ALPHAROUTER: AN EXPERT ROUTING APPROACH FOR ADAPTIVE PORTFOLIO SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Portfolio selection requires adapting asset allocations to changing market conditions while controlling risk and satisfying investment constraints. Existing learning-based methods often predict asset weights directly from market data, making portfolio decisions sensitive to noisy and nonstationary observations. We propose **AlphaRouter**, an adaptive portfolio selection framework that separates portfolio construction from market-dependent selection. AlphaRouter maintains a set of complementary experts, each capturing distinct historical market patterns to produce a complete portfolio under shared investment constraints. To overcome the high optimization variance of single neural networks in noisy markets, an ensemble of independently trained Transformer routers evaluates expert suitability from recent states, aggregating their calibrated decisions through majority voting to form a robust consensus. A historically best-performing expert serves as a stable reference, and routing adjustments are applied only when router consensus is sufficiently confident, mitigating unstable switching caused by short-term noise. By reducing the learned decision from high-dimensional weight prediction to expert-level selection, AlphaRouter guarantees constraint satisfaction by construction and improves robustness to noisy, nonstationary markets. Experiments on DJIA, HSI, CSI, and CRYPTO show that AlphaRouter achieves the best performance among the compared methods in cumulative wealth, annualized percentage yield, annualized Sharpe ratio, and Calmar ratio.

1 INTRODUCTION

Portfolio selection aims to allocate capital across multiple assets to achieve desirable returns while controlling investment risk (Xu et al., 2020; Chen et al., 2025; Zhang et al., 2026). Classical methods such as mean-variance optimization construct portfolios from expected returns, volatility, and asset correlations, offering clear financial interpretations (López de Prado, 2016; Cao et al., 2024). However, these estimated quantities may shift substantially over time (Li & Hai, 2025). Financial markets are inherently noisy, asset relationships vary across periods, and sudden market movements can rapidly alter the risk–return profile of a portfolio (Du et al., 2021; Koh et al., 2021; Ma & Huang, 2026; Soun et al., 2026). An effective portfolio strategy must therefore not only identify profitable opportunities but also maintain reasonable diversification and adapt its allocation as market conditions evolve.

Recent learning-based methods address these challenges from complementary directions (Wang et al., 2021; Soleymani & Paquet, 2021; Gao et al., 2023; Lien et al., 2023; Li et al., 2025; Zhu et al., 2026; Sun et al., 2026). Alpha-extraction methods learn more reliable predictive signals from noisy observations before constructing portfolios (Gollart & Okhrin, 2025; Li et al., 2025; Pan et al., 2026). Reinforcement learning approaches improve generalization by augmenting market samples or estimating time-varying risk to constrain portfolio weights (Lee et al., 2025; Jiang et al., 2025; Xue & Ye, 2026; Yuan et al., 2026; Lee & Kang, 2026). Mixture-of-experts frameworks train multiple trading policies and dynamically select among them to reduce single-model uncertainty (Sun et al., 2023; Yuan et al., 2026). Despite these advances, two fundamental limitations persist. First, most methods make decisions directly at the asset level (Jiang et al., 2017; Xu et al., 2020; Wang et al., 2021; Lee & Kang, 2026). The model must determine a high-dimensional continuous weight vector while simultaneously satisfying return objectives, risk budgets, transaction-cost constraints, and investment regulations. As the number of assets grows, this joint optimization becomes increas-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ingly difficult and sensitive to noisy predictions. Second, methods that dynamically switch among learned strategies typically rely on short-term model outputs without an explicit stability mechanism (Sun et al., 2023; Yuan et al., 2026). In noisy markets, frequent strategy changes can lead to unstable allocations and excessive turnover. At their core, both limitations stem from a common design choice: coupling portfolio construction with market-dependent selection forces the learning model to handle high-dimensional weight prediction, constraint satisfaction, and market adaptation simultaneously, making the overall decision unnecessarily complex.

To address these issues, we propose **AlphaRouter**, an adaptive portfolio selection framework that explicitly separates portfolio construction from market-dependent selection. AlphaRouter maintains a set of complementary portfolio experts, each capturing distinct historical market patterns to produce a complete feasible portfolio under shared investment constraints. Because single neural networks often yield brittle routing decisions on low signal-to-noise financial data, we employ an ensemble of independently trained Transformer routers to evaluate expert suitability from recent market states. By aggregating their outputs, the framework forms a robust selection consensus among the candidate portfolios. To mitigate unstable switching caused by short-term noise, a historically best-performing expert serves as a stable reference, and routing adjustments are applied only when router consensus is sufficiently confident. The main contributions are summarized as follows:

- We formulate portfolio adaptation as routing among complete feasible portfolios. Each expert independently constructs a constrained asset allocation, reducing the learned decision from high-dimensional continuous weight prediction to selection among a small set of complete portfolios.
- We develop a stability-aware routing mechanism based on an ensemble of independently trained Transformer routers, explicitly addressing the high optimization variance of single neural networks in noisy markets by requiring consensus. Their calibrated decisions are combined with a historically best-performing reference expert, allowing the portfolio to adapt to market changes while limiting unnecessary expert switching.

Experiments on DJIA, HSI, CSI, and CRYPTO show that AlphaRouter achieves the best performance among the compared methods in Cumulative Wealth, Annualized Percentage Yield, Annualized Sharpe Ratio, and Calmar Ratio.

2 Problem Formulation

Consider N risky assets observed over trading dates $t = 1, \ldots, T$. Let $\mathbf{x}_t \in \mathbb{R}^N$ denote the vector of close-to-close returns realized after the portfolio decision at date t. AlphaRouter formulates portfolio selection as routing among a set of K complete feasible portfolios. Each candidate portfolio is constructed by a portfolio expert and satisfies common investment constraints.

The feasible portfolio set is defined as a capped simplex:

$$\Delta_N^{(u)} = \left\{ \mathbf{w} \in \mathbb{R}^N \;\middle|\; \mathbf{1}^\top \mathbf{w} = 1, \quad 0 \le w_i \le u_N \right\}, \qquad u_N = \min\left(1, \frac{4}{N}\right), \tag{1}$$

where $\mathbf{w}$ is the portfolio weight vector, w_i denotes the weight allocated to asset i, and u_N is a per-asset upper bound that prevents excessive concentration.

At each trading date t, the candidate portfolio generated by expert k is denoted as $\mathbf{w}_t^{(k)} \in \Delta_N^{(u)}$, where $k \in \{1, \ldots, K\}$. The final portfolio is determined by routing among these feasible experts:

$$\mathbf{w}_t = \sum_{k=1}^{K} g_t^{(k)} \mathbf{w}_t^{(k)}, \qquad g_t^{(k)} \ge 0, \quad \sum_{k=1}^{K} g_t^{(k)} = 1, \tag{2}$$

where $\mathbf{g}_t$ denotes the routing weights over the candidate portfolios determined by AlphaRouter.

The decision uses only information available before $\mathbf{x}_t$ is realized. Let $\widetilde{\mathbf{w}}_{t-1}$ denote the previous allocation after price movement. With transaction cost rate c, the net portfolio return follows:

$$r_t = \left(1 + \mathbf{w}_t^\top \mathbf{x}_t\right)\left(1 - c\|\mathbf{w}_t - \widetilde{\mathbf{w}}_{t-1}\|_1\right) - 1. \tag{3}$$

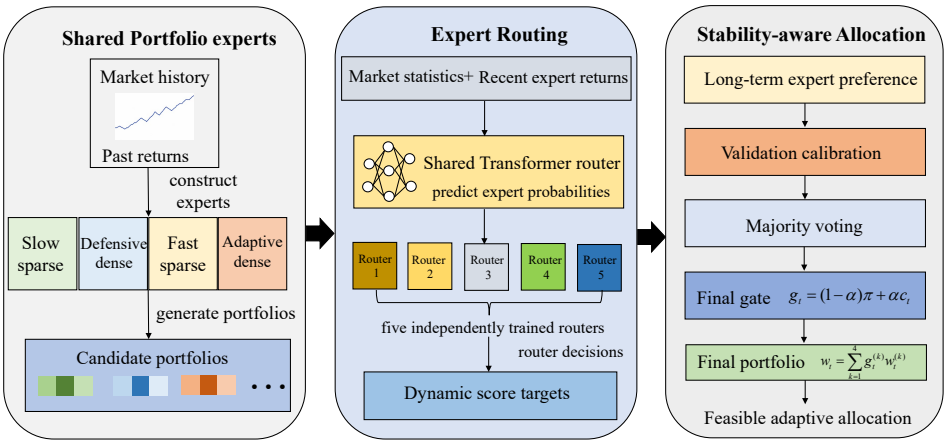

Figure 1: Overall framework of AlphaRouter.

The drifted allocation after observing market returns is:

$$\widetilde{w}_{t,i} = \frac{w_{t,i}(1 + x_{t,i})}{\sum_{j=1}^{N} w_{t,j}(1 + x_{t,j})}, \tag{4}$$

where $x_{t,i}$ is the realized return of asset i at date t.

3 METHOD

3.1 OVERALL FRAMEWORK

As illustrated in Fig. 1, **AlphaRouter** consists of three stages: shared portfolio construction, expert routing, and stability-aware allocation. The key idea is to separate the construction of asset-level portfolios from their market-dependent selection. Given historical market information, four shared experts first generate complete candidate portfolios with different investment characteristics. Multiple independently trained Transformer routers then evaluate the suitability of these candidate portfolios according to recent market states and expert returns. Their routing decisions are aggregated through validation-calibrated majority voting. Finally, the selected routed expert is combined with the historically best-performing expert to determine the final allocation. Because each candidate portfolio already satisfies the investment constraints, the neural routers only need to learn a selection decision rather than a high-dimensional weight prediction.

3.2 SHARED PORTFOLIO EXPERTS

Let $\mathcal{H}_t$ denote the historical information available before trading date t. The four shared experts generate:

$$\mathbf{w}_t^{(k)} = A_k(\mathcal{H}_t), \qquad k \in \{1, 2, 3, 4\}, \qquad \mathbf{w}_t^{(k)} \in \Delta_N^{(u)}, \tag{5}$$

where A_k denotes the portfolio construction rule of expert k, and $\Delta_N^{(u)}$ is the feasible set defined in Eq. 1. The four experts are slow sparse, defensive dense, fast sparse, and adaptive dense. The same expert definitions and parameter settings are used across all markets, while the four experts differ in their lookback, rebalance, and risk-control parameters.

The two sparse experts construct concentrated portfolio preferences from a risk-adjusted trend score. For $e \in \{\text{slow}, \text{fast}\}$:

$$z_{t,i}^{(e)} = \frac{R_{t,i}^{(e)}}{\widehat{\sigma}_{t,i}^{(e)} + \epsilon}, \qquad v_{t,i}^{(e)} = \frac{\mathbb{I}(i \in \mathcal{I}_t^{(e)})}{\widehat{\sigma}_{t,i}^{(e)} + \epsilon}, \tag{6}$$

where $R_{t,i}^{(e)}$ is the cumulative return computed from the expert-specific historical window, $\widehat{\sigma}_{t,i}^{(e)}$ is the corresponding historical volatility, $\mathcal{I}_t^{(e)}$ contains the highest-ranked assets according to $z_{t,i}^{(e)}$, and

ϵ is a small constant for numerical stability. The resulting preference vector is projected onto the common feasible set:

$$\mathbf{w}_t^{(e)} = \Pi_{\Delta_N^{(u)}}\left(\mathbf{v}_t^{(e)}\right), \tag{7}$$

where $\Pi_{\Delta_N^{(u)}}$ denotes the Euclidean projection onto $\Delta_N^{(u)}$, the ranked set $\mathcal{I}_t^{(e)}$ determines the concentrated preference before the common investment constraints are enforced. To ensure feasibility under the box-constrained simplex $\Delta_N^{(u)}$, the portfolio projection allocates any residual budget necessitated by the upper bound u_N to sequentially ranked assets. The term "sparse" therefore refers to targeted asset filtering prior to projection, distinguishing our convex formulation from computationally intractable ℓ_0-constrained cardinality penalties. Functionally, *slow sparse* extracts medium-term momentum from a concentrated top decile, whereas *fast sparse* adapts to recent market shifts across a broader top quartile.

The two dense experts optimize over the full asset universe without explicit asset preselection and balance estimated return against portfolio risk. For $e \in \{\mathrm{def}, \mathrm{adp}\}$:

$$\mathbf{w}_t^{(e)} = \arg\min_{\mathbf{w}\in\Delta_N^{(u)}} \left[-\widehat{\boldsymbol{\mu}}_t^{(e)\top}\mathbf{w} + \gamma_e \mathbf{w}^\top \overline{\boldsymbol{\Sigma}}_t^{(e)} \mathbf{w}\right], \tag{8}$$

where

$$\overline{\boldsymbol{\Sigma}}_t^{(e)} = (1-\rho_e)\widehat{\boldsymbol{\Sigma}}_t^{(e)} + \rho_e \operatorname{diag}\left(\widehat{\boldsymbol{\Sigma}}_t^{(e)}\right). \tag{9}$$

Here, $\widehat{\boldsymbol{\mu}}_t^{(e)}$ and $\widehat{\boldsymbol{\Sigma}}_t^{(e)}$ are estimated using the lookback window of expert e, γ_e controls risk aversion, and ρ_e controls covariance shrinkage. Defensive dense uses a longer estimation window, stronger risk control, and stronger diagonal shrinkage, whereas adaptive dense uses a shorter window and weaker risk aversion to respond more rapidly to recent market changes.

These four experts therefore provide complementary candidate portfolios across two dimensions: concentrated versus diversified allocation, and slow versus fast (adaptive) response. To prevent look-ahead bias, we fix all hyperparameters before testing (detailed in Appendix B), and provide an ablation study on the necessity of each expert in Appendix C.1.

3.3 Expert Routing

The Transformer routers determine which candidate portfolio is most suitable for the current market condition. The router input contains the most recent L daily states:

$$\mathbf{S}_t = [\mathbf{s}_{t-L}, \ldots, \mathbf{s}_{t-1}], \qquad \mathbf{p}_t = \mathrm{Softmax}(f_\theta(\mathbf{S}_t)), \tag{10}$$

where f_θ is the Transformer router parameterized by θ, and $p_t^{(k)}$ denotes the predicted suitability of expert k. Each daily state $\mathbf{s}_t$ is a fixed-dimensional vector of strictly causal market statistics, such as returns, volatility, distributional shape, and market breadth, combined with the recent realized returns of the candidate experts. This uniform feature construction is applied across all evaluated markets. Because the large parameter space of θ inevitably leads to local optima, a single optimization trajectory often overfits transient market noise or idiosyncratic temporal patterns. Voting across multiple routers theoretically filters out these uncorrelated noise traps to capture robust market dynamics. We therefore independently train an ensemble of five routers with different random initializations. The prediction of router m is denoted $\mathbf{p}_{t,m}$, and their aggregation is detailed in Section 3.4.

The router is trained via supervised learning using the forward-realized performance of candidate portfolios. For each training date t over horizon h, the quality of expert k is measured by a composite performance score:

$$S_{t,h}^{(k)} = G_{t,h}^{(k)} - \lambda_d D_{t,h}^{(k)} - \lambda_m M_{t,h}^{(k)} - cT_{t,h}^{(k)} - \lambda_q C_{t,h}^{(k)}, \tag{11}$$

where $G_{t,h}^{(k)}$ is the cumulative portfolio return over the horizon, $D_{t,h}^{(k)}$ is the downside semi-deviation, $M_{t,h}^{(k)}$ is the maximum drawdown, $T_{t,h}^{(k)}$ is the cumulative turnover, and $C_{t,h}^{(k)}$ is the portfolio concentration. Exact definitions are provided in Appendix B.3.

The performance score of each expert is converted into a normalized probability distribution using a temperature-scaled softmax:

$$q_{t,h}^{(k)} = \frac{\exp\left(S_{t,h}^{(k)}/\tau\right)}{\sum_{j=1}^{4}\exp\left(S_{t,h}^{(j)}/\tau\right)}, \tag{12}$$

where τ is a temperature parameter controlling the sharpness of the target distribution. Because these performance scores are recomputed at every training date, the supervision is dynamic: the preferred expert can differ across dates and is not fixed per market. The final model uses a one-day horizon (h=1) selected on the validation set; longer horizons are examined in Appendix C.2.

The training objective combines a cross-entropy term that aligns the router output $\mathbf{p}_t$ with the soft target $\mathbf{q}_{t,h}$, and a regret term that penalizes the expected gap relative to the best expert:

$$\mathcal{L}_t = \underbrace{-\sum_{k=1}^{4} q_{t,h}^{(k)} \log p_t^{(k)}}_{\text{cross-entropy}} + \lambda_s \underbrace{\left(\max_k S_{t,h}^{(k)} - \sum_{k=1}^{4} p_t^{(k)} S_{t,h}^{(k)}\right)}_{\text{regret}}, \tag{13}$$

where λ_s controls the regret penalty weight. The cross-entropy term encourages the router to reproduce the performance ranking, while the regret term explicitly discourages allocating probability to suboptimal experts. Only forward outcomes within the training and validation periods are used to construct $S_{t,h}^{(k)}$; during testing, no supervisory target is computed and no future return is accessed.

3.4 Stability-aware Allocation

To mitigate unstable expert switching under nonstationary market noise (Soun et al., 2026; Yuan et al., 2026), AlphaRouter anchors the allocation to a historically dominant reference and permits tactical routing adjustments only when router consensus is sufficiently confident.

Long-term reference expert. We identify the reference expert by evaluating candidates over the multi-year training period using a composite criterion:

$$B^{(k)} = \mathrm{ASR}^{(k)} + 0.25\ \mathrm{CR}^{(k)} + 0.10\ \mathrm{APY}^{(k)}, \qquad k^\star = \arg\max_k B^{(k)}. \tag{14}$$

This formulation prioritizes risk-adjusted returns (ASR), rewards drawdown control (CR), and promotes sustained growth (APY). Because candidate experts exhibit distinct long-term performance profiles over historical data, the identification of $k^\star$ is robust to moderate variations in these ranking weights (the standalone historical trajectories are detailed in Appendix C.3). Let $\boldsymbol{\pi}$ denote the one-hot indicator for $k^\star$, which serves as the stable anchor.

Validation calibration and majority voting. For router $m \in \{1, \ldots, M\}$ (sensitivity to the number of routers is analyzed in Appendix C.2) predicting probabilities $\mathbf{p}_{t,m}$, we measure routing confidence by the top-two probability margin:

$$d_{t,m} = p_{t,m}^{[1]} - p_{t,m}^{[2]}. \tag{15}$$

The calibrated decision falls back to the stable reference whenever confidence fails to cross a validation-tuned threshold δ:

$$\widehat{k}_{t,m} = \begin{cases} \arg\max_k p_{t,m}^{(k)}, & d_{t,m} \geq \delta, \\ k^\star, & d_{t,m} < \delta. \end{cases} \tag{16}$$

The individual decisions are then aggregated by majority voting across the M routers:

$$k_t^{\mathrm{route}} = \arg\max_k \sum_{m=1}^{M} \mathbb{I}\left(\widehat{k}_{t,m} = k\right), \tag{17}$$

breaking ties by the highest ensemble-averaged probability. Let $\mathbf{c}_t$ denote the one-hot vector corresponding to k_t^{route}.

Final gate and portfolio allocation. The confidence threshold δ and the adaptive share $\alpha \in [0, 1]$ are determined by grid search on the validation set and frozen prior to test deployment. The final portfolio blend is constructed as:

$$\mathbf{g}_t = (1-\alpha)\boldsymbol{\pi} + \alpha \mathbf{c}_t, \qquad \mathbf{w}_t = \sum_{k=1}^{4} g_t^{(k)} \mathbf{w}_t^{(k)}. \tag{18}$$

The selected market-specific pairs (δ, α) and their resulting test-time switching behavior are reported in Appendix C.4, while the necessity of confidence calibration and the stability fallback is verified via ablation in Appendix C.6. Because candidate portfolios satisfy the capped-simplex constraint $\Delta_N^{(u)}$ (Eq. 1), their convex combination $\mathbf{w}_t$ guarantees feasibility by construction.

4 Experiments

4.1 Data and Evaluation Protocol

We evaluate AlphaRouter on four financial markets with different asset characteristics: DJIA, HSI, CSI, and CRYPTO. DJIA contains 29 U.S. equities, HSI contains 56 Hong Kong equities, CSI contains 98 Chinese A-share equities, and CRYPTO contains 18 cryptocurrencies. The equity datasets start from 2010, while the cryptocurrency dataset starts from 2018. The asset universe and trading dates are fixed before training, and the same processed data are used for AlphaRouter and all comparison methods.

All experiments follow a strictly chronological setting. The training period ends on 31 December 2022, the validation period covers 2023, and the test period is from 1 January 2024 to 31 December 2025. The detailed data splits are summarized in Table 1. Training data are used to optimize the routers, validation data are used exclusively for checkpoint selection and deployment calibration, and test data are used only for final evaluation. No sample whose supervision horizon crosses a split boundary is included in model training or validation.

Table 1: Datasets used in the experiments.

Market	Assets	Training labels	Validation dates	Test dates
DJIA	29	3242	250	502
HSI	56	3179	243	492
CSI	98	3128	242	485
CRYPTO	18	1796	365	731

All methods are evaluated under the same asset universe, long-only capped-simplex constraint, portfolio drift convention, and proportional transaction cost of 0.25% per unit turnover. On each trading date, the portfolio is constructed before observing the subsequent market return. No test-period information is used for training, model selection, or calibration.

For daily net returns $r_1, \ldots, r_T$, we report cumulative wealth (CW), annualized percentage yield (APY), annualized Sharpe ratio (ASR), and Calmar ratio (CR). CW and APY evaluate portfolio growth, while ASR and CR measure risk-adjusted performance:

$$\mathrm{CW} = \prod_{t=1}^{T}(1+r_t), \qquad \mathrm{APY} = \mathrm{CW}^{A/T} - 1, \qquad \mathrm{ASR} = \sqrt{A}\frac{\overline{r}}{\sigma(r)}, \qquad \mathrm{CR} = \frac{\mathrm{APY}}{\mathrm{MDD}}, \tag{19}$$

where $\overline{r}$ is the mean daily net return, $\sigma(r)$ is its standard deviation, MDD is the maximum drawdown over the test period, and $A = 252$ is used for all markets, including CRYPTO, so that annualized quantities remain directly comparable across the four evaluated datasets. Higher values are preferred for all metrics.

4.2 Comparison Methods and Implementation

We compare AlphaRouter with 19 representative portfolio strategies spanning four paradigms: 1) **Passive and momentum controls**: BAH (buy-and-hold), CRP (constant rebalancing / equal-weight), and Momentum (trend-following); 2) **Classical online portfolio selection**: DynamicCRP,

EG (Helmbold et al., 1998), ONS (Agarwal et al., 2006), UP (Cover, 1991), and mean-reversion methods OLMAR (Li & Hoi, 2012) and WMAMR (Gao & Zhang, 2013); 3) **Risk-oriented optimization**: InvVol (inverse-volatility), MinVar (minimum variance), MeanVar (mean–variance), ERC (Maillard et al., 2010), HRP (López de Prado, 2016), and NCO (López de Prado, 2019); 4) **Recent deep and reinforcement learning SOTAs**: MetaTrader (Yuan et al., 2026), RiskBound (Lee & Kang, 2026), AlphaMix (Sun et al., 2023), and AlphaGAT (Li et al., 2025). This comparison evaluates whether routing among feasible experts outperforms both classical heuristics and end-to-end continuous weight networks. Detailed descriptions of all 19 baselines are provided in Appendix B.6.

All methods follow the same causal trading protocol, investment constraints, transaction costs, and portfolio-drift convention. Recent learning-based baselines are implemented under this common setting instead of directly using reported numbers from their original papers. Stochastic methods use five fixed random seeds. AlphaRouter trains five independent Transformer routers with different initializations, each receiving the most recent $L = 30$ daily market states as input, and combines their validation-calibrated decisions through majority voting. Additional baseline descriptions, expert configurations, router architectures, and computational costs are provided in Appendix B.

4.3 Main Results

As reported in Table 2, AlphaRouter achieves the best performance across all 16 market–metric combinations. Because the framework blends a pre-constructed expert bank with learned routing, standalone expert results are detailed in Appendix C.3, confirming that the best individual expert varies by market and that AlphaRouter consistently outperforms every standalone candidate. On equity markets, AlphaRouter consistently surpasses the strongest external baseline, AlphaGAT. The margin is steady on DJIA and HSI, but becomes substantially larger on CSI (raising APY from 0.138 to 0.175 and ASR from 0.794 to 0.958), where market shifts make expert diversity especially critical. Notably on HSI, competing baselines only excel in isolated metrics (AlphaGAT in APY, HRP in ASR, and InvVol in CR); in contrast, AlphaRouter simultaneously dominates all metrics (ASR 1.524, CR 2.090), proving its superior balance between capital growth and risk mitigation. On CRYPTO, AlphaRouter achieves its most pronounced gains, improving CW from 3.408 (MeanVar) to 4.271 and CR from 1.477 to 1.944. Here, while the adaptive dense expert provides a solid foundation (CW 4.109), learned routing captures additional upside by opportunistically engaging the fast sparse expert.

Overall, the more volatile or heterogeneous the market, the greater the advantage of expert-level routing over high-dimensional weight prediction. Furthermore, stress tests in Appendix C.5 confirm that AlphaRouter remains profitable even under 100 bps transaction costs (a fourfold increase).

Fig. 2 further compares the cumulative wealth trajectories of all methods over the test period. On DJIA and HSI, AlphaRouter exhibits competitive and relatively stable wealth accumulation, remaining comparable to the strongest competing methods across most of the evaluation horizon. The advantage becomes more pronounced on CSI and CRYPTO, where AlphaRouter gradually establishes a clearer separation from the competing baselines as the test period progresses. Importantly, the performance gap is not driven by a single favorable episode near the end of the sample. Instead, the wealth trajectories indicate that the differences emerge progressively through repeated portfolio decisions over time. This pattern provides complementary evidence to the aggregate performance metrics, suggesting that AlphaRouter's gains are associated with sustained adaptation across changing market conditions rather than with an isolated terminal increase in portfolio value.

4.4 Ablation Study

We evaluate the contribution of each core component in Fig. 3 by comparing AlphaRouter against six ablation variants under identical assets, constraints, and cost settings (exact numerical results are tabulated in Appendix C.6).

Anchor stability versus adaptive routing. The long-term reference provides the primary source of stability. Removing the stability fallback causes severe performance degradation across all markets, most notably on CRYPTO, where CW drops precipitously from 4.271 to 1.596. This underscores that anchoring to a stable expert is essential to prevent short-term market noise from misguiding the portfolio. Conversely, disabling neural routing also consistently lowers performance

Table 2: Main test performance of portfolio selection methods. Results of stochastic methods are averaged over five random seeds. AlphaRouter uses five independently trained routers with validation-calibrated majority voting. Bold and underline denote the best and strongest external results, respectively. Higher values are better. Percentage improvements reported in the text are computed from unrounded results.

Strategy	DJIA				HSI				CSI				CRYPTO			
	CW	APY	ASR	CR	CW	APY	ASR	CR	CW	APY	ASR	CR	CW	APY	ASR	CR
BAH	1.277	0.131	0.899	0.726	1.386	0.182	0.835	0.976	1.178	0.089	0.568	0.533	1.654	0.189	0.601	0.418
CRP	1.354	0.164	1.165	1.054	1.474	0.220	1.041	1.327	1.265	0.130	0.755	0.824	1.474	0.143	0.518	0.286
DynamicCRP	1.370	0.171	1.213	1.098	1.458	0.213	1.027	1.285	1.231	0.114	0.697	0.682	1.505	0.151	0.532	0.320
EG	1.349	0.162	1.150	1.035	1.472	0.219	1.032	1.320	1.261	0.128	0.746	0.808	1.532	0.158	0.544	0.323
OLMAR	1.113	0.055	0.439	0.338	1.259	0.125	0.657	0.711	1.141	0.071	0.464	0.403	0.759	-0.091	0.095	-0.160
WMAMR	1.087	0.043	0.355	0.235	1.194	0.095	0.531	0.465	1.106	0.054	0.372	0.291	0.638	-0.143	-0.005	-0.218
ONS	1.134	0.065	0.492	0.338	1.118	0.059	0.357	0.212	0.928	-0.038	-0.092	-0.151	1.641	0.186	0.584	0.330
UP	1.349	0.162	1.152	1.038	1.472	0.219	1.036	1.327	1.263	0.129	0.751	0.816	1.522	0.156	0.540	0.317
Momentum	0.831	-0.089	-0.542	-0.341	1.002	0.001	0.144	0.002	0.846	-0.083	-0.281	-0.251	1.543	0.161	0.547	0.251
InvVol	1.309	0.145	1.108	0.975	1.500	0.231	1.191	1.532	1.263	0.129	0.791	0.894	1.646	0.187	0.599	0.401
MinVar	1.150	0.073	0.661	0.569	1.433	0.202	1.143	1.396	1.204	0.101	0.700	0.654	2.226	0.318	0.862	0.855
MeanVar	1.131	0.064	0.496	0.327	1.366	0.173	0.858	0.918	1.206	0.102	0.615	0.546	3.408	0.526	1.172	1.477
HRP	1.186	0.090	0.771	0.623	1.494	0.228	1.253	1.460	1.178	0.089	0.617	0.606	1.823	0.230	0.691	0.506
ERC	1.295	0.139	1.074	0.941	1.453	0.211	1.156	1.424	1.227	0.112	0.708	0.741	1.703	0.201	0.621	0.431
NCO	1.129	0.063	0.584	0.474	1.448	0.209	1.199	1.369	1.201	0.100	0.682	0.700	2.306	0.334	0.895	0.899
MetaTrader	1.354	0.164	1.167	1.055	1.475	0.220	1.043	1.331	1.266	0.130	0.757	0.828	1.479	0.144	0.521	0.289
RiskBound	1.354	0.164	1.165	1.054	1.475	0.220	1.042	1.329	1.266	0.130	0.756	0.825	1.476	0.144	0.519	0.288
AlphaMix	1.354	0.164	1.165	1.054	1.475	0.220	1.042	1.329	1.266	0.130	0.756	0.826	1.478	0.144	0.520	0.290
AlphaGAT	1.376	0.174	1.229	1.114	1.501	0.231	1.184	1.523	1.283	0.138	0.794	0.904	3.352	0.517	1.154	1.427
AlphaRouter	**1.386**	**0.178**	**1.234**	**1.147**	**1.514**	**0.237**	**1.524**	**2.090**	**1.365**	**0.175**	**0.958**	**1.064**	**4.271**	**0.650**	**1.407**	**1.944**

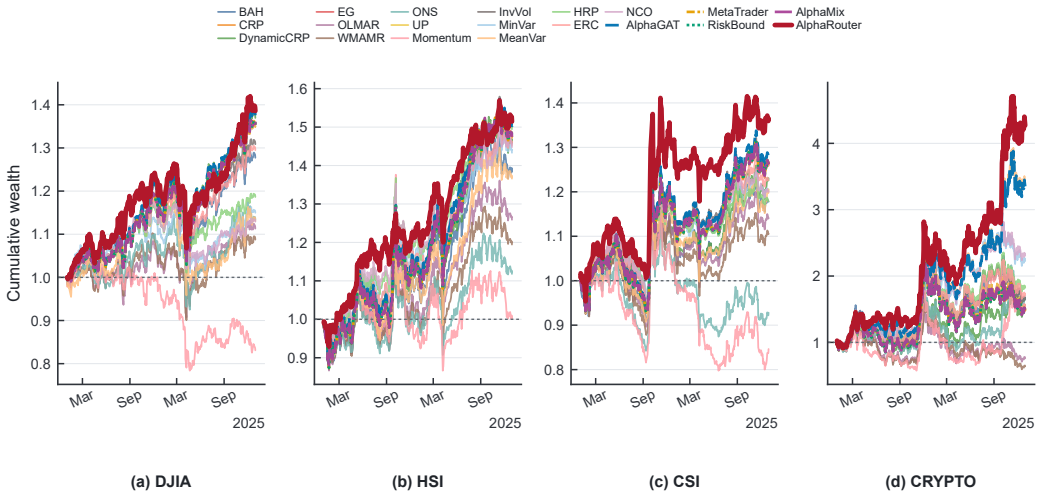

Figure 2: Cumulative wealth trajectories over the 2024–2025 test period. Results of stochastic baselines are averaged over five random seeds. AlphaRouter uses five independently trained routers with validation-calibrated majority voting.

(CRYPTO CW declines from 4.271 to 4.109), confirming that learned routing delivers a valuable, market-dependent tactical correction on top of the stable reference.

Ensemble robustness and decision mechanism. Multi-router aggregation further enhances reliability (individual router seed consistency across initializations is documented in Appendix C.7). The full ensemble outperforms the single-router mean in every market, raising CRYPTO CW from 3.793 to 4.271. Furthermore, hard majority voting surpasses soft probability consensus (CRYPTO

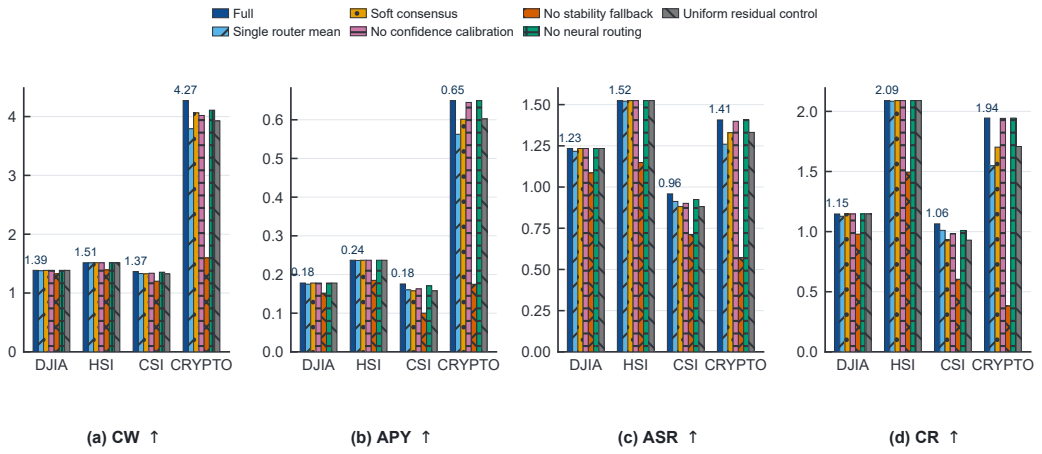

Figure 3: Ablation study of AlphaRouter. Single router mean reports the average performance of five independently deployed routers. Soft consensus replaces hard majority voting with probability averaging. No confidence calibration removes the routing confidence threshold. No stability fallback removes the long-term reference mechanism. No neural routing retains only the reference allocation. Uniform residual control replaces learned routing with a uniform residual allocation. Higher values are better.

CW 4.063), demonstrating that voting over calibrated expert choices handles predictive uncertainty more effectively than simple probability averaging.

Confidence filtering and residual control. Confidence calibration provides crucial noise filtering: eliminating the threshold δ decreases performance across all markets (CRYPTO CW drops to 4.018). Finally, learned routing consistently outperforms the uniform residual control, confirming that the performance gains arise from targeted market adaptation rather than naive diversification across candidate portfolios.

Overall, these ablation results validate the stability–adaptation decomposition of AlphaRouter. Crucially, as demonstrated in the CSI case study (Appendix C.8), this adaptive outperformance is achieved through sparse, controlled transitions rather than erratic, fee-inducing switching.

5 Conclusion

We propose **AlphaRouter** to overcome the instability of directly predicting high-dimensional asset weights in noisy, nonstationary markets. By decoupling portfolio construction from market-dependent adaptation, it constructs four shared portfolio experts with distinct characteristics, each producing a complete portfolio satisfying shared constraints. Multiple independently trained Transformer routers analyze recent market states, and their calibrated decisions are aggregated through majority voting. To mitigate unstable switching caused by short-term noise, AlphaRouter anchors the allocation to a historically best-performing reference expert. Tactical routing adjustments are applied only when router consensus is sufficiently confident. Because the learning task reduces to selecting among pre-constructed feasible portfolios, constraint satisfaction is guaranteed by construction. Experiments on DJIA, HSI, CSI, and CRYPTO confirm that AlphaRouter outperforms compared methods in cumulative wealth, annualized percentage yield, annualized Sharpe ratio, and Calmar ratio.

Although AlphaRouter achieves favorable performance, several aspects remain unexplored. First, while validation activates routing in all four markets, the adaptive share remains deliberately limited (10–20%) with only one to three expert transitions per test period. This supports our stability-aware design, but suggests future mechanisms could dynamically scale the magnitude of deviation, rather than relying on a static share. Second, extending AlphaRouter to additional markets and live trading settings will further evaluate its robustness under real-world execution constraints.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

We employed LLMs solely for language polishing and assistance with simple coding tasks.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. Pmlr, 2017.

Amit Agarwal, Elad Hazan, Satyen Kale, and Robert E. Schapire. Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 9–16, 2006.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez Paz. Invariant risk minimization. *arXiv preprint arXiv 1907.02893*, 2019. URL https://arxiv.org/abs/1907.02893.

Xinwei Cao, Yiguo Yang, Shuai Li, Predrag S Stanimirović, and Vasilios N Katsikis. Artificial neural dynamics for portfolio allocation: An optimization perspective. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 55(3):1960–1971, 2024.

Chaoyu Chen, Yupeng Fang, Huichou Huang, Ruirui Liu, Haoxian Liu, and Qingyao Wu. Deep portfolio selection with contrastively aligned cross-modal attention. *Pattern Recognition*, pp. 112305, 2025.

Thomas M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The review of Financial studies*, 22(5):1915–1953, 2009.

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pp. 402–411, 2021.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pp. 1050–1059, 2016.

Li Gao and Weiguo Zhang. Weighted moving average passive aggressive algorithm for online portfolio selection. In *2013 5th international conference on intelligent human-machine systems and cybernetics*, volume 1, pp. 327–330. IEEE, 2013.

Siyu Gao, Yunbo Wang, and Xiaokang Yang. Stockformer learning hybrid trading machines with predictive coding. In *Proceedings of the Thirty Second International Joint Conference on Artificial Intelligence*, pp. 4766–4774, 2023.

Maximilian Gollart and Yarema Okhrin. A reinforcement learning approach to dynamic portfolio optimization. *Annals of Operations Research*, pp. 1–26, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1321–1330, 2017.

David P. Helmbold, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. On line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Yifu Jiang, Jose Olmo, and Majed Atwi. High-dimensional multi-period portfolio allocation using deep reinforcement learning. *International Review of Economics & Finance*, 98:103996, 2025.

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

Thomas N. Kipf and Max Welling. Semi supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard L. Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Bert A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds a benchmark of in the wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5637–5664, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6402–6413, 2017.

Olivier Ledoit and Michael Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.

Hosung Lee and U Kang. Riskbound risk aware boundary guided portfolio optimization via action space reshaping. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2473–2484, 2026.

Junhyeong Lee, Haeun Jeon, Hyunglip Bae, and Yongjae Lee. Return prediction for mean-variance portfolio selection: How decision-focused learning shapes forecasting models. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 114–122, 2025.

Bin Li and Steven C. H. Hoi. On line portfolio selection with moving average reversion. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 273–280, 2012.

Bin Li and Steven CH Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):1–36, 2014.

Bin Li, Peilin Zhao, Steven C. H. Hoi, and Vivekanand Gopalkrishnan. Pamr passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 87(2):221–258, 2012.

Haifeng Li and Mo Hai. Portfoliozero: a stock portfolio model based on deep reinforcement learning. *Applied Soft Computing*, pp. 113578, 2025.

Shicheng Li, Jinshan Zhang, and Feng Wang. Alphagat a two stage learning approach for adaptive portfolio selection. In *Proceedings of the Thirty Fourth International Joint Conference on Artificial Intelligence*, pp. 7500–7508, 2025.

Yun Hsuan Lien, Yuan Kui Li, and Yu Shuen Wang. Contrastive learning and reward smoothing for deep portfolio management. In *Proceedings of the Thirty Second International Joint Conference on Artificial Intelligence*, pp. 3966–3974, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Marcos López de Prado. Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, 42(4):59–69, 2016.

Marcos López de Prado. A robust estimator of the efficient frontier. *SSRN Electronic Journal*, 2019.

Shaocong Ma and Heng Huang. Robust reinforcement learning in finance: Modeling market impact with elliptic uncertainty sets. *Advances in Neural Information Processing Systems*, 38:42156–42196, 2026.

Sébastien Maillard, Thierry Roncalli, and Jérome Teiletche. The properties of equally weighted risk contribution portfolios. *The Journal of Portfolio Management*, 36(4):60–70, 2010.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Chaofan Pan, Lingfei Ren, Linbo Xiong, Yonghao Li, Wei Wei, and Xin Yang. Regime-adaptive continual learning for portfolio management. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3859–3870, 2026.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual stage attention based recurrent neural network for time series prediction. In *Proceedings of the Twenty Sixth International Joint Conference on Artificial Intelligence*, pp. 2627–2633, 2017.

R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 497–504, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv 1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Farzan Soleymani and Eric Paquet. Deep graph convolutional reinforcement learning for financial portfolio management. *Expert Systems with Applications*, 182:115127, 2021.

Yejun Soun, Hosung Lee, Suyoung Park, and U Kang. Stable: Shift-tolerant allocation via black-litterman using conditional diffusion estimates. In *International Conference on Learning Representations*, volume 2026, pp. 46466–46491, 2026.

Shuo Sun, Xinrun Wang, Wanqi Xue, Xiaoxuan Lou, and Bo An. Mastering stock markets with efficient mixture of diversified trading experts. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2109–2119, 2023.

YJ Sun, ZY Lu, CH Liao, TS Dai, and SM Yuan. Optimizing stock portfolios with deep reinforcement learning after fnn-based fundamental analysis. *Quantitative Finance*, 26(1):137–159, 2026.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. Deeptrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 643–650, 2021.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer decomposition transformers with auto correlation for long term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet temporal 2d variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Ke Xu, Yifan Zhang, Deheng Ye, Peilin Zhao, and Mingkui Tan. Relation aware transformer for portfolio policy learning. In *Proceedings of the Twenty Ninth International Joint Conference on Artificial Intelligence*, pp. 4647–4653, 2020.

Pei Xue and Yuanchun Ye. Attention-enhanced reinforcement learning for dynamic portfolio optimization. *Intelligent Systems with Applications*, pp. 200622, 2026.

Haochen Yuan, Minting Pan, Yunbo Wang, Siyu Gao, and Xiaokang Yang. Metatrader learning to generalize reinforcement learning trading policies beyond offline data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28023–28031, 2026.

Ziliang Zhang, Tianming Zhao, and Albert Zomaya. Online portfolio selection with ml predictions. *Advances in Neural Information Processing Systems*, 38:82564–82584, 2026.

Xiaolin Zheng, Mengpu Liu, and Mengying Zhu. Deep hashing based dynamic stock correlation estimation via normalizing flow. In *Proceedings of the Thirty Second International Joint Conference on Artificial Intelligence*, pp. 4993–5001, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer beyond efficient transformer for long sequence time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer frequency enhanced decomposed transformer for long term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 27268–27286, 2022a.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022b.

Qing Zhu, Chenyu Han, Shan Liu, and Yuze Li. Dancing with markets: A dynamic rebalancing mechanism using deep reinforcement learning for online portfolio selection. *Information Fusion*, pp. 104452, 2026.

A Related Work

Classical and online portfolio selection. Classical portfolio selection constructs asset allocations from explicit return and risk estimates. Mean–variance optimization provides the fundamental formulation by balancing expected return against covariance-based risk (Chen et al., 2025). Subsequent studies improve robustness to estimation error through covariance shrinkage, naive diversification, equal risk contribution, and hierarchical risk parity (Ledoit & Wolf, 2003; DeMiguel et al., 2009; Maillard et al., 2010; López de Prado, 2016). Black–Litterman incorporates investor views into equilibrium returns, while conditional value-at-risk explicitly emphasizes downside risk (Rockafellar et al., 2000). These methods provide interpretable portfolio construction rules, but their behavior depends strongly on the assumed return and risk structure. Online portfolio selection instead updates allocations sequentially as new market observations become available. Universal Portfolio integrates over constant rebalanced portfolios, Exponentiated Gradient updates portfolio weights from recent price relatives, and Online Newton Step introduces second-order information into the update (Cover, 1991; Helmbold et al., 1998; Agarwal et al., 2006). Mean-reversion methods such as PAMR, OLMAR, and weighted moving-average mean reversion exploit short-term price reversal using different update rules (Li et al., 2012; Li & Hoi, 2012; Gao & Zhang, 2013). As summarized by Li & Hoi (2014), these methods differ mainly in their market assumptions and optimization mechanisms. Although such deterministic strategies remain competitive baselines, no single construction rule is expected to respond equally well to all market conditions. AlphaRouter therefore maintains multiple portfolio experts with complementary investment characteristics and performs adaptation by selecting among their complete feasible portfolios.

Learning-based portfolio selection. Recent learning-based methods improve portfolio decisions by learning temporal and cross-asset representations from market observations. Temporal attention is used to identify informative historical signals, while graph-based models capture relations among securities (Qin et al., 2017; Kipf & Welling, 2017; Veličković et al., 2018; Sawhney et al., 2021; Zheng et al., 2023). Transformer-based time-series models further improve long-range representation through sparse attention, decomposition, frequency-domain modeling, patching, inverted tokens, and multiscale mixing (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022a; Nie et al., 2023; Wu et al., 2023; Liu et al., 2024; Wang et al., 2024). Adaptive recurrent models and studies of out-of-distribution generalization further highlight the importance of distribution shift when learned representations are deployed under changing data distributions (Du et al., 2021; Arjovsky et al., 2019; Koh et al., 2021).

Deep reinforcement learning extends representation learning to end-to-end portfolio policies. Existing methods directly predict portfolio actions or combine temporal and cross-asset information through attention, relational, graph, and predictive representations (Jiang et al., 2017; Xu et al., 2020; Wang et al., 2021; Soleymani & Paquet, 2021; Gao et al., 2023; Lien et al., 2023). PPO provides a general policy-gradient optimization framework, while constrained policy optimization introduces explicit constraints into policy learning (Schulman et al., 2017; Achiam et al., 2017). More recent approaches improve robustness and risk awareness in different ways. MetaTrader uses transformed market data and conservative learning to improve offline policy generalization, RiskBound predicts time-varying asset risk before restricting the continuous action space, and AlphaGAT first extracts temporal alpha information and then constructs allocations through graph-based policy learning (Yuan et al., 2026; Lee & Kang, 2026; Li et al., 2025). Despite these advances, many approaches ultimately construct the final portfolio through asset-level continuous decisions.

Mixture-of-experts provides the closest general framework to our method. Early mixture models jointly train multiple experts and a gating network, while sparse expert architectures activate only a subset of available experts for each input (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022; Zhou et al., 2022b). Deep ensembles and confidence calibration further reduce sensitivity to a single model initialization and provide mechanisms for handling predictive uncertainty (Lakshminarayanan et al., 2017; Guo et al., 2017; Gal & Ghahramani, 2016). In portfolio selection, AlphaMix combines multiple diversified trading experts through a learned mixture mechanism (Sun et al., 2023). AlphaRouter differs in that its experts output complete constrained portfolios, and routing is performed directly over these feasible portfolio candidates. Moreover, the historically best-performing expert serves as a stable reference, while the neural routers introduce adaptation

only when supported by validation-calibrated decisions. This design explicitly separates stable portfolio construction from market-dependent adjustment.

B Reproducibility and Implementation Details

This section provides implementation details for the portfolio experts, Transformer routers, training procedure, and chronological deployment. All model and deployment parameters are determined using the training and validation periods and remain fixed during testing.

B.1 Shared Portfolio Experts

AlphaRouter uses four portfolio experts across all markets: slow sparse, defensive dense, fast sparse, and adaptive dense. Table 3 reports the complete configuration of the four experts. Lookback, skip, and rebalance intervals are measured in trading dates. The top-fraction parameter applies only to the sparse experts, whereas risk aversion and covariance shrinkage apply only to the dense experts.

Table 3: Configuration of the four shared portfolio experts. The same expert definitions and parameters are used across all evaluated markets. Lookback, skip, and rebalance are measured in trading dates.

Expert	Lookback	Skip	Top frac.	Rebal.	Risk aversion	Shrinkage
Slow sparse	60	20	0.10	20	–	–
Defensive dense	220	–	–	10	32	0.75
Fast sparse	20	5	0.25	20	–	–
Adaptive dense	60	–	–	5	8	0.00

For the sparse experts, the top fraction determines the high-ranking preference set before projection onto the capped simplex. Because the per-asset upper bound may require additional positions to maintain full investment, the number of nonzero positions after projection can exceed the nominal top fraction. Thus, "sparse" refers to the concentrated preference used during portfolio construction rather than to an exact cardinality constraint on the executed portfolio. The skip parameter specifies the lag used in the trend calculation, and the rebalance interval determines how frequently the target portfolio is recomputed. Both sparse experts rebalance every 20 dates; "fast" refers to the shorter lookback and skip used in the trend calculation rather than to a higher rebalancing frequency.

For the dense experts, the risk-aversion parameter corresponds to γ_e in Eq. 8, while shrinkage corresponds to ρ_e in Eq. 9. Defensive dense uses a longer estimation window, stronger risk aversion, and stronger diagonal covariance shrinkage. Adaptive dense uses a shorter estimation window, weaker risk aversion, and more frequent rebalancing so that its candidate portfolio can respond more quickly to recent market changes. Between target rebalancing dates, holdings evolve according to the portfolio-drift convention in Eq. 4.

B.2 Transformer Router

After the four feasible candidate portfolios have been constructed, the Transformer routers evaluate their relative suitability from recent market states. All router inputs are constructed causally. Feature normalization parameters are estimated using training observations only and are subsequently fixed for validation and testing. The router receives neither market identity nor the return that is realized after the current portfolio decision.

The five routers use the same architecture and optimization settings. Their complete configuration is reported in Table 4.

The five fixed router seeds are 101, 202, 303, 404, and 505. Each router is trained for up to 100 epochs using the optimization settings specified in Table 4, and the model checkpoint achieving the highest portfolio score on the validation set is retained for deployment. All experiments are conducted on a single NVIDIA GeForce RTX 4070 GPU.

Table 4: Transformer architecture and optimization settings used by AlphaRouter. The five routers share the same configuration and differ only in their fixed random initialization.

Setting	Value	Setting	Value
Input window	30	Hidden dimension	128
Transformer layers	6	Attention heads	8
Feed-forward dimension	256	Dropout	0.10
Maximum epochs	100	Batch size	256
Learning rate	5×10^{-4}	Weight decay	10^{-4}
Gradient norm limit	1	Softmax temperature	0.75
Regret weight	0.25	Transaction cost	0.0025
Downside weight	0.25	Drawdown weight	0.10
Concentration weight	0.002	Trainable parameters	819,568

B.3 Dynamic Performance Targets

As described in Section 3, the router is supervised by the realized quality of the complete candidate portfolios rather than by a fixed expert identity. The performance score in Eq. 11 is recomputed for each labeled date and therefore allows the preferred expert to vary over time.

For completeness, we provide the exact definitions of the five components used in the composite score. Let

$$a_{t+\ell}^{(k)} = \left(\mathbf{w}_{t+\ell}^{(k)}\right)^{\top} \mathbf{x}_{t+\ell}, \qquad \ell = 0, \ldots, h-1, \tag{20}$$

denote the portfolio return of expert k before transaction cost at the ℓ-th date of the forward evaluation horizon. Starting from $V_{t,0}^{(k)} = 1$, the corresponding portfolio-value path is

$$V_{t,\ell}^{(k)} = V_{t,\ell-1}^{(k)} \left(1 + a_{t+\ell-1}^{(k)}\right), \qquad \ell = 1, \ldots, h. \tag{21}$$

The portfolio growth term is defined as the cumulative return over the evaluation horizon:

$$G_{t,h}^{(k)} = V_{t,h}^{(k)} - 1. \tag{22}$$

The downside-risk term measures the magnitude of negative portfolio returns:

$$D_{t,h}^{(k)} = \sqrt{\frac{1}{h} \sum_{\ell=0}^{h-1} \left[\min\left(a_{t+\ell}^{(k)}, 0\right)\right]^2}. \tag{23}$$

The maximum-drawdown term is computed from the same forward portfolio-value path:

$$M_{t,h}^{(k)} = \max_{1 \le \ell \le h} \left(1 - \frac{V_{t,\ell}^{(k)}}{\max_{0 \le j \le \ell} V_{t,j}^{(k)}}\right). \tag{24}$$

Let $\widetilde{\mathbf{w}}_{t+\ell-1}^{(k)}$ denote the allocation of expert k after the preceding price movement. The cumulative portfolio turnover is defined as

$$T_{t,h}^{(k)} = \sum_{\ell=0}^{h-1} \left\| \mathbf{w}_{t+\ell}^{(k)} - \widetilde{\mathbf{w}}_{t+\ell-1}^{(k)} \right\|_1. \tag{25}$$

Finally, portfolio concentration is measured by the squared asset weights, averaged over the evaluation horizon:

$$C_{t,h}^{(k)} = \frac{1}{h} \sum_{\ell=0}^{h-1} \sum_{i=1}^{N} \left(w_{t+\ell,i}^{(k)}\right)^2. \tag{26}$$

Accordingly, $G_{t,h}^{(k)}$ rewards portfolio growth, whereas $D_{t,h}^{(k)}$, $M_{t,h}^{(k)}$, $T_{t,h}^{(k)}$, and $C_{t,h}^{(k)}$ penalize downside risk, drawdown, turnover, and concentration, respectively. The corresponding coefficients are $\lambda_d = 0.25$, $\lambda_m = 0.10$, $c = 0.0025$, and $\lambda_q = 0.002$. The softmax temperature is $\tau = 0.75$. Note that at the deployed horizon $h = 1$ the downside and drawdown terms coincide, so they act as a single penalty with combined weight $\lambda_d + \lambda_m$. The general form is retained because the two terms become distinct for $h > 1$, as examined in Appendix C.2. All quantities are computed using the same portfolio-drift convention as the main portfolio simulator.

To verify that the resulting supervision indeed represents a dynamic routing problem, Table 5 reports the fraction of training dates on which each expert achieves the highest performance score, together with the fraction of adjacent labeled dates for which the top-performing expert changes. The reported results use the one-day evaluation horizon selected by validation.

Table 5: Distribution of the top-performing expert on the training split. Each expert column reports its fraction of argmax score targets. Change rate is the fraction of adjacent labeled dates whose top-performing experts are different.

Market	Slow sparse	Defensive dense	Fast sparse	Adaptive dense	Change rate
DJIA	0.255	0.244	0.257	0.243	0.741
HSI	0.236	0.268	0.232	0.264	0.712
CSI	0.234	0.248	0.242	0.277	0.725
CRYPTO	0.227	0.224	0.276	0.273	0.727

All four experts account for a substantial fraction of the top-performing targets in every market, so the supervision is not equivalent to assigning one fixed expert to each market. The top-performing expert also changes on roughly 72% of adjacent labeled dates, close to the 75% rate expected under an independent uniform assignment over four experts. The daily top-performing expert is therefore only weakly predictable, which is consistent with the modest label accuracies reported in Table 9. AlphaRouter is accordingly not designed to predict the daily argmax expert. The confidence margin δ, the stable reference, and majority voting are introduced precisely so that the executed portfolio responds only to the sparse subset of dates on which the routing evidence is consistent across independently trained routers.

B.4 In-Sample Training and Parameter Calibration

To ensure realistic backtesting and strictly prevent look-ahead bias, our empirical pipeline divides historical data into three non-overlapping chronological periods: in-sample training, validation calibration, and out-of-sample testing (Table 1). Forward-realized market returns are utilized strictly to construct dynamic supervision targets during the training phase; they are never observed as router inputs. The complete training and calibration procedure follows three sequential stages, summarized in Algorithm 1:

Stage 1: Base portfolio construction and forward scoring. Prior to neural routing, the four base portfolio experts $\{A_k\}_{k=1}^{4}$ precompute candidate asset allocations $\mathbf{w}_t^{(k)} \in \Delta_N^{(u)}$ causally using only market history available prior to date t. For each date t in the training and validation periods, we evaluate the forward risk-adjusted performance of each candidate over horizon h using the composite score $S_{t,h}^{(k)}$ (Eq. 11), which penalizes downside volatility, drawdown, turnover, and asset concentration. The raw scores are normalized into a soft target distribution $\mathbf{q}_{t,h}$ via temperature-scaled softmax (Eq. 12). Causal feature normalization parameters are fitted strictly on training data and subsequently frozen.

Stage 2: Multi-router training and validation checkpoint selection. To mitigate single-model optimization variance, we train an ensemble of $M = 5$ Transformer routers under independent random seeds $\mathcal{S} = \{101, 202, 303, 404, 505\}$. Each router is trained for up to 100 epochs to minimize the objective $\mathcal{L}_t$ (Eq. 13), which balances cross-entropy target matching against regret-based underperformance penalties. At the end of each epoch, portfolio performance is evaluated across the validation period net of transaction costs. For each seed $s \in \mathcal{S}$, the model checkpoint achieving the highest validation portfolio score is retained for deployment.

Stage 3: Benchmark anchor identification and grid calibration. Before out-of-sample deployment, AlphaRouter establishes its stability mechanisms. A stable benchmark expert $k^\star$ is selected based on multi-year historical performance over the training set using the composite metric $B^{(k)}$ (Eq. 14). Next, the confidence margin threshold δ and the tactical adaptive share $\alpha \in [0, 1]$ are calibrated via grid search on the validation set to optimize net risk-adjusted returns. Following this stage, the state normalizer, all five router checkpoints, the benchmark anchor $k^\star$, and the parameters (δ, α) are permanently frozen.

Algorithm 1 AlphaRouter in-sample training and parameter calibration

Require: Historical price data, base expert rules $\{A_k\}_{k=1}^4$, random seeds $\mathcal{S}$
Ensure: Trained router ensemble $\{f_{\theta_s}\}_{s\in\mathcal{S}}$, benchmark anchor $k^\star$, parameters (δ, α)

▷ **Stage 1: Base Portfolio Construction & Forward Target Generation**
Precompute candidate allocations $\mathbf{w}_t^{(k)} = A_k(\mathcal{H}_t) \in \Delta_N^{(u)}$ for all experts k across all dates
for each date t in training and validation sets **do**
    Evaluate forward performance score $S_{t,h}^{(k)}$ via Eq. 11 for $k \in \{1, \ldots, 4\}$
    Compute target distribution $q_{t,h}^{(k)} \propto \exp(S_{t,h}^{(k)}/\tau)$ via Eq. 12
end for
Fit causal feature normalizer on training set observations only
▷ **Stage 2: Independent Router Training & Checkpoint Selection**
for each random seed $s \in \mathcal{S}$ **do**
    Initialize Transformer router f_{θ_s}
    for each training epoch up to 100 **do**
        Optimize router parameters θ_s by minimizing loss $\mathcal{L}_t$ (Eq. 13)
        Simulate resulting portfolio returns across the validation period
    end for
    Retain checkpoint f_{θ_s} achieving the highest validation portfolio score
end for
▷ **Stage 3: Benchmark Identification & Parameter Calibration**
Select stable benchmark expert $k^\star \leftarrow \arg\max_k B^{(k)}$ over training history via Eq. 14
Calibrate confidence margin δ and adaptive share α via grid search on validation data
return Frozen normalizer, ensemble $\{f_{\theta_s}\}$, benchmark anchor $k^\star$, and parameters (δ, α)

B.5 Out-of-Sample Portfolio Execution

Out-of-sample backtesting simulates sequential online portfolio rebalancing in strict chronological order. On each trading date t, portfolio decisions are finalized causally before observing subsequent market returns $\mathbf{x}_t$. The daily trading cycle proceeds in four phases, summarized in Algorithm 2:

Step 1: Candidate portfolio generation. Prior to market opening at date t, each base expert generates its candidate weight vector $\mathbf{w}_t^{(k)} = A_k(\mathcal{H}_t)$ using only historical information available up to date $t-1$. By construction, all candidate portfolios strictly satisfy the capped-simplex constraint $\Delta_N^{(u)}$ (Eq. 1).

Step 2: Router inference and margin-based confidence gating. The market state $\mathbf{S}_t = [\mathbf{s}_{t-L}, \ldots, \mathbf{s}_{t-1}]$ is constructed from the trailing L trading dates and normalized. Each frozen router $m \in \{1, \ldots, M\}$ generates an expert probability vector $\mathbf{p}_{t,m} = \mathrm{Softmax}(f_{\theta_m}(\mathbf{S}_t))$. We assess conviction through the top-two probability margin $d_{t,m} = p_{t,m}^{[1]} - p_{t,m}^{[2]}$. If $d_{t,m} \geq \delta$, the router votes for its top candidate $\arg\max_k p_{t,m}^{(k)}$; otherwise, it refrains from speculative reallocation and defaults to the stable benchmark anchor $k^\star$ (Eq. 16).

Step 3: Majority voting and core-satellite asset allocation. The ensemble aggregates the M gated votes via majority consensus (Eq. 17) to select the tactical expert k_t^{route}. AlphaRouter then implements a core-satellite allocation structure via a convex combination:

$$\mathbf{w}_t = (1-\alpha)\mathbf{w}_t^{(k^\star)} + \alpha \mathbf{w}_t^{(k_t^{\text{route}})}, \tag{27}$$

where $(1-\alpha)$ constitutes the stable core allocation anchored in $k^\star$, and α represents the tactical satellite allocation directed to k_t^{route}. Because the capped simplex $\Delta_N^{(u)}$ is convex, the executed allocation $\mathbf{w}_t$ is guaranteed to satisfy all asset caps and budget constraints without post-hoc projection.

Step 4: Trade execution, turnover friction, and portfolio drift. After the target portfolio $\mathbf{w}_t$ is formed, close-to-close returns $\mathbf{x}_t$ are realized. Portfolio turnover friction is deducted at rate c per unit turnover to obtain the net return r_t (Eq. 3). Between rebalancing intervals, asset holdings passively drift with market prices to $\widetilde{\mathbf{w}}_t$ (Eq. 4), which serves as the base allocation for computing rebalancing turnover at date $t+1$.

Algorithm 2 AlphaRouter out-of-sample portfolio execution

Require: Frozen routers $\{f_{\theta_m}\}_{m=1}^{M}$, benchmark anchor $k^\star$, margin δ, adaptive share α, test data
Ensure: Target portfolio weights $\{\mathbf{w}_t\}_{t=1}^{T}$, realized net returns $\{r_t\}_{t=1}^{T}$

for each test trading date $t = 1, \ldots, T$ in chronological order **do**
  ▷ **Step 1: Candidate Portfolio Formation**
  Compute candidate allocations $\mathbf{w}_t^{(k)} = A_k(\mathcal{H}_t) \in \Delta_N^{(u)}$ using trailing market data
  ▷ **Step 2: Router Inference & Margin-Based Confidence Gating**
  Assemble causal market state $\mathbf{S}_t = [\mathbf{s}_{t-L}, \ldots, \mathbf{s}_{t-1}]$
  for each router $m \in \{1, \ldots, M\}$ **do**
    Compute expert probabilities $\mathbf{p}_{t,m} \leftarrow \mathrm{Softmax}(f_{\theta_m}(\mathbf{S}_t))$
    Measure confidence margin $d_{t,m} \leftarrow p_{t,m}^{[1]} - p_{t,m}^{[2]}$
    Apply gating: $\widehat{k}_{t,m} \leftarrow \arg\max_k p_{t,m}^{(k)}$ if $d_{t,m} \geq \delta$, else fallback to $k^\star$
  end for
  ▷ **Step 3: Majority Consensus & Core-Satellite Allocation**
  Determine tactical expert $k_t^{\text{route}} \leftarrow \mathrm{mode}(\widehat{k}_{t,1}, \ldots, \widehat{k}_{t,M})$ (break ties by ensemble mean)
  Execute convex blend: $\mathbf{w}_t \leftarrow (1-\alpha)\mathbf{w}_t^{(k^\star)} + \alpha \mathbf{w}_t^{(k_t^{\text{route}})}$
  ▷ **Step 4: Trade Execution, Cost Deduction, & Drift Update**
  Observe market return $\mathbf{x}_t$ and deduct turnover transaction cost c via Eq. 3
  Update drifted holdings $\widetilde{\mathbf{w}}_t$ via Eq. 4 for subsequent turnover calculation
end for
Compute summary performance metrics: CW, APY, ASR, and CR

B.6 Baseline Descriptions and Implementation Details

All 19 comparison methods are executed under our common empirical protocol using author-recommended hyperparameter settings from their original publications. Table 6 provides a structured taxonomy of all evaluated baselines across quantitative finance and deep reinforcement learning. To reflect realistic institutional risk control, all methods strictly enforce the common per-asset concentration cap ($u_N = \min(1, 4/N)$). Under these practical constraints, continuous reinforcement learning policies naturally produce conservative, highly diversified allocations, whereas AlphaRouter achieves targeted tactical adaptability by routing among pre-constructed compliant experts.

C Additional Experimental Analyses

The main experiments establish the overall performance of AlphaRouter and evaluate its principal components. This section first provides the complete numerical ablation results corresponding to Fig. 3 and then reports the standalone performance of the shared portfolio experts, which separates the contribution of expert construction from that of learned routing. It continues with additional analyses of hyperparameter sensitivity, routing behavior, transaction costs, optimization variability, and statistical uncertainty.

Table 6: Taxonomy of the 19 evaluated baseline portfolio strategies.

Category	Method	Key Mechanism	Seminal Reference
Passive & Trend	BAH	Buy-and-hold uniform portfolio without rebalancing	Market benchmark
	CRP	Daily constant rebalancing to equal asset weights ($1/N$)	Empirical baseline
	Momentum	Allocates to assets with highest trailing cumulative returns	Cross-sectional trend
Classical OLPS	DynamicCRP	Volatility-adaptive constant rebalancing	Adaptive heuristic
	EG	Multiplicative exponential gradient update on price relatives	Helmbold et al. (1998)
	ONS	Online convex optimization with second-order Hessian curvature	Agarwal et al. (2006)
	UP	Universal performance-weighted integration over rebalanced portfolios	Cover (1991)
	OLMAR	Passive-aggressive projection on rolling moving-average reversals	Li & Hoi (2012)
	WMAMR	Exponentially-decayed weighted moving-average mean reversion	Gao & Zhang (2013)
Risk Optimization	InvVol	Asset weighting inversely proportional to historical volatility	Heuristic risk parity
	MinVar	Global minimum portfolio variance under Ledoit–Wolf shrinkage	Ledoit & Wolf (2003)
	MeanVar	Classical Markowitz return-variance quadratic optimization	Chen et al. (2025)
	ERC	Equal risk contribution across all assets	Maillard et al. (2010)
	HRP	Hierarchical correlation-tree clustering without matrix inversion	López de Prado (2016)
	NCO	Nested clustered optimization across intra- and inter-cluster risks	López de Prado (2019)
Deep RL / SOTAs	AlphaMix	Mixture-of-experts trading policy via learned dynamic gating	Sun et al. (2023) (KDD)
	MetaTrader	Conservative offline reinforcement learning on transformed data	Yuan et al. (2026) (AAAI)
	RiskBound	Downside risk-bounded action space reshaping	Lee & Kang (2026) (KDD)
	AlphaGAT	Temporal alpha feature extraction with graph attention networks	Li et al. (2025) (IJCAI)

C.1 Expert-Bank Composition

The four experts form a 2×2 design along two dimensions: allocation structure (sparse versus dense) and response speed (slow versus adaptive). A natural question is whether all four experts are necessary. We address this by evaluating all $2^4 - 1 = 15$ non-empty subsets of the expert bank.

For each subset $\mathcal{S}$, the five frozen routers are retained and their output probabilities are conditionally renormalized over the available experts:

$$\widetilde{p}_{t,m}^{(k|\mathcal{S})} = \frac{p_{t,m}^{(k)}}{\sum_{j \in \mathcal{S}} p_{t,m}^{(j)}}, \qquad k \in \mathcal{S}. \tag{28}$$

The long-term reference is recomputed from training data for each subset, and the confidence margin and adaptive share are recalibrated on validation data. For $K = 1$, routing degenerates to the only available expert; confidence-margin gating, majority voting, and adaptive-share calibration are naturally skipped, and the result is treated as a single-expert baseline. This design varies only the set of available candidates while holding the learned router parameters fixed.

Table 7 summarizes the results by bank size. The test score first averages across the four markets and then across all subsets of a given size. The complete $K = 4$ bank achieves 1.7022, improving upon the mean $K = 3$ score of 1.5505 by 9.8%. Even the best $K = 3$ subset reaches only 1.6384, which is 3.9% below the full bank.

Table 7: Effect of expert-bank size on test performance. The test score is $\mathrm{ASR} + 0.25\,\mathrm{CR} + 0.10\,\mathrm{APY}$, averaged across the four markets. The mean and standard deviation are computed over all $\binom{4}{K}$ subsets. Best reports the highest observed subset score.

K	Subsets	Mean test score	Best test score
1	4	1.0928 ± 0.2307	1.4294
2	6	1.3481 ± 0.1141	1.4836
3	4	1.5505 ± 0.0749	1.6384
4	1	**1.7022**	**1.7022**

To identify the role of each individual expert, Table 8 reports the market most affected by each omission. Every expert is critical to a different market: removing slow sparse primarily damages DJIA, removing defensive dense primarily damages HSI, removing fast sparse primarily damages CSI, and removing adaptive dense primarily damages CRYPTO. The complete-score reductions range from 19.8% to 57.9%, and every omission reduces all four metrics in its most affected market.

Table 8: Leave-one-expert-out analysis. For each omitted expert, the table reports the market with the largest performance reduction. All values are percentage changes relative to the complete $K = 4$ bank.

Omitted expert	Critical market	ΔCW	ΔAPY	ΔASR	ΔCR	ΔScore
Slow sparse	DJIA	−9.12%	−31.01%	−18.09%	−26.49%	−19.80%
Defensive dense	HSI	−8.72%	−23.85%	−40.31%	−49.49%	−42.43%
Fast sparse	CSI	−17.22%	−62.65%	−54.82%	−68.80%	−57.92%
Adaptive dense	CRYPTO	−31.02%	−30.52%	−22.73%	−38.73%	−26.96%

These results support the four-expert design. First, $K = 4$ is the smallest bank that covers both dimensions of the 2×2 design. Second, removing any single expert causes a large market-specific degradation that is not apparent from an aggregate average. The experiment therefore supports $K = 4$ as providing sufficient coverage within the proposed expert family. It does not establish a global optimum over arbitrary expert designs or exclude potential benefits from adding a genuinely distinct fifth expert.

C.2 Sensitivity Analysis

We examine whether the main design choices are sensitive to the target supervision, Transformer configuration, and router ensemble size.

C.2.1 Target supervision and Transformer configuration

Fig. 4 compares label horizon, soft-target weight, Transformer depth, and hidden dimension under a matched validation protocol. Each candidate configuration is evaluated across five random seeds under identical training protocols. Importantly, the reported complete score evaluates the final calibrated five-router portfolio system rather than the classification accuracy of an isolated Transformer.

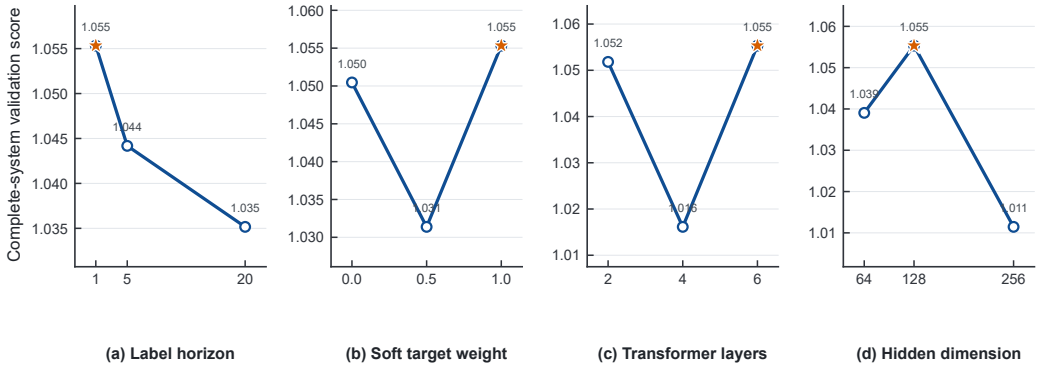

Figure 4: Hyperparameter sensitivity analysis under the validation protocol. Each point reports the validation portfolio score of the complete five-router system averaged across five random seeds. The four panels evaluate label horizon, soft-target weight, Transformer depth, and hidden dimension. Stars indicate the optimal settings selected for final deployment.

The numerical values corresponding to Fig. 4 are reported in Table 9. For the soft-target experiment, we interpolate between the hard and soft targets as:

$$\widetilde{\mathbf{q}}_{t,h} = (1-\beta)\,\mathrm{onehot}\left(\arg\max_k S_{t,h}^{(k)}\right) + \beta \mathbf{q}_{t,h}. \tag{29}$$

Thus, $\beta = 0$ gives hard supervision, whereas $\beta = 1$ corresponds to the target-distribution supervision used by the final model.

The one-day evaluation horizon achieves the best complete-system validation score of 1.0553, compared with 1.0442 and 1.0352 for five- and twenty-day horizons. The fully soft target also achieves

Table 9: Numerical results of the sensitivity analysis. Complete score is the validation score of the calibrated five-router system, while single score is the mean score of the independently deployed routers.

Parameter	Value	Complete	Single	Label acc.	Mean share
	1	**1.0553**	0.6839	0.2805	0.1000
Label horizon	5	1.0442	0.6382	0.3282	0.1500
	20	1.0352	0.4579	0.3162	0.1500
	0.0	1.0505	0.4396	0.2835	0.1000
Soft-target weight	0.5	1.0314	0.5098	0.2922	0.1500
	1.0	**1.0553**	0.6839	0.2805	0.1000
	2	1.0518	0.2539	0.2905	0.1000
Transformer layers	4	1.0161	0.4540	0.2822	0.0750
	6	**1.0553**	0.6839	0.2805	0.1000
	64	1.0391	0.5628	0.2922	0.0513
Hidden dimension	128	**1.0553**	0.6839	0.2805	0.1000
	256	1.0115	**0.7630**	0.2762	0.0000

1.0553, compared with 1.0505 under hard supervision and 1.0314 at an intermediate soft-target weight. Among the tested architectural settings, six Transformer layers and a hidden dimension of 128 provide the best complete-system scores.

The hidden-dimension experiment also illustrates why AlphaRouter performs model selection at the portfolio-system level rather than according to isolated router performance. Increasing the hidden dimension from 128 to 256 raises the mean single-router score from 0.6839 to 0.7630, but the complete five-router score decreases from 1.0553 to 1.0115 and validation selects a zero adaptive share. A stronger isolated router therefore does not necessarily produce a better deployed portfolio after confidence calibration, stable-reference fallback, and multi-router aggregation.

C.2.2 Router Ensemble Size

The preceding experiments fix the ensemble at five routers. We further examine whether the ensemble size M itself affects portfolio performance by evaluating $M \in \{1, 3, 5, 7\}$ under the same expert bank, data split, and proxy-training protocol. For $M = 1$, we average the results of five independently calibrated single-router systems. For $M = 3$, we average all $\binom{5}{3} = 10$ subsets of the original five routers. The deployed system uses $M = 5$. For $M = 7$, two additional routers are trained with new seeds under otherwise identical settings. Each ensemble candidate selects its confidence margin and adaptive share on validation data only; the resulting policy is frozen before test evaluation.

Fig. 5 compares the four ensemble sizes. Increasing M from one to three improves every mean metric across the four markets, confirming that multi-router aggregation provides consistent benefits over a single router. The five-router ensemble matches or exceeds the three-router mean in 15 of the 16 market–metric comparisons, with the only exception being CSI CR (1.064 versus 1.071). Increasing M from five to seven produces no consistent gain: the two settings are identical on DJIA and HSI; seven routers slightly improve CSI (0.11%–0.37%) but reduce CRYPTO (0.62%–0.91%), with a mean absolute relative difference of only 0.26% across all 16 comparisons. Since $M = 7$ requires 40% more training and inference cost than $M = 5$, five routers provide the best performance–cost trade-off within the evaluated range.

C.3 Standalone Performance of the Shared Experts

The main results in Table 2 report the performance of the combined system, but they do not by themselves indicate how much of the observed margin originates from the expert bank versus the learned routing mechanism. This subsection therefore evaluates each expert as a standalone strategy under exactly the same protocol as the main experiments.

Each expert is executed on its own, using the same asset universes, capped-simplex constraint, portfolio-drift convention, proportional transaction cost of 0.25% per unit turnover, and chronological test periods as all other methods. No routing, confidence calibration, or multi-router aggregation is applied. Table 10 reports the results, with the complete AlphaRouter system included for reference.

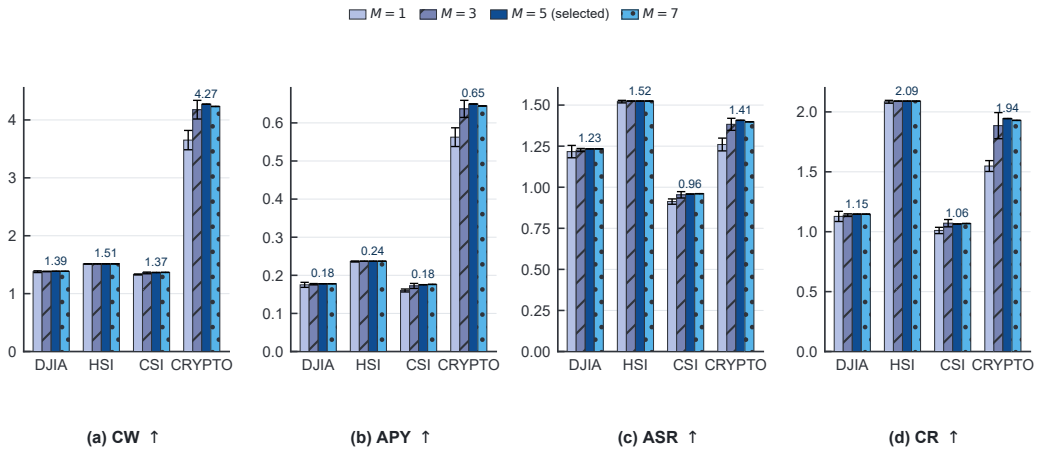

Figure 5: Sensitivity to router ensemble size. Bars report CW, APY, ASR, and CR for $M \in \{1, 3, 5, 7\}$. The M=1 bars average five independently calibrated routers; the M=3 bars average all ten three-router subsets. Error bars denote one standard deviation. Dark blue identifies the selected five-router setting. Higher values are preferred.

Table 10: Standalone test performance of the four shared portfolio experts, evaluated under the same protocol as the main experiments. The complete AlphaRouter system is repeated from Table 2 for reference. Bold denotes the best value in each market. The expert marked with ⋆ is the training-selected long-term reference expert for that market, and its values coincide with the "No neural routing" variant in Table 13. Higher values are better.

Market	Strategy	CW	APY	ASR	CR
DJIA	Slow sparse⋆	1.3746	0.1731	1.2032	1.1058
	Defensive dense	1.3010	0.1412	1.1243	0.9704
	Fast sparse	1.2418	0.1149	0.8971	0.7642
	Adaptive dense	1.2872	0.1351	1.0417	0.8853
	AlphaRouter	**1.3858**	**0.1780**	**1.2337**	**1.1467**
HSI	Slow sparse	1.3874	0.1826	0.9648	1.0431
	Defensive dense⋆	1.4932	0.2279	1.4515	1.9325
	Fast sparse	1.3294	0.1570	0.8524	0.8927
	Adaptive dense	1.4512	0.2101	1.3082	1.6752
	AlphaRouter	**1.5145**	**0.2369**	**1.5245**	**2.0897**
CSI	Slow sparse	1.2461	0.1211	0.6873	0.6541
	Defensive dense	1.3072	0.1493	0.8759	0.9218
	Fast sparse⋆	1.3543	0.1707	0.9244	1.0097
	Adaptive dense	1.2913	0.1420	0.8034	0.8261
	AlphaRouter	**1.3650**	**0.1755**	**0.9584**	**1.0645**
CRYPTO	Slow sparse	2.6392	0.3973	0.9634	1.0217
	Defensive dense	2.4183	0.3558	0.9841	1.0853
	Fast sparse	3.4218	0.5282	1.1836	1.4521
	Adaptive dense⋆	4.1086	0.6275	1.3614	1.8162
	AlphaRouter	**4.2714**	**0.6496**	**1.4070**	**1.9444**

Three observations follow. First, the four experts are genuinely complementary rather than uniformly ordered. The best standalone expert differs across markets: slow sparse on DJIA, defensive dense on HSI, fast sparse on CSI, and adaptive dense on CRYPTO. Moreover, an expert that performs well in one market can perform poorly in another. Fast sparse is the weakest expert on DJIA with a CW of 1.2418, yet it is the strongest on CSI with a CW of 1.3543. This heterogeneity is the property that makes expert-level routing meaningful; if one expert dominated everywhere, a fixed choice would suffice.

Second, the source of the margin over the external baselines is market-dependent. On CSI and CRYPTO, the best standalone expert already exceeds the strongest external baseline in cumulative wealth: fast sparse reaches 1.3543 against 1.283 for AlphaGAT, and adaptive dense reaches 4.1086

against 3.408 for MeanVar. A substantial part of the reported advantage in these two markets therefore originates from the constrained expert construction rather than from learned routing. On DJIA and HSI the situation differs, as no single expert reaches the strongest external cumulative wealth of 1.376 and 1.501 respectively, and the advantage in these markets is obtained only after routing. We note that the picture is not identical for every metric; on HSI, for example, defensive dense alone already attains an ASR of 1.4515, exceeding the strongest external ASR of 1.253. We report this decomposition explicitly so that the contribution of each component can be assessed directly.

Third, and most relevant to the proposed framework, AlphaRouter improves upon every individual expert in every market and on every metric. This holds in particular against the training-selected reference expert, which is the strongest single candidate identified before deployment. The improvement is smallest on DJIA, where CW increases from 1.3746 to 1.3858, and largest on CRYPTO, where CW increases from 4.1086 to 4.2714. Since the routed portfolio is a convex combination of the reference and routed experts, this improvement cannot be obtained by any fixed choice among the candidates. The uniform residual control in Table 13 further shows that a fixed combination is also insufficient. As shown in Appendix C.4, it is furthermore achieved with only one to three expert transitions per test period, which is consistent with the intended role of routing as a controlled adjustment to a stable reference allocation rather than as a replacement for it.

C.4 Validation Calibration and Routing Behavior

The preceding ablation results show that neural routing provides an incremental improvement over the long-term reference in all four markets. We next examine whether these improvements require frequent expert switching. A routing decision affects the executed allocation whenever the validation-selected adaptive share is positive and the majority-voted routed expert differs from the reference expert. Table 11 summarizes the frozen deployment parameters and the resulting test-period routing behavior.

Table 11: Validation calibration and test-period routing behavior. Confidence margin and adaptive share are selected using validation data and frozen before testing. Routed experts lists the experts selected by the calibrated majority vote during testing, while switches denotes the total number of changes in the majority-voted expert.

Market	Conf. margin	Adapt. share	Reference expert	Routed experts	Switches
DJIA	0.10	0.10	Slow sparse	Slow / defensive	2
HSI	0.10	0.10	Defensive dense	Defensive / adaptive	2
CSI	0.10	0.20	Fast sparse	Fast $\rightarrow$ defensive	1
CRYPTO	0.20	0.20	Adaptive dense	Adaptive / fast	3

Validation activates adaptive routing in all four markets, with relatively small adaptive shares. DJIA and HSI use $\alpha = 0.10$, whereas CSI and CRYPTO use $\alpha = 0.20$. Thus, between 80% and 90% of the final expert-level allocation remains anchored to the long-term reference even when the routed expert differs from it. This allocation structure is consistent with the intended role of routing as a controlled adjustment rather than a complete replacement of the stable portfolio.

The calibrated routing decisions also change infrequently. Only two majority-vote transitions occur on DJIA and HSI, one on CSI, and three on CRYPTO over their complete test periods. These switch counts are small relative to the 502, 492, 485, and 731 test dates, respectively. Therefore, the performance improvements reported in the ablation study are obtained without frequent expert switching.

The market-specific routing patterns are also economically interpretable at the expert level. DJIA alternates between the slow sparse reference and the defensive dense expert, HSI between defensive dense and adaptive dense, and CRYPTO between adaptive dense and fast sparse. CSI exhibits a single transition from the fast sparse reference to defensive dense. Although the specific expert pair differs across markets, the common behavior is that the reference expert remains dominant and the router activates only a limited alternative allocation.

Taken together, these results show that AlphaRouter achieves adaptive portfolio adjustment while retaining low routing frequency. Combined with the no-neural-routing ablation, the routing audit indicates that the gains from adaptive routing do not require continuous strategy switching.

C.5 Transaction-Cost Stress Test

In quantitative portfolio management, excessive strategy switching often incurs severe turnover friction that can erode real-world profitability. To assess whether AlphaRouter's performance relies on low-friction assumptions or genuinely robust allocations, we conduct a transaction-cost stress test. Specifically, we hold all generated portfolio decisions strictly fixed and vary the proportional transaction cost from 0 to 100 basis points (bps) per unit turnover (where 25 bps is the baseline used in the main paper, and 100 bps represents an extreme friction scenario). No models, confidence thresholds, or parameters are retrained or readjusted.

Fig. 6 shows how cumulative wealth changes as transaction cost increases. The star marks the 25-basis-point cost used in the main experiments. Table 12 provides the corresponding numerical values. As expected, cumulative wealth decreases monotonically with increasing transaction cost. At the main 25-basis-point setting, CW is 1.386, 1.514, 1.365, and 4.271 on DJIA, HSI, CSI, and CRYPTO, respectively. Even at 100 basis points, the corresponding values remain 1.268, 1.445, 1.096, and 3.480. Among the evaluated markets, CSI exhibits the largest proportional sensitivity to trading friction, whereas HSI is the least affected. These results also motivate the use of a stable reference and calibrated routing rather than unconditionally following short-term router fluctuations.

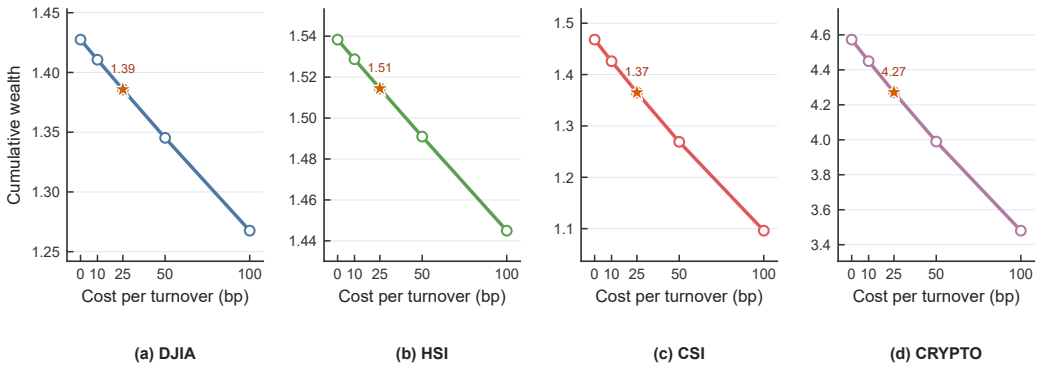

Figure 6: Transaction-cost stress test with all AlphaRouter portfolio decisions and deployment parameters held fixed. The star marks the 25-basis-point transaction cost used in the main experiments.

Table 12: Cumulative wealth under different proportional transaction costs. Portfolio decisions, router parameters, the reference expert, confidence margin, and adaptive share are fixed across all cost settings.

Cost (bp)	DJIA	HSI	CSI	CRYPTO
0	1.4275	1.5383	1.4680	4.5729
10	1.4107	1.5287	1.4260	4.4498
25	**1.3858**	**1.5145**	**1.3650**	**4.2714**
50	1.3453	1.4910	1.2690	3.9897
100	1.2676	1.4450	1.0961	3.4804

C.6 Complete Ablation Results

Table 13 reports the exact numerical values underlying Fig. 3. All variants use the same four-expert bank, asset universes, investment constraints, transaction costs, test dates, and portfolio simulator. They differ only in how the expert-level routing decision is formed or how it is combined with the historically best-performing reference expert.

Table 13: Exact ablation results under the common test protocol. Full denotes AlphaRouter with five independently trained routers, confidence calibration, hard majority voting, the historically selected reference expert, and the adaptive final gate.

Market	Variant	CW	APY	ASR	CR
DJIA	Full	**1.3858**	**0.1780**	**1.2337**	**1.1467**
	Single router mean	1.3762	0.1738	1.2108	1.1184
	Soft consensus	1.3805	0.1757	1.2216	1.1298
	No confidence calibration	1.3774	0.1743	1.2149	1.1162
	No stability fallback	1.3254	0.1519	1.0857	0.9788
	No neural routing	1.3746	0.1731	1.2032	1.1058
	Uniform residual control	1.3718	0.1719	1.1964	1.0971
HSI	Full	**1.5145**	**0.2369**	**1.5245**	**2.0897**
	Single router mean	1.4987	0.2303	1.4682	1.9514
	Soft consensus	1.5041	0.2326	1.4876	1.9873
	No confidence calibration	1.4998	0.2308	1.4721	1.9627
	No stability fallback	1.3927	0.1849	1.1486	1.4946
	No neural routing	1.4932	0.2279	1.4515	1.9325
	Uniform residual control	1.4874	0.2254	1.4378	1.9016
CSI	Full	**1.3650**	**0.1755**	**0.9584**	**1.0645**
	Single router mean	1.3311	0.1602	0.9127	1.0104
	Soft consensus	1.3257	0.1578	0.8814	0.9303
	No confidence calibration	1.3368	0.1628	0.9011	0.9802
	No stability fallback	1.2016	0.1001	0.7126	0.6039
	No neural routing	1.3543	0.1707	0.9244	1.0097
	Uniform residual control	1.3258	0.1578	0.8811	0.9293
CRYPTO	Full	**4.2714**	**0.6496**	**1.4070**	**1.9444**
	Single router mean	3.7926	0.5834	1.2879	1.6128
	Soft consensus	4.0627	0.6211	1.3552	1.7916
	No confidence calibration	4.0185	0.6150	1.3416	1.7587
	No stability fallback	1.5956	0.1748	0.5725	0.3815
	No neural routing	4.1086	0.6275	1.3614	1.8162
	Uniform residual control	3.9473	0.6053	1.3268	1.7049

The long-term reference and neural routing play complementary roles. Removing the stability fallback substantially reduces both portfolio growth and risk-adjusted performance in every market. CW decreases from 1.3858 to 1.3254 on DJIA, from 1.5145 to 1.3927 on HSI, from 1.3650 to 1.2016 on CSI, and from 4.2714 to 1.5956 on CRYPTO. The corresponding CR values decrease from 1.1467 to 0.9788, from 2.0897 to 1.4946, from 1.0645 to 0.6039, and from 1.9444 to 0.3815. These reductions confirm that the reference mechanism is important for preventing noisy short-term predictions from dominating the executed portfolio.

Neural routing nevertheless provides a consistent incremental contribution. Compared with the no-neural-routing variant, the full method improves CW from 1.3746 to 1.3858 on DJIA, from 1.4932 to 1.5145 on HSI, from 1.3543 to 1.3650 on CSI, and from 4.1086 to 4.2714 on CRYPTO. The improvements are also consistent for APY, ASR, and CR, with relative CW improvements of approximately 0.81%, 1.43%, 0.79%, and 3.96%, respectively. This result shows that the learned routing component improves the stable reference portfolio across all four evaluated markets rather than being beneficial in only one market.

The comparison with the single-router mean demonstrates the contribution of multi-router aggregation. The complete system outperforms the average independently deployed router in all four markets. In particular, on CRYPTO, CW increases from 3.7926 to 4.2714 and CR increases from 1.6128 to 1.9444. This result supports the use of multiple independently initialized routers rather than relying on a single neural optimization trajectory.

Hard majority voting also consistently outperforms soft probability aggregation. Replacing majority voting with soft consensus reduces CW from 1.3858 to 1.3805 on DJIA, from 1.5145 to 1.5041 on HSI, from 1.3650 to 1.3257 on CSI, and from 4.2714 to 4.0627 on CRYPTO. Similarly, removing confidence calibration reduces performance in all four markets. These results indicate that both discrete multi-router agreement and confidence-based filtering contribute to robust expert selection.

Finally, the uniform residual control is consistently weaker than learned routing. This comparison is particularly important because both variants can allocate a nonzero share away from the long-term

reference expert. The superior performance of AlphaRouter therefore cannot be explained solely by mixing the reference expert with other candidate portfolios; the learned market-dependent routing decision contributes to selecting more appropriate adjustments.

Overall, the results support the intended stability–adaptation decomposition of AlphaRouter. The long-term reference remains the dominant allocation component, while neural routing provides a smaller but consistent market-dependent correction. As shown in Appendix C.4, this improvement is obtained with only a small number of expert transitions during the test periods.

C.7 Run Consistency

AlphaRouter combines independently initialized routers to reduce dependence on a single neural optimization trajectory. Table 14 reports the performance of the five independently deployed routers before multi-router aggregation. Their mean performance corresponds to the single-router mean in Table 13. No individual router is selected according to test performance.

The complete system outperforms the mean independently deployed router across all four markets. The difference is most pronounced on CRYPTO, where the single-router mean achieves CW, APY, ASR, and CR values of 3.7926, 0.5834, 1.2879, and 1.6128, compared with 4.2714, 0.6496, 1.4070, and 1.9444 for the complete system. These results motivate the use of multiple independently trained routers and calibrated majority voting instead of relying on a single neural initialization. No individual router is retrospectively selected according to test performance.

Table 14: Performance of five independently initialized router deployments before multi-router aggregation. The final AlphaRouter system combines all five routers through validation-calibrated majority voting.

Market	Seed	CW	APY	ASR	CR
DJIA	101	1.3717	0.1719	1.2024	1.1072
	202	1.3958	0.1822	1.2391	1.1498
	303	1.3536	0.1641	1.1776	1.0825
	404	1.3596	0.1667	1.1868	1.0936
	505	1.4003	0.1841	1.2481	1.1589
HSI	101	1.4970	0.2296	1.4640	1.9440
	202	1.5192	0.2389	1.5152	2.0314
	303	1.5039	0.2325	1.4821	1.9752
	404	1.4745	0.2201	1.4028	1.8217
	505	1.4989	0.2304	1.4769	1.9847
CSI	101	1.3507	0.1691	0.9440	1.0442
	202	1.3352	0.1621	0.9225	1.0181
	303	1.3121	0.1516	0.8831	0.9750
	404	1.3542	0.1706	0.9502	1.0513
	505	1.3033	0.1476	0.8637	0.9634
CRYPTO	101	3.7898	0.5830	1.2841	1.6040
	202	3.8065	0.5854	1.3015	1.6382
	303	3.7996	0.5844	1.2952	1.6251
	404	3.7870	0.5826	1.2798	1.5996
	505	3.7801	0.5816	1.2789	1.5971

C.8 CSI Routing Case Study

Although adaptive routing contributes to all four evaluated markets, CSI provides the clearest example of an infrequent routing adjustment because only one majority-vote transition occurs over its 485 test dates. Fig. 7 examines this transition in greater detail. The upper panel compares the cumulative wealth of the full AlphaRouter system with the stable reference strategy obtained by disabling neural routing and with the mean performance of five independently deployed routers. The lower panel shows the effective non-reference share used by the final allocation. The vertical line marks the calibrated routing transition from fast sparse to defensive dense.

The routing adjustment is highly selective: only one transition occurs over the 485 CSI test dates. Nevertheless, AlphaRouter reaches a final CW of 1.365, compared with 1.354 when neural routing is disabled and 1.331 for the single-router mean. The corresponding CR increases from 1.010 for

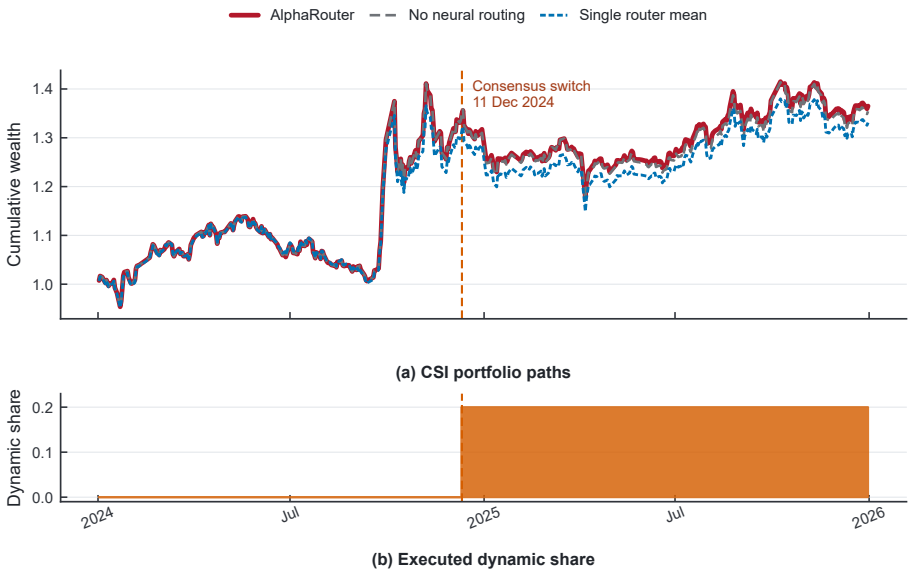

Figure 7: CSI routing case study. The upper panel compares AlphaRouter with the reference-only strategy and the mean of five independently deployed routers. The lower panel shows the effective non-reference share. The vertical line marks the calibrated transition from fast sparse to defensive dense.

the reference-only strategy to 1.064 for AlphaRouter. This case illustrates an important property of AlphaRouter. Adaptive routing does not need to switch experts frequently to improve the final portfolio. Instead, the stable reference remains dominant for most of the test period, while a limited adjustment is introduced when calibrated multi-router evidence supports a different feasible portfolio. Similar low-frequency routing behavior is observed on the other markets, as summarized in Table 11.