a median terminal wealth of approximately 2.42×, exceeding PPO–MOM (1.93×), EW (2.23×), mean–CVaR (2.04×) and the DIA ETF benchmark (2.00×).

Tables 1 and 2 report mean out-of-sample performance metrics computed from daily returns. Learning-based strategies are averaged across the 30 runs, while Equal Weight and index ETFs are reported as single-path outcomes. Reported performance metrics are computed from gross portfolio returns; the implied transaction-cost drag based on realized turnover is reported separately in Section 4.6 and Appendix D.

In the Sector ETF universe, HSTP achieves the highest annual return (14.60%), outperforming Momentum–PPO (7.11%), Equal Weight (10.11%), mean–CVaR (10.75%), and SPY (12.41%). Although HSTP exhibits slightly higher volatility, it delivers the strongest risk-adjusted performance, with the highest return-to-volatility ratio (0.75) and Sortino ratio (0.98). A similar pattern is observed in the DJIA stock universe. HSTP again delivers the highest annual return (12.58%), exceeding Momentum–PPO (9.26%), Equal Weight (11.59%), mean–CVaR (10.09%), and the DIA ETF (9.93%). While volatility levels are broadly comparable across strategies, HSTP attains the strongest overall risk-adjusted efficiency, with a return-to-volatility ratio of 0.68 and a Sortino ratio of 0.90. Relative to the Equal Weight portfolio, which performs competitively in this diversified large-cap universe, HSTP shows incremental gains in both absolute return and downside-adjusted performance. Compared with Momentum–PPO and the passive ETF, the hierarchical framework exhibits a clear advantage, highlighting the value of combining predictive modeling and risk-aware reinforcement learning for both universes.



Panel (**a**): Sector ETF universe



Panel (**b**): DJIA stock universe

Figure 3. Out-of-sample median cumulative growth among strategies (test window: 2017–2024; 21-trading-day rebalancing).

Table 1. Sector ETF universe out-of-sample test results.

Strategy	Annual Return	Annual Volatility	Return/Volatility	Sortino
HSTP (Proposed)	14.60%	19.48%	0.75	0.98
Momentum–PPO	7.11%	18.80%	0.38	0.56
Mean–CVaR	10.75%	18.07%	0.59	0.79
Equal Weight	10.11%	18.38%	0.55	0.72
SPY ETF	12.41%	19.00%	0.65	0.85

Notes: Metrics are computed from daily portfolio returns and annualized where applicable. HSTP, mean–CVaR and Momentum–PPO results are averaged across 30 runs; EW and SPY are single-path benchmarks.

Table 2. DJIA stock universe out-of-sample test results.

Strategy	Annual Return	Annual Volatility	Return/Volatility	Sortino
HSTP (Proposed)	12.58%	18.62%	0.68	0.9
Momentum–PPO	9.26%	17.97%	0.52	0.7
Mean–CVaR	10.09%	21.39%	0.47	0.71
Equal Weight	11.59%	17.92%	0.65	0.83
DIA ETF	9.93%	18.85%	0.53	0.7

Notes: Metrics are computed from daily portfolio returns and annualized. HSTP, mean–CVaR and Momentum–PPO are averaged across 30 runs; Equal Weight and DIA are single-path benchmarks.

4.4.2. Downside Risk and Tail Behavior

Table 3 reports downside risk characteristics using maximum drawdown and monthly CVaR at the 5% level (lower CVaR values indicate better downside protection). In the Sector ETF universe, HSTP achieves both the smallest drawdown and the least severe tail losses. In the DJIA stock universe, HSTP again delivers the lowest maximum drawdown and remains highly competitive on tail risk. Although EW exhibits a marginally better CVaR than HSTP, the difference is economically small relative to HSTP's superior return and efficiency.

Table 3. Downside-risk measures across asset universes.

Universe	Strategy	Maximum Drawdown (%)	CVaR (5%, 21-Trading-Day Horizon)
Sector ETFs	HSTP (Proposed)	−31.84%	10.10%
	Momentum–PPO	−34.00%	10.15%
	Mean–CVaR	−32.67%	9.70%
	Equal Weight	−37.18%	10.52%
	SPY ETF	−34.10%	10.42%
DJIA Stocks	HSTP (Proposed)	−31.43%	7.72%
	Momentum–PPO	−31.60%	7.80%
	Mean–CVaR	−29.40%	9.17%
	Equal Weight	−32.66%	7.71%
	DIA ETF	−37.06%	8.83%

Notes: HSTP, mean–CVaR and Momentum–PPO strategies are averaged across 30 runs; EW and index ETFs are single-path benchmarks. CVaR statistics are reported as a positive loss magnitude at the 5% level over a 21-trading-day horizon, matching the reward function horizon; lower values indicate better downside protection.

4.5. Statistical Robustness and Run-Level Stability

4.5.1. Run-Level Win Rates

Because PPO training is stochastic, we further examine robustness using run-level descriptive statistics. Table 4 reports win rates, defined as the percentage of runs in which HSTP strictly outperforms a given benchmark on a specific metric.

In the Sector ETF universe, HSTP shows 100% dominance on return-based metrics, outperforming Momentum–PPO, Equal Weight, and the index benchmark among all 30 runs for both annual return and return-to-volatility. Notably, HSTP also achieves superior drawdown control, with win rates of 100% across all benchmarks for maximum drawdown. While differences in monthly CVaR outcomes are more mixed, reflecting the inherently conservative nature of tail risk metrics, HSTP continues to outperform the Equal Weight and index benchmarks in the majority of runs, indicating consistent downside risk performance.

In the DJIA stock universe, where diversification effects naturally reduce performance dispersion, HSTP maintains strong run-level robustness. HSTP outperforms Momentum–PPO and the index benchmark in all runs for both annual return and return-to-volatility ratio and exceeds Equal Weight in a large majority of cases. For downside-risk measures, HSTP demonstrates particularly strong performance relative to Equal Weight and the index benchmark, achieving higher win rates for maximum drawdown and monthly CVaR. Although Momentum–PPO has slightly lower drawdowns and CVaR in individual runs, these outcomes are not

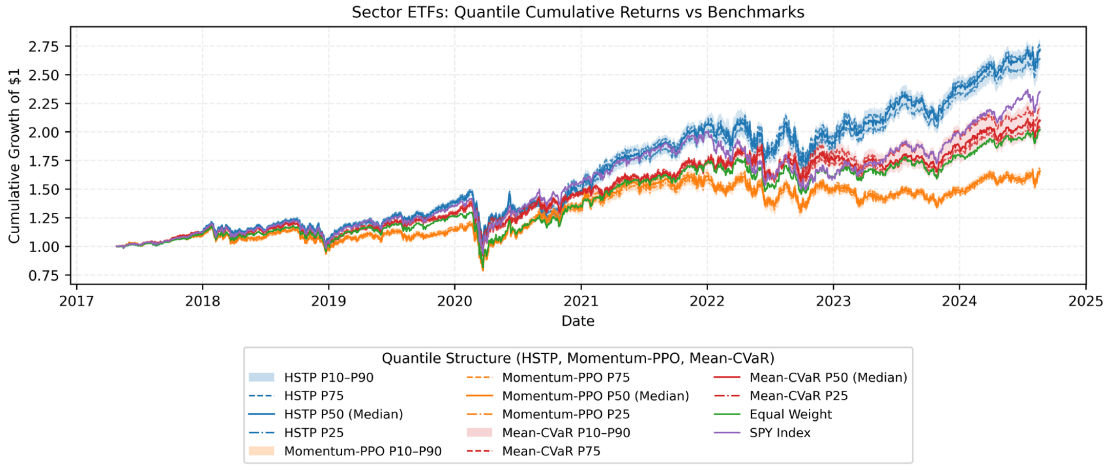
systematic and occur alongside weaker performance on the return-based metric. Appendix C provides additional window-level risk–return diagnostics across rolling out-of-sample periods.

Table 4. Run-level win rates of HSTP (%).

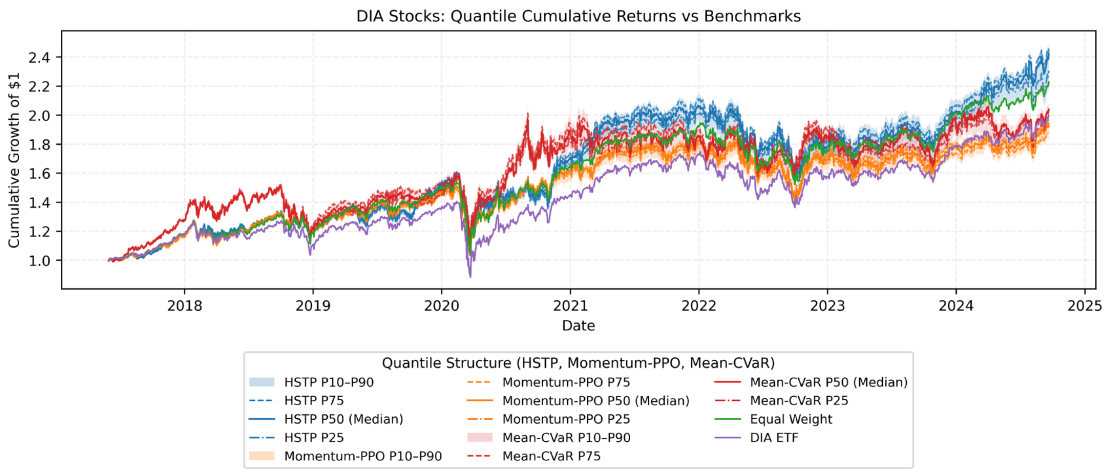
Universe	Metric	Vs. Momentum–PPO	Vs. Equal Weight	Vs. Index ETF
Sector ETFs	Annual Return	100	100	100
	Return/Volatility	100	100	100
	Maximum Drawdown	100	100	100
	CVaR (5%, 21-trading-day horizon)	53.33	90	83.33
DJIA Stocks	Annual Return	100	86.67	100
	Return/Volatility	100	80	100
	Maximum Drawdown	46.67	93.33	100
	CVaR (5%, 21-trading-day horizon)	66.67	53.33	100

Notes: Win rates compare each HSTP run against the benchmark value. These results are descriptive and not formal hypothesis tests.

Finally, Figure 4 visualizes the distribution of performance outcomes across the 30 runs. The plots show that HSTP improves not only average outcomes but also reliability, with fewer runs exhibiting weak growth or prolonged stagnation relative to Momentum–PPO.



Panel (**a**): Sector ETF universe



Panel (**b**): DJIA stock universe

Figure 4. Distributional robustness of out-of-sample performance (test window: 2017–2024; 21-trading-day rebalancing). The figure displays quantile cumulative wealth paths across rolling windows, highlighting the dispersion and lower-tail behavior of learning-based strategies relative to benchmarks.

4.5.2. Run-Level Paired Statistical Inference

For each independent training run $i = 1, \ldots, 30$, we compute paired performance differences $\Delta_i = M(S_i) - M(B_i)$, where S_i denotes the HSTP strategy, B_i denotes a benchmark evaluated on the same out-of-sample period, and $M(\cdot)$ denotes either compound annual growth rate (CAGR) or the annualized Sharpe ratio. Statistical inference is conducted using two complementary paired tests.

- Paired *t*-test (mean-based inference)

The paired *t*-test evaluates whether the mean of the paired differences is positive. Formally, the null hypothesis is $H_0 : \mathbb{E}(\Delta) \leq 0$, against the one-sided alternative $H_1 : \mathbb{E}(\Delta) > 0$. In Table 5, mean ΔCAGR and mean ΔSharpe report the sample mean of Δ_i across the 30 runs. The Sharpe ratio is defined as the annual return divided by the annual volatility, assuming a risk-free rate of zero. t(CAGR) and t(Sharpe) denote the corresponding paired t-statistics, while p_t(CAGR) and p_t(Sharpe) report the one-sided *p*-values of the paired *t*-test.

Table 5. Run-level paired statistical inference across 30 training runs.

Universe	Benchmark	Mean ΔCAGR	t (CAGR)	p_t (CAGR)	Median ΔCAGR	p_w (CAGR)	Mean ΔSharpe	t (Sharpe)	p_t (Sharpe)	Median ΔSharpe	p_w (Sharpe)
Sector ETFs	Momentum–PPO	0.0749	44.85	<0.001	0.0748	<0.001	0.3381	38.11	<0.001	0.3419	<0.001
Sector ETFs	Mean–CVaR	0.0385	18.83	<0.001	0.0374	<0.001	0.1428	16.63	<0.001	0.1366	<0.001
Sector ETFs	Equal Weight	0.0449	37.03	<0.001	0.0456	<0.001	0.1816	26.48	<0.001	0.1799	<0.001
Sector ETFs	Index ETF (SPY)	0.0219	18.06	<0.001	0.0226	<0.001	0.0868	12.65	<0.001	0.0851	<0.001
DJIA Stocks	Momentum–PPO	0.0332	18.03	<0.001	0.0342	<0.001	0.1466	16.51	<0.001	0.1505	<0.001
DJIA Stocks	Mean–CVaR	0.0249	14.57	<0.001	0.0249	<0.001	0.1728	22.48	<0.001	0.1861	<0.001
DJIA Stocks	Equal Weight	0.0099	6.59	<0.001	0.0129	<0.001	0.0278	3.81	<0.001	0.0434	<0.001
DJIA Stocks	Index ETF (DIA)	0.0265	17.59	<0.001	0.0294	<0.001	0.1324	18.12	<0.001	0.148	<0.001

Notes: Results are based on 30 independent training runs. All *p*-values are one-sided. Win rates indicate the fraction of runs in which HSTP outperforms the benchmark and are included for descriptive reference.

- Wilcoxon signed-rank test (median-based robustness)

To mitigate sensitivity to non-normality or outlier runs, we additionally apply the Wilcoxon signed-rank test, a nonparametric paired test that evaluates whether the median of the paired differences is positive. The null hypothesis is $H_0 : \text{median}(\Delta) \leq 0$, against $H_1 : \text{median}(\Delta) > 0$. In Table 5, median ΔCAGR and median ΔSharpe report the sample median of the paired differences, while p_w(CAGR) and p_w(Sharpe) denote the corresponding one-sided Wilcoxon *p*-values.

In the Sector ETF universe, HSTP achieves uniformly positive mean and median improvements relative to all benchmarks, with both the paired *t*-test and Wilcoxon signed-rank test strongly rejecting the null hypothesis of no improvement. These results indicate that performance gains are systematic across training realizations rather than driven by isolated runs. In the DJIA stock universe, performance improvements are smaller but remain statistically significant across all benchmarks. Gains are strongest relative to Momentum–PPO, mean–CVaR, and the index ETF, while improvements over the Equal Weight portfolio are more modest, reflecting the more constrained opportunity set for those large-cap stocks.

4.5.3. Sensitivity Analysis of the Reward Function Parameters

To assess whether the empirical performance of the proposed hierarchical signal-to-policy (HSTP) framework is driven by tuning of optimization of penalty parameters, we conduct a sensitivity analysis that varies the key components of the reward function governing downside risk and trading frictions. The analysis is implemented as a grid-based stress test over the Conditional Value-at-Risk (CVaR) penalty coefficient and the turnover penalty, while holding all other aspects of the modeling framework fixed. In particular, the hierarchical architecture, state representation, action space, PPO hyperparameters, rolling training scheme, rebalancing frequency, portfolio constraints, and out-of-sample evaluation protocol are kept identical to the baseline specification. This design isolates the sensitivity of the learned allocation policy to economically meaningful preference parameters rather than to modeling or data choices.

The stress-test grid spans three levels of CVaR risk aversion, $\lambda_{\text{CVaR}} \in \{0.10,\ 0.25,\ 0.50\}$, and three levels of turnover penalization $\lambda_{\text{TO}} \in \{0.001,\ 0.003,\ 0.005\}$, combined with three independent random seeds, yielding 27 distinct parameter configurations for each investment universe.

For comparison, the HSTP model uses the proposed configuration described above based on 30 independent PPO runs. Both the HSTP model and stress-test runs are evaluated over identical out-of-sample periods with the same portfolio constraints and rebalancing frequency, ensuring a fair and controlled comparison.

The resulting cumulative return distribution over time is illustrated in Figure 5. It reports cumulative wealth trajectories for both the sector ETF universe and the DJIA stock universe, where the baseline HSTP ensemble is shown alongside the distribution of stress-test permutations summarized by their median trajectory and 5–95% envelope. Rather than highlighting individual runs, this presentation emphasizes distributional robustness. The baseline HSTP trajectory closely tracks the center of the stress-test distribution throughout the sample and remains well within the envelope during major market episodes, including the sharp drawdown and recovery surrounding the 2020 market stress. The absence of systematic divergence between the baseline and the stress-test median indicates that long-run performance is not sensitive to the chosen risk or turnover penalties.

Robustness in terms of risk-adjusted performance is further examined in Figure 6, which reports the distribution of annualized Sharpe ratios (with the risk-free rate as zero) across all 27 stress-test configurations for the DJIA stock and sector ETF universes. In both cases, the Sharpe ratio of the baseline HSTP ensemble lies close to the center of the permutation distribution.

A quantitative summary of these results is provided in Table 6, which compares the baseline HSTP ensemble with the stress-test permutations across different performance metrics, i.e., annualized return, volatility, Sharpe ratio, maximum drawdown, CVaR at the 5% level, and terminal wealth. Differences between the baseline and the permutation mean are economically small relative to cross-configuration dispersion, and no single parameter setting dominates performance across metrics.

Taken together, these findings indicate that the empirical gains of the proposed HSTP framework are not sensitive to the risk or transaction-cost parameter choices. Instead, performance remains stable across a broad and economically plausible range of CVaR and turnover penalty values. This robustness shows that the observed improvements are mainly from the hierarchical signal-to-policy architecture and the learning dynamics of the PPO agent.

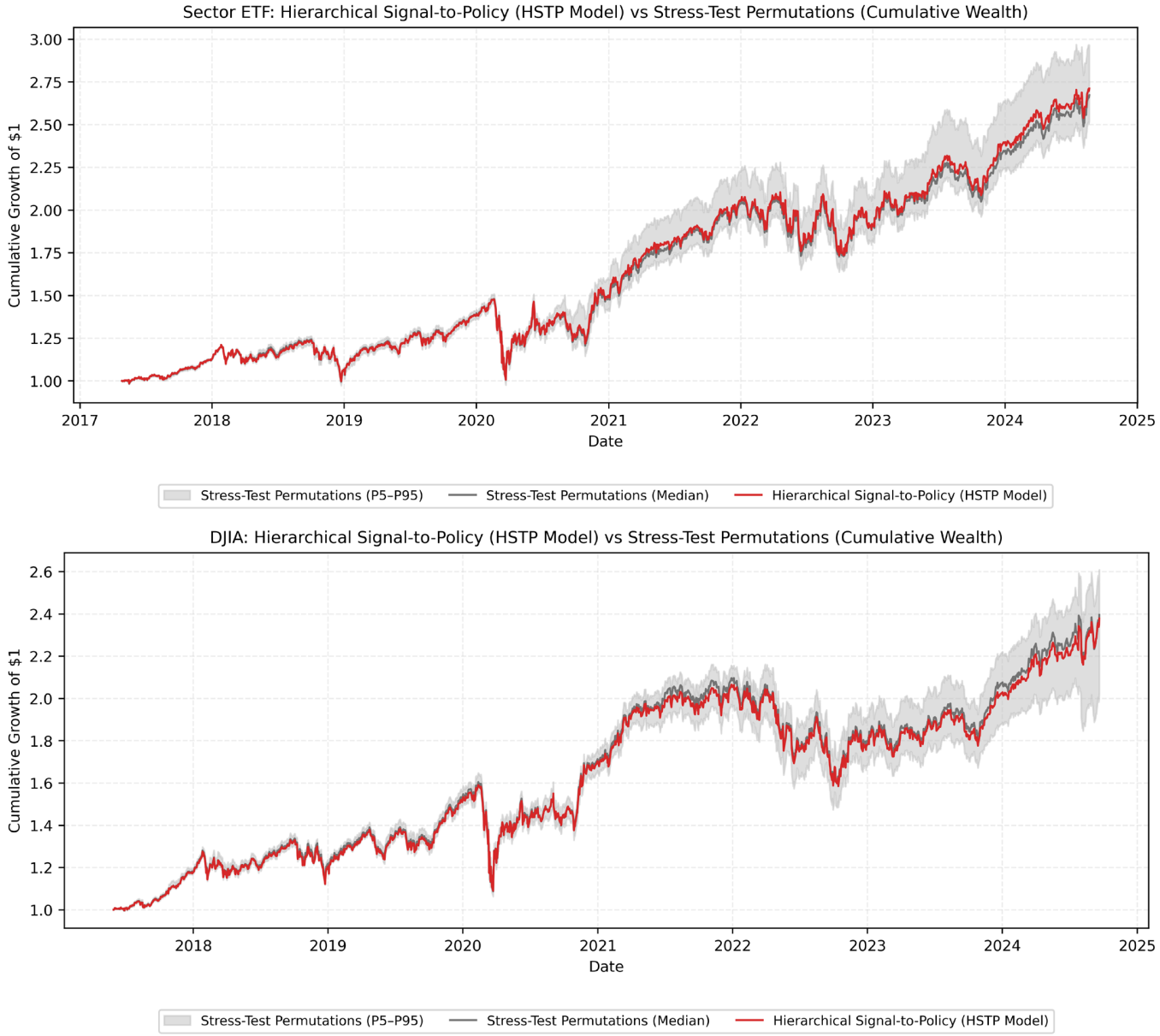


Figure 5. Cumulative wealth under stress-test permutations (test window: 2017–2024; 21-trading-day rebalancing). Out-of-sample cumulative wealth for the Sector ETF (**top**) and DJIA stock (**bottom**) universes. The baseline HSTP ensemble is shown together with the median and 5–95% envelope across stress-test parameter permutations, illustrating distributional robustness of long-run performance.

Table 6. Robustness to CVaR and turnover penalty.

Metric	HSTP (DJIA Stocks)	Permutation Mean (DJIA Stocks)	Perm. Std (DJIA Stocks)	HSTP (Sector ETFs)	Perm. Mean (Sector ETFs)	Perm. Std (Sector ETFs)
Annualized Return	0.1359	0.1338	0.0125	0.1553	0.1543	0.0067
Annualized Volatility	0.1854	0.1881	0.002	0.1936	0.1938	0.0042
Sharpe (rf = 0)	0.733	0.7113	0.0685	0.8022	0.797	0.0415
Max Drawdown	−0.3144	−0.3209	0.0103	−0.3184	−0.3193	0.0077
CVaR (5%, 21-trading-day horizon)	0.0765	0.0784	0.0021	0.1006	0.1018	0.0032
Terminal Wealth of $1 Investment	2.3789	2.343	0.2115	2.7123	2.6949	0.1384

Notes: Comparison of baseline HSTP ensemble performance with the distribution of stress-test permutations across CVaR and turnover penalty configurations. Stress-test statistics are computed across 27 parameter combinations.

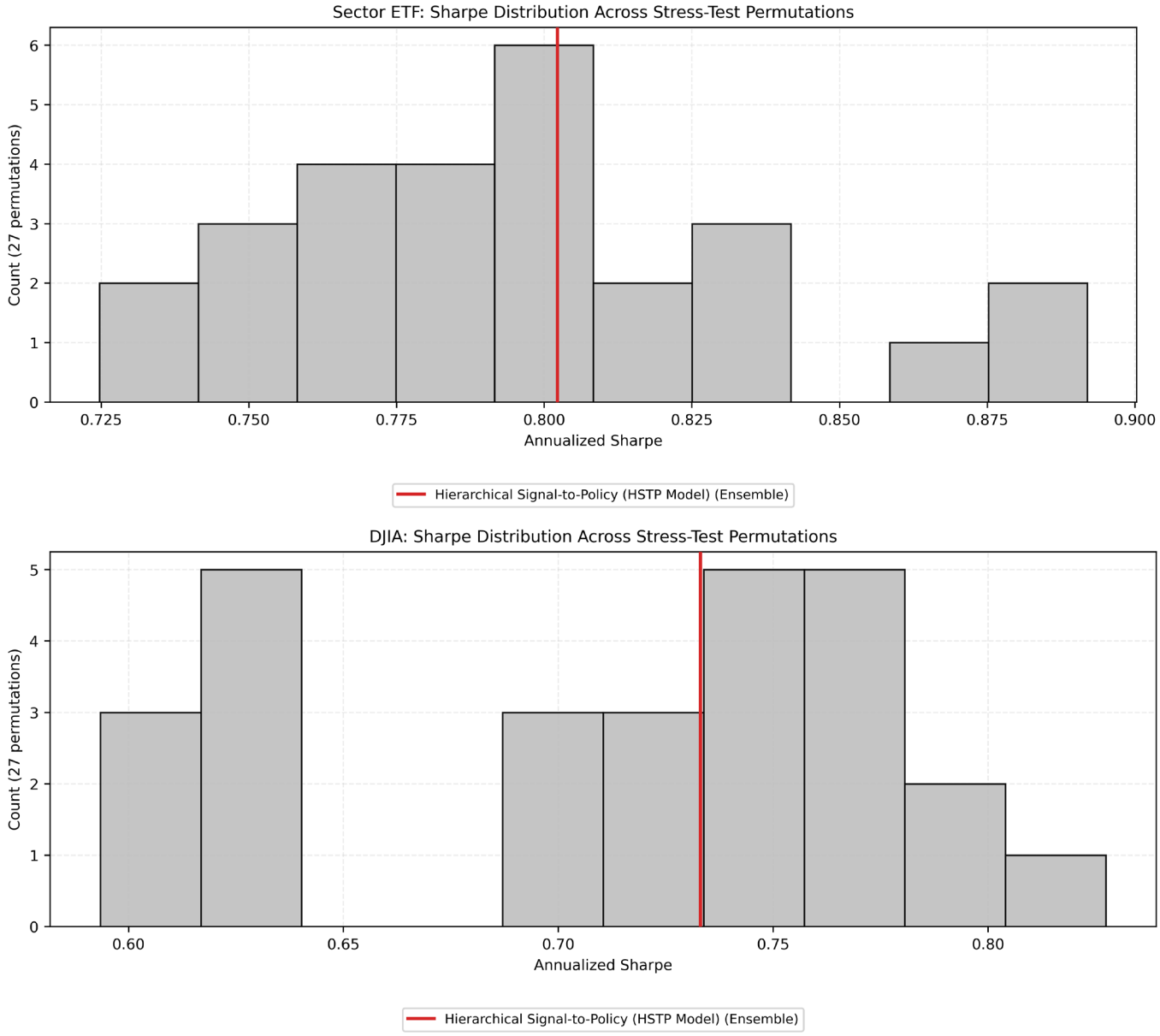


Figure 6. Sharpe ratio distribution under stress-test permutations (test window: 2017–2024; 21-trading-day rebalancing). Distribution of annualized Sharpe ratios (risk-free rate set to zero) across stress-test permutations for the Sector ETF (**top**) and DJIA stock (**bottom**) universes. The baseline HSTP ensemble (red line) lies near the center of the distribution in both cases, indicating stable risk-adjusted performance.

4.6. Turnover Dynamics and Transaction Cost Impact

To evaluate the practical feasibility of the proposed hierarchical signal-to-policy framework, we analyze realized portfolio turnover and the implied transaction cost impact using out-of-sample portfolio weights. The objective of this analysis is to assess whether the learned rebalancing behavior remains economically reasonable under realistic implementation assumptions, rather than to mechanically minimize trading activity.

Across the Sector ETF universe, the strategy follows a stable monthly rebalancing pattern, resulting in an average one-way turnover of approximately 8–9% per rebalance and an annualized one-way turnover of roughly 100–110%. In the DJIA component universe, turnover is higher—averaging 30–35% per rebalance and approximately 400% annually—reflecting the larger number of individual securities and the finer allocation adjustments required at the stock level.

To quantify the economic impact of trading activity, we adopt a linear transaction cost model:

$$\text{Cost}_y = \tau_y \times c_{\text{side}}, \tag{52}$$

where τ_y denotes one-way annual turnover and c_{side} represents a per-side transaction cost in basis points. Under a base scenario of 5 bps per side for ETFs and 10 bps per side for equities, the implied annual transaction cost drag is modest. For the Sector ETF portfolios, the estimated cost impact averages approximately 0.05–0.10% per year, while for the DJIA portfolios, the corresponding drag is approximately 0.4–0.5% per year. These indicate that trading frictions contribute only a small reduction to net performance.

Additional detailed analysis related to portfolio transaction cost as well as concentration and position sizing are provided in Appendix D.

5. Conclusions and Future Work

This study proposes a hierarchical signal-to-policy (HSTP) learning framework for dynamic portfolio optimization that explicitly separates return signal generation from risk-aware portfolio decision-making. By combining an interpretable machine learning prediction layer with a Proximal Policy Optimization (PPO) agent trained under a mean–CVaR objective, the framework integrates predictive modeling, reinforcement learning, and classical portfolio theory within a transparent and modular architecture.

Empirical results across two distinct asset universes—U.S. sector ETFs and equities from the Dow Jones Industrial Average—provide strong and consistent evidence in support of the proposed approach. Relative to three widely used benchmarks, namely an end-to-end Momentum–PPO agent, an Equal-Weight strategy, a mean–CVaR strategy, and passive market-tracking index ETFs, the HSTP framework delivers economically meaningful improvements in out-of-sample performance. These gains are reflected not only in higher annualized returns but, more importantly, in superior risk-adjusted efficiency, as measured by return-to-volatility and the Sortino ratio. At the same time, the framework maintains competitive or improved downside protection, achieving smaller or comparable maximum drawdowns and more favorable tail risk behavior, as captured by monthly CVaR.

Robustness and stability checks show that these gains are applicable across different reinforcement learning training settings. Across 30 independent PPO runs with distinct random initializations, the HSTP strategy exhibits consistently higher run-level win rates on return and efficiency metrics than competing benchmarks, along with tighter performance dispersion and more stable cumulative wealth trajectories. Rather than forcing a single black-box agent to simultaneously infer market structure and execute portfolio actions, the hierarchical architecture allows each component to focus on a well-defined role: Stage 1 produces economically interpretable return signals, while Stage 2 learns how to translate those signals into portfolio decisions under explicit risk preferences and portfolio constraints. This separation highlights a key practical advantage of hierarchical design. By isolating signal extraction within a dedicated predictive layer and constraining policy learning through a risk-aware objective, the allocation policy becomes less sensitive to noise, initialization, and training uncertainty than fully end-to-end reinforcement learning approaches.

Several directions for future research naturally follow. First, while this study focuses on equities and sector ETFs, the framework readily extends to multi-asset portfolios incorporating fixed income, commodities, or alternative assets, where regime shifts and tail risks may be even more pronounced. Second, future work may explore adaptive risk preferences, allowing the agent to dynamically adjust its risk modeling beyond a fixed CVaR specification in response to evolving market conditions. Finally, enhancing the signal layer with additional macroeconomic information or market-level sentiment indicators

could provide a useful direction for improving predictive power while preserving the modular structure of the hierarchical framework.

Overall, this study demonstrates that hierarchical signal-to-policy learning provides a principled and effective approach to risk-aware portfolio optimization. By integrating predictive modeling, explainable AI, and reinforcement learning within a coherent framework, the proposed method advances both the methodological foundations and the practical applicability of data-driven portfolio management, bridging the gap between modern AI techniques and the demands of transparent, robust financial decision-making.

Author Contributions: Conceptualization, J.Y. and K.-C.C.; methodology, J.Y. and K.-C.C.; validation, J.Y. and K.-C.C.; formal analysis, J.Y. and K.-C.C.; data curation, J.Y.; writing—original draft preparation, J.Y.; writing—review and editing, K.-C.C.; visualization, J.Y.; supervision, K.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are derived from publicly available financial market sources. Redistribution of raw price and factor data is subject to vendor licensing restrictions.

Acknowledgments: During the preparation of this manuscript, the author(s) used ChatGPT 5.2 to assist with wording and language refinement. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Stage-1 Feature Definition

This table summarizes the full set of features used in the Stage-1 return prediction models. Features are organized into security-specific technical indicators and global Fama–French factor families. All numerical subscripts explicitly denote trading days. Technical indicators are computed separately for each asset, while global factors are shared across assets and included with multiple daily lags to capture delayed market effects.

Table A1. Securities-specific technical indicators.

Feature Name	Category	Definition (in Trading Days)	Economic Interpretation
SMA_5	Trend	Simple moving average over the past 5 trading days	Short-term price trend
EMA_12	Trend	Exponential moving average with a 12-day decay window	Momentum with recency weighting
RSI_7	Momentum	Relative Strength Index computed over 7 trading days	Overbought/oversold signal
MACD	Momentum	Difference between fast and slow EMAs (standard 12–26-day construction)	Trend acceleration and reversal
ATR	Volatility	Average True Range computed using daily high-low-close data (standard 14-day window unless stated otherwise)	Intraday and short-term risk
Vol_5	Volatility	Realized return volatility over the past 5 trading days	Short-horizon uncertainty
Mom_3	Momentum	Cumulative return over the past 3 trading days	Very short-term momentum
LagRet_1	Autoregressive	Return lagged by 1 trading day	Return persistence/reversal
LagRet_2	Autoregressive	Return lagged by 2 trading days	Short-term dynamics
LagRet_3	Autoregressive	Return lagged by 3 trading days	Short-term regime continuation

Notes: All indicators are computed for each asset using only information available up to time t, ensuring no look-ahead bias. The feature set jointly captures trend, momentum, volatility, and short-horizon autoregressive effects, which are well documented to exhibit predictive power at daily to weekly horizons.

Table A2. Global Fama–French factor families.

Factor Family	Included Variables	Definition (Daily Lags)	Financial Interpretation
MKT	Mkt-RF, Mkt-RF_lag_1, _lag_2, _lag_3	Market excess return contemporaneous and lagged by 1–3 trading days	Aggregate market risk premium
SMB	SMB, SMB_lag_1, _lag_2, _lag_3	Size factor with 1–3-day lags	Small-minus-big capitalization effect
HML	HML, HML_lag_1, _lag_2, _lag_3	Value factor with 1–3-day lags	Value vs. growth premium
RMW	RMW, RMW_lag_1, _lag_2, _lag_3	Profitability factor with 1–3-day lags	Operating profitability premium
CMA	CMA, CMA_lag_1, _lag_2, _lag_3	Investment factor with 1–3-day lags	Conservative vs. aggressive investment

Notes: All global factors are measured at daily frequency and multiple daily lags are included to allow the model to capture delayed transmission of macroeconomic information into asset prices.

Appendix B

Technical Indicator Feature Definitions

Table A3 reports the complete set of features included in the Stage-2 observation space, with universe-specific applicability indicated explicitly. All quantities are computed using information available up to time t and stacked over a 21-day lookback window prior to each rebalance date. Importantly, all features used in Stage 2 are either outputs of Stage 1 or inputs originally used to train the Stage-1 predictive models; no new feature families are introduced at this stage.

Table A3. Stage-2 Observation Features by Investment Universe.

Feature Block	Variables	Sector ETFs	DJIA Stocks	Description
Predicted returns	Predicted_Return_{i}	✓	✓	One-day-ahead return forecasts produced by Stage-1 XGBoost models; asset-level alpha signals.
Volatility	Volatility_{i}	✓	✓	Short-horizon realized or estimated volatility computed in Stage 1; used as asset-level risk context.
Cross-sectional rank	Rank_PredRet_{i}	✓	✓	Rank of predicted returns across assets on date (t), normalized by universe size; provides scale-free relative performance signals.
SHAP mean summaries	Avg_SHAP_* (gated groups)	✓	✓	Cross-sectional mean of group-level SHAP attributions computed in Stage 1; summarize dominant explanatory drivers of forecasts at the universe level.
SHAP dispersion summaries	Std_SHAP_* (gated groups)	✓	✓	Cross-sectional standard deviation of group-level SHAP attributions; capture heterogeneity in factor influence across assets.
Relative SHAP metrics	*_REL	✓	✓	Cross-sectionally demeaned SHAP summaries from Stage 1; isolate asset-specific explanatory deviations from common effects.
Aggregate SHAP totals	Avg_SHAP_Total, Std_SHAP_Total	✓	✓	Aggregate magnitude and dispersion of SHAP contributions across all gated feature groups.
Cross-sectional signal context	CrossSec_Mean_PredRet, CrossSec_Std_PredRet	✓	✓	Cross-sectional mean and dispersion of predicted returns computed in Stage 1; describe overall signal strength and opportunity spread.
Cross-sectional risk context	CrossSec_Mean_Volatility	✓	✓	Cross-sectional mean of asset-level volatility; summarizes prevailing risk conditions across the universe.

Table A3. *Cont.*

Feature Block	Variables	Sector ETFs	DJIA Stocks	Description
Macro/factor variables	Mkt-RF, SMB_lag_2, CMA, RMW_lag_3	✓	–	Daily or lagged factor realizations used as inputs in Stage 1 and reused in Stage 2 as common conditioning variables (sector ETF universe).
Macro/factor variables	Mkt-RF, SMB_lag_2, HML_lag_2, CMA_lag_2	–	✓	Daily or lagged factor realizations used as inputs in Stage 1 and reused in Stage 2 (DJIA stock universe).
Asset-level technical indicators	EMA_12, SMA_5, RSI_7, LagRet_1, Mom_3, Vol_5	✓	–	Asset-specific technical indicators originally used as Stage-1 inputs; reused in Stage 2 to provide short-term trend and volatility context for ETFs.
Asset-level technical indicators	EMA_12, MACD, ATR, LagRet_1, LagRet_2, Mom_3	–	✓	Asset-specific technical indicators used as Stage-1 inputs; reused in Stage 2 to capture firm-level trend, momentum, and volatility heterogeneity.
Momentum factor block	Vol_20, Mom_5, Mom_10, Mom_20	✓	✓	Multi-horizon volatility and momentum measures used as Stage-1 inputs and reused in Stage 2 to encode trend persistence and regime shifts.

Note: ✓ indicates the feature group is included in the Stage-2 observation space for corresponding universe; – indicates it is not used; * denotes multiple SHAP feature groups are generated by the Stage-1 feature gate.

Appendix C

Risk–Return Diagnostics Across Rolling Windows

To complement the median performance statistics and run-level robustness analysis presented in Section 4.4, this appendix provides additional diagnostic visualizations of the risk–return trade-off across rolling out-of-sample evaluations. The figures plot realized compound annual growth rates (CAGR) against annualized volatility for each rolling window, highlighting the dispersion of outcomes and the relative positioning of the proposed HSTP strategy versus benchmark portfolios.

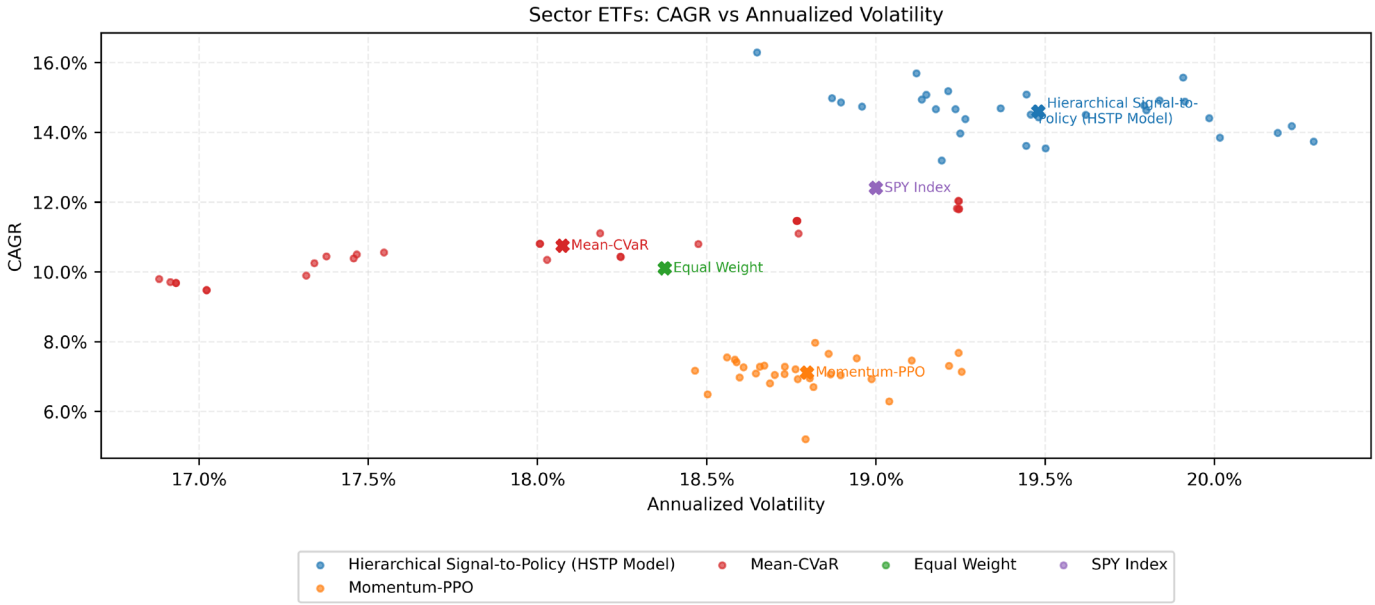


Figure A1. Risk–return scatter plots across iterations (sector ETF universe). The figure plots out-of-sample compound annual growth rate (CAGR) against annualized volatility for each rolling evaluation window in the Sector ETF universe. Each point represents one iteration. Reinforcement learning-based strategies are shown as scatter plots, while the Equal Weight benchmark is shown as a single reference point.

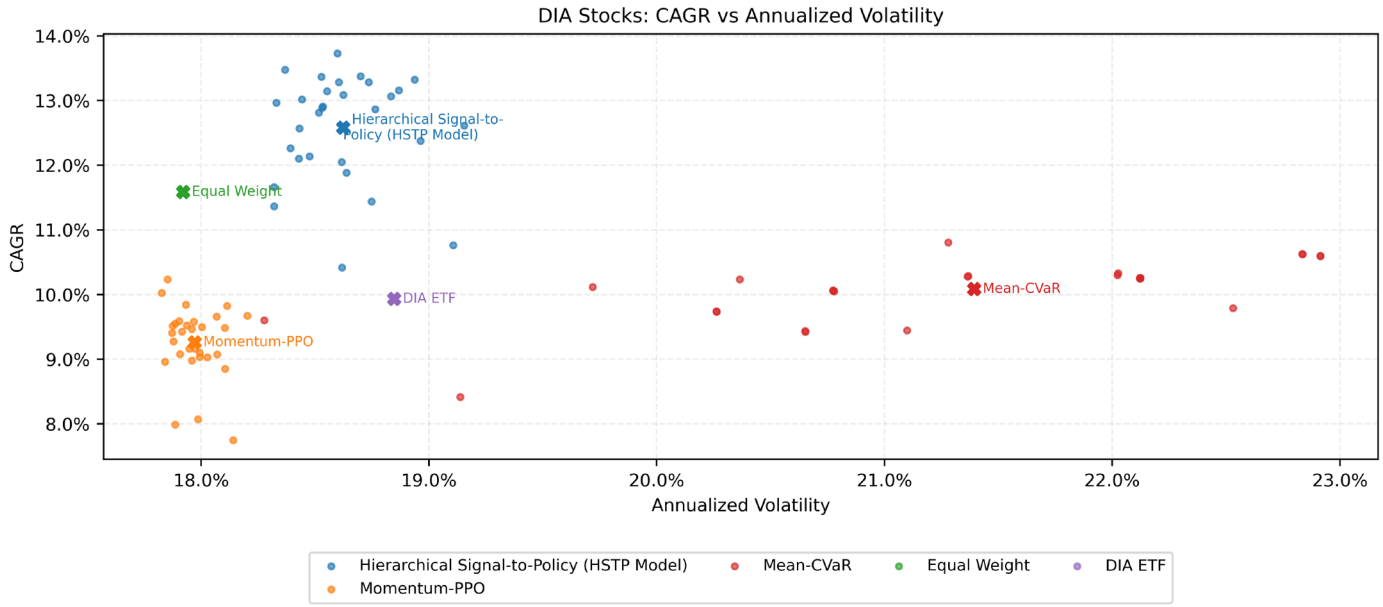


Figure A2. Risk–return scatter plots across iterations (DJIA stock universe).

Appendix D

This appendix provides additional analysis on turnover, transaction cost behavior and portfolio structure to support the implementation feasibility discussion in Section 4.6.

Appendix D.1. Annual Turnover and Transaction Cost Analysis

Figure A3 reports the annual one-way turnover and the corresponding implied transaction cost impact for the Sector ETF and DJIA universes. Turnover is aggregated at the calendar year level to highlight persistent trading patterns rather than short-term fluctuations.

For the Sector ETF universe, annual turnover remains close to a single portfolio rotation per year in most sample years, consistent with a stable monthly rebalancing schedule. The implied transaction cost series mirrors this behavior, remaining low and stable over time. For the DJIA component universe, annual turnover is higher, reflecting the larger cross-section of individual securities and more granular reallocation. Nevertheless, the corresponding transaction cost impact remains moderate and exhibits no signs of excessive variability or structural instability.

Overall, the figure confirms that trading activity is systematic and economically well-behaved, supporting the feasibility of implementing the proposed framework under standard market frictions.



Figure A3. Annual portfolio turnover and implied transaction cost impact. The figure reports one-way annual turnover and the corresponding implied transaction cost drag for the Sector ETF and DJIA universes. Annual turnover is computed as the sum of one-way turnover across rebalancing events within each calendar year. Implied transaction costs are calculated using a linear cost model with 5 bps per side for ETFs and 10 bps per side for equities. Dashed horizontal lines indicate sample averages over near-full calendar years.

Appendix D.2. Portfolio Concentration and Active-Weight Diagnostics

Figure A4 reports the distribution of portfolio concentration, measured by the Herfindahl–Hirschman Index (HHI), together with the distribution of absolute active weights relative to an Equal-Weight benchmark.

Across both universes, the HHI distributions are centered above the Equal-Weight benchmark, reflecting intentional deviations from equal weighting when predictive signals justify tilts. Importantly, the distributions remain far from extreme concentration levels, indicating that the strategy avoids excessive reliance on a small number of assets.

The active-weight distributions further show that most position adjustments are modest in magnitude, with large deviations occurring infrequently. This pattern suggests

that the framework primarily makes incremental reallocations around a diversified baseline rather than relying on aggressive position sizing.

In summary, the concentration and active-weight diagnostics provide additional evidence that the proposed framework generates adaptive yet well-controlled portfolios, consistent with real-world portfolio construction and execution constraints.

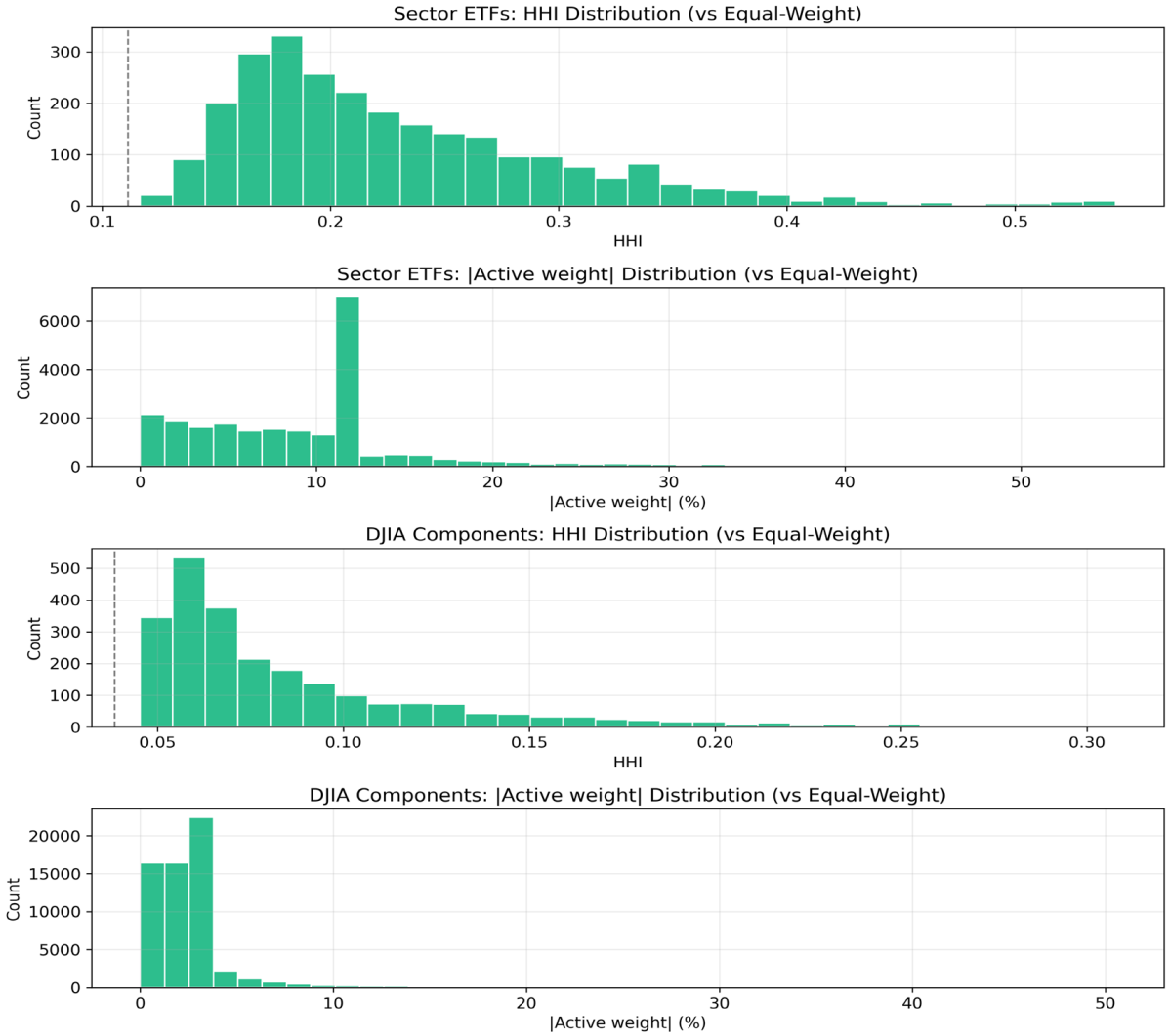


Figure A4. Portfolio concentration and active-weight diagnostics. The figure reports the empirical distribution of portfolio concentration and position deviations relative to an Equal-Weight benchmark for the Sector ETF and DJIA universes. Concentration is measured using the Herfindahl–Hirschman Index (HHI), defined as $\mathrm{HHI}_t = \sum_i w_{t,i}^2$, with dashed vertical lines indicating the Equal-Weight benchmark $1/N$. Active weights are defined as deviations from equal weighting, $a_{t,i} = w_{t,i} - 1/N$, and are summarized using the distribution of absolute active weights $|a_{t,i}|$. The results illustrate controlled and economically reasonable deviations from equal weighting while avoiding extreme concentration or position sizing.

References

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

Black, F., & Litterman, R. (1992). Global portfolio optimization. *Financial Analysts Journal*, *48*(5), 28–43. [CrossRef]

Brandt, M. W., Santa-Clara, P., & Valkanov, R. (2009). Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns. *The Review of Financial Studies*, *22*(9), 3411–3447. [CrossRef]

Chang, K. C., Tian, Z., & Yu, J. (2017, July 10–13). *Dynamic asset allocation—Chasing a moving target*. 2017 20th International Conference on Information Fusion (Fusion) (pp. 1–8), Xi'an, China. [CrossRef]

Chaves, D., Hsu, J., Li, F., & Shakernia, O. (2011). Risk parity portfolio vs. other asset allocation heuristic portfolios. *The Journal of Investing*, *20*(1), 108–118. [CrossRef]

Chen, L., Pelger, M., & Zhu, J. (2024). Deep learning in asset pricing. *Management Science*, *70*(2), 714–750. [CrossRef]

Chen, T., & Guestrin, C. (2016, August 13–17). *XGBoost: A scalable tree boosting system*. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'16 (pp. 785–794), San Francisco, CA, USA. [CrossRef]

Chen, Y., & Israelov, R. (2024a). Alpha now, taxes later: Tax-efficient long-only factor investing. *The Journal of Investing*, *33*(6), 10-3905. [CrossRef]

Chen, Y., & Israelov, R. (2024b). Exclude with impunity: Personalized indexing and stock restrictions. *Financial Analysts Journal*, *80*(2), 7–25. [CrossRef]

Chen, Y., & Israelov, R. (2024c). Income illusions: Challenging the high yield stock narrative. *Journal of Asset Management*, *25*(2), 190–202. [CrossRef]

Choudhary, H., Orra, A., Sahoo, K., & Thakur, M. (2025). Risk-adjusted deep reinforcement learning for portfolio optimization: A multi-reward approach. *International Journal of Computational Intelligence Systems*, *18*(1), 126. [CrossRef]

de-la-Rica-Escudero, A., Garrido-Merchán, E. C., & Coronado-Vaca, M. (2025). Explainable post hoc portfolio management financial policy of a Deep Reinforcement Learning agent. *PLoS ONE*, *20*(1), e0315528. [CrossRef]

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, *22*(5), 1915–1953. [CrossRef]

Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. *The Journal of Finance*, *47*(2), 427–465. [CrossRef]

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3–56. [CrossRef]

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, *116*(1), 1–22. [CrossRef]

Feng, G., Giglio, S., & Xiu, D. (2020). Taming the factor zoo: A test of new factors. *The Journal of Finance*, *75*(3), 1327–1370. [CrossRef]

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. [CrossRef]

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. [CrossRef]

Gao, S., Wang, Y., & Yang, X. (2023, August 19–25). *StockFormer: Learning hybrid trading machines with predictive coding*. Thirty-Second International Joint Conference on Artificial Intelligence (pp. 4766–4774), Macao, China. [CrossRef]

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273. [CrossRef]

Harvey, C. R., Liechty, J. C., Liechty, M. W., & Müller, P. (2010). Portfolio selection with higher moments. *Quantitative Finance*, *10*(5), 469–485. [CrossRef]

Hurley, W. J., & Brimberg, J. (2015). A note on the sensitivity of the strategic asset allocation problem. *Operations Research Perspectives*, *2*, 133–136. [CrossRef]

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

Kehinde, T. O., Chung, S.-H., & Olanrewaju, O. A. (2025). Inverse DEA for portfolio volatility targeting: Industry evidence from Taiwan stock exchange. *International Journal of Financial Studies*, *13*(4), 192. [CrossRef]

Kelly, B. T., Pruitt, S., & Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, *134*(3), 501–524. [CrossRef]

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689–702. [CrossRef]

Liu, X.-Y., Xiong, Z., Zhong, S., Yang, H., & Walid, A. (2022). Practical deep reinforcement learning approach for stock trading. *arXiv*, arXiv:1811.07522. [CrossRef]

Liu, X.-Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B., & Wang, C. (2020). FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. *arXiv*, arXiv:2011.09607. [CrossRef]

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*(1), 56–67. [CrossRef]

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv*, arXiv:1705.07874. [CrossRef]

Man Group. (2023, August 23). *Shining light into the machine learning black box*. Available online: https://www.man.com/insights/shining-light-machine-learning (accessed on 18 November 2025).

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

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, *17*(5–6), 441–470. [CrossRef]

Mugerman, Y., Hecht, Y., & Wiener, Z. (2019). On the failure of mutual fund industry regulation. *Emerging Markets Review*, *38*, 51–72. [CrossRef]

Oyewola, D., Omotehinwa, T. O., Kehinde, T., & Hammawa, Y. (2026). IAOA-LSTM: A hybrid model for stock portfolio optimization. *Computational Statistics*, *41*, 31. [CrossRef]

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *The Journal of Risk*, *2*(3), 21–41. [CrossRef]

Rockafellar, R. T., & Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, *26*(7), 1443–1471. [CrossRef]

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, *13*(3), 341–360. [CrossRef]

Roy, A. D. (1952). Safety first and the holding of assets. *Econometrica*, *20*(3), 431–449. [CrossRef]

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*, arXiv:1707.06347. [CrossRef]

Shalit, H. (2021). The Shapley value decomposition of optimal portfolios. *Annals of Finance*, *17*(1), 1–25. [CrossRef]

Shapley, L. S. (1953). 17. A value for n-person games. In H. W. Kuhn, & A. W. Tucker (Eds.), *Contributions to the theory of games (AM-28), Volume II* (pp. 307–318). Princeton University Press. [CrossRef]

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, *19*(3), 425–442. [CrossRef] [PubMed]

Sun, Q., Wei, X., & Yang, X. (2024). GraphSAGE with deep reinforcement learning for financial portfolio optimization. *Expert Systems with Applications*, *238*, 122027. [CrossRef]

Welch, I., & Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, *21*(4), 1455–1508. [CrossRef]

Yu, J., & Chang, K.-C. (2020). Neural network predictive modeling on dynamic portfolio management—A simulation-based portfolio optimization approach. *Journal of Risk and Financial Management*, *13*(11), 285. [CrossRef]

Yu, J., & Chang, K.-C. (2025). Smart tangency portfolio: Deep reinforcement learning for dynamic rebalancing and risk–return trade-off. *International Journal of Financial Studies*, *13*(4), 227. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.