PORTFOLIOROUTER: CONSEQUENCE-AWARE MULTI-TASK ROUTING FOR PORTFOLIO AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Portfolio agents can recommend materially different allocations for the same trading date, but standard best-expert classification does not weight errors by their financial cost. The cost of a routing error depends on the return forgone by choosing one proposal over another. We introduce PortfolioRouter, a consequence-aware portfolio router that learns when agent choice matters and evaluates the value of switching between proposals. Its multi-task routing module uses market-state representations and current portfolio proposals to jointly learn a general selector, a critical-date detector, a specialist trained for critical-date selection, and an auxiliary prediction of volatility-normalized relative asset returns. A default-relative value model then uses realized historical feedback to estimate whether changing the default is worthwhile. Its objective penalizes harmful replacements, and estimates across history lengths are combined conservatively. Selective deployment uses a gate calibrated on the preceding quarter to decide whether to add a switch. In U.S. and China backtests covering 2025 through June 2026, PortfolioRouter achieves mean cumulative returns of 34.86% and 45.92% and mean Sharpe ratios of 1.51 and 1.87 respectively, outperforming 17 baselines in both markets. Code and artifacts are avaliable in `https://anonymous.4open.science/r/PortfolioRouter`

1 INTRODUCTION

Investment decisions are increasingly supported by portfolio agents that translate market observations and forecasts into asset allocations. These agents use different signals, modeling assumptions, and decision rules, and may therefore recommend materially different allocations for the same trading date. Recent financial systems combine multiple experts rather than commit to a single policy (Liu & Lo, 2025; Zhou et al., 2026). When portfolio agents produce complete candidate allocations, the system must choose which one to execute on each date. This routing decision is financially consequential because the selected allocation's return compounds into portfolio wealth (Cover, 1991; Helmbold et al., 1998).

Common routing objectives do not fully reflect this financial consequence. Mixture-of-experts, cascade, and LLM-routing methods commonly optimize prediction quality or trade it against inference cost (Jacobs et al., 1991; Jitkrittum et al., 2023; Ding et al., 2024; Ong et al., 2025). In portfolio routing, treating the task as best-expert classification is likewise incomplete: every incorrect choice is penalized equally, even though some mistakes forgo far more return than others. As presented in Figure 1, two policies are equally accurate across six dates, but correct choices on three one-basis-point dates sum to 3 basis points, while correct choices on dates worth 8, 6, and 4 basis points sum to 18. The example shows that routing quality depends on where errors occur, not only on how often they occur. In this illustration, the best agent earns the displayed return and every other agent earns zero, so the values also equal the top-two return gaps. We add one-period basis-point outcomes here for clarity; the reported backtests compound daily returns.

We therefore formulate *consequence-aware portfolio routing* around two questions: when does agent choice matter, and which alternative, if any, improves the allocation that would otherwise be selected? The realized top-two expert-return gap retrospectively identifies critical dates on which choosing the right proposal matters most. Switching value instead measures the signed incremental value of a specific alternative relative to a reference allocation. A large top-two gap can occur when

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.

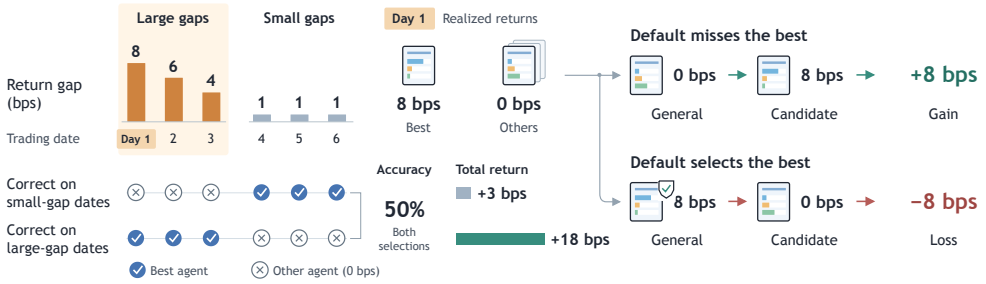


Figure 1: Equal routing accuracy can yield different portfolio value.

the current allocation is already best, so criticality alone does not justify a switch. Both quantities depend on next-period returns and therefore serve as supervision, not inputs at decision time. At deployment, the router must make both judgments from current market history, current proposals, and previously matured outcomes.

Realizing this formulation raises three learning challenges. First, critical dates form a small subset of the data. A single selector receives most of its supervision from ordinary dates, whereas a specialist trained only on critical dates has limited data. Second, a critical-date label says that the choice matters, but not which choice to make. The current choice may already be best, and the alternatives can differ in both benefit and harm. The router must therefore estimate the signed value of each possible change. Third, deployment must decide how selective to be. Because current-period outcomes are unavailable, this decision must be calibrated solely from previously matured feedback.

PortfolioRouter addresses these challenges through multi-task routing, default-relative value estimation, and selective deployment. First, a multi-scale encoder represents the current market state from price and volume history. Together with current portfolio proposals and their disagreement, this representation supports four jointly trained tasks: general selection, critical-date detection, critical-date selection, and auxiliary relative-return prediction. A specialist handles critical-date selection, sharing a market representation with the general selector while retaining a distinct objective. Second, default-relative value models estimate each candidate's signed incremental value relative to the general selector's choice, denoted General. Estimates from long, medium, and recent histories are combined conservatively to discount alternatives whose predicted advantages are inconsistent across histories. Third, selective deployment preserves the base decision produced by the multi-task router and otherwise switches from General only when positive value passes a gate calibrated on the preceding quarter. Figure 2 follows the same structure: panel (a) prepares the shared inputs, and panels (b)–(d) implement the three routing modules.

Our contributions are:

- We formulate consequence-aware portfolio routing, distinguishing date-level criticality from the candidate-specific value of an alternative while restricting deployment to information available at decision time (Sections 3 and 4.2).
- We develop PortfolioRouter, which combines multi-task routing with default-relative value estimation and validation-calibrated selective deployment (Section 4).
- In zero-cost U.S. and China walk-forward backtests, PortfolioRouter improves mean return over General by 8.69 and 13.59 percentage points, respectively. Paired attribution shows positive critical-date contributions in both markets, with critical dates accounting for most of the gain in China (Section 5).

2 Related Work

Selective prediction and deferral. Selective prediction augments a predictor with a reject option and optimizes risk at a prescribed coverage (Geifman & El-Yaniv, 2017; 2019). Learning to defer instead allocates decisions between a model and a downstream expert according to their instance-dependent losses (Madras et al., 2018; Mozannar & Sontag, 2020). PortfolioRouter conditionally

selects among available portfolio proposals. Its gate either retains the general selector's choice or replaces it with an expert supported by the specialist or default-relative value model.

Model and expert routing. Learned gates are central to mixtures of experts (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022), while confidence-based cascades, Hybrid LLM, and RouteLLM route queries between models with different quality and cost profiles (Jitkrittum et al., 2023; Ding et al., 2024; Ong et al., 2025). Because every candidate proposal is produced before routing, PortfolioRouter changes portfolio choice rather than proposal-generation cost. Its gate evaluates the economic value of replacing a default allocation.

Expert routing for financial decisions. Financial routing methods differ in what their experts produce and where selection occurs. LLMoE uses an LLM to interpret prices and news, routes to optimistic or pessimistic prediction experts, and converts the selected forecast into an all-in/all-out trading decision (Liu & Lo, 2025). TradingMoE routes tokens to residual experts inside an LLM and evaluates whether sampled inactive experts should replace an active expert during training (Zhou et al., 2026). PortfolioRouter operates on a pool of completed portfolio proposals. Its supervision separates large-gap dates from the benefit and harm of a particular replacement, and its deployment rule retains a default proposal unless a switching branch accepts an alternative with a predicted economic advantage.

Decision-focused portfolio learning. Decision-focused learning evaluates predictions through downstream decisions (Donti et al., 2017; Wilder et al., 2019; Elmachtoub & Grigas, 2022). In portfolio selection, allocations are ultimately judged by their wealth paths rather than classification accuracy alone (Cover, 1991; Helmbold et al., 1998). Accordingly, we define a retrospective critical-date label from the realized top-two expert-return gap and learn switching value from the signed log-return difference to the default. The former identifies dates with a distinctly best proposal; the latter evaluates a specific replacement.

3 Problem Formulation

We use *expert* to denote a candidate portfolio agent. On date t, the router observes a market window $\mathbf{X}_t \in \mathbb{R}^{L\times F}$ and E portfolio proposals, each covering A risky assets. With $\mathbf{w}_{t,e} \in \mathbb{R}^A$, the proposal matrix is $\mathbf{P}_t = [\mathbf{w}_{t,1}^\top; \ldots; \mathbf{w}_{t,E}^\top] \in \mathbb{R}^{E\times A}$; unallocated wealth remains in zero-return cash. Both $\mathbf{X}_t$ and $\mathbf{P}_t$ contain observations available through close t. The backtest assumes zero-latency execution at that same close and evaluates the selected weights through close $(t+1)$. For risky-asset return vector $\boldsymbol{\rho}_{t\to t+1}$, expert e realizes

$$r_{t,e} = \mathbf{w}_{t,e}^\top \boldsymbol{\rho}_{t\to t+1}. \tag{1}$$

Once realized, the return in Equation 1 supplies supervision and evaluation labels. Online feedback features and monthly value-model updates use only outcomes strictly preceding the current decision date. Figure 2 summarizes the pipeline in four panels. Panel (a) encodes the market history and agent proposals. The three routing modules then follow: panel (b) forms the General and base choices, panel (c) estimates default-relative value and aggregates evidence across history lengths, and panel (d) makes the final selective-deployment decision. Dashed paths carry matured outcomes to the selectors and value model; the auxiliary return task shown in panel (b) is used only during training.

A routing policy selects $a_t = \pi(\mathbf{X}_t, \mathbf{P}_t, \mathcal{M}_t)$, where $\mathcal{M}_t$ contains matured historical outcomes, and realizes r_{t,a_t}. Its gross terminal wealth over test dates $\mathcal{T}$ is

$$W_{\mathcal{T}} = \prod_{t\in\mathcal{T}} (1 + r_{t,a_t}). \tag{2}$$

Let $\mathcal{B}_t = \arg\max_e r_{t,e}$ denote the set of realized best experts. The general selector treats every member of $\mathcal{B}_t$ as a winner. The specialist trained for critical-date selection instead uses the deterministic label $e_t^\star = \min \mathcal{B}_t$, breaking numerically equal returns by the fixed expert ordering.

Let $r_{t,(1)} \geq r_{t,(2)}$ be the two largest expert returns. We define the realized top-two gap

$$\Delta_t = r_{t,(1)} - r_{t,(2)}. \tag{3}$$

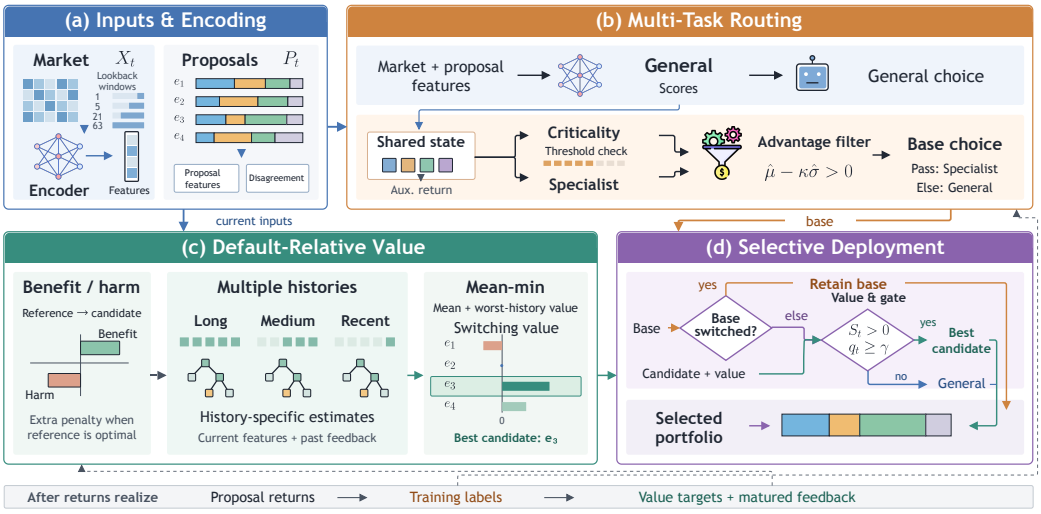


Figure 2: PortfolioRouter separates (a) shared input encoding from three routing modules: (b) multi-task routing, (c) default-relative value, and (d) selective deployment. The three routing modules address the three learning challenges, respectively; dashed paths carry matured feedback.

Within each walk-forward training split, we set $\tau_{0.8}$ to the 80th percentile of the gap in Equation 3 and define $y_t^{\mathrm{D}} = \mathbb{I}\{\Delta_t \geq \tau_{0.8}\}$. This retrospective label defines critical dates by realized expert separation, identifying approximately the highest quintile of training dates; quantile ties can make the fraction slightly larger.

A large gap identifies a clear advantage of the best proposal over the runner-up. When the top two proposals are close, a lower-ranked proposal may still incur a substantial opportunity cost. We therefore use this label to specialize selection, while learning the value of specific replacements on all dates. At inference, neither Δ_t nor $\mathcal{B}_t$ is known; the router must act from current observations, proposals, and matured feedback.

4 Method

Figure 2 separates shared input processing from the three routing modules. Panel (a) encodes the market history and agent proposals. Panel (b) forms General and a base choice through multi-task routing (Section 4.1). Panel (c) estimates candidate-specific value relative to General and aggregates evidence across history lengths (Section 4.2). Panel (d) preserves an accepted base intervention and otherwise switches from General only when the value and calibrated gate support the candidate (Section 4.3).

4.1 Multi-Task Routing

The backbone learns a general selector for all dates and a specialist trained for critical-date selection, labeled General and Specialist in Figure 2. It first maps $\mathbf{X}_t$ to market state $\mathbf{h}_t^{\mathrm{m}}$ and $(\mathbf{h}_t^{\mathrm{m}}, \mathbf{P}_t)$ to general logits $\mathbf{u}_t$. It then constructs shared state $\mathbf{z}_t$ from the market, proposals, general logits, and proposal disagreement. This state feeds the criticality logit g_t, specialist selection logits $\mathbf{c}_t$, and auxiliary relative-return prediction.

Market state encoding. The input $\mathbf{X}_t$ contains price and volume features for each asset.

The reference encoder applies an independent, single-layer Transformer to the most recent $s \in \{1, 5, 21, 63\}$ sessions and mean-pools each output (Vaswani et al., 2017). The sample-dependent

softmax gate in Equation 4 fuses the four scale representations before projection to the market state:

$$\mathbf{v}_{t,s} = T_s(\mathbf{X}_{t,-s:}), \qquad \mathbf{h}_t^{\mathrm{m}} = P_m\left(\sum_s \alpha_{t,s}\mathbf{v}_{t,s}\right), \quad \boldsymbol{\alpha}_t = \operatorname{softmax} G_m([\mathbf{v}_{t,s}]_s). \tag{4}$$

Proposal-aware routing state. Market history alone does not describe the actions under consideration. We therefore retain the full proposal matrix and construct disagreement vector $\mathbf{d}(\mathbf{P}_t)$ from the cross-expert standard deviation and range for each asset, together with every pairwise L_1 distance between proposals.

The general selector embeds the proposals with ψ and produces

$$\mathbf{u}_t = G([\mathbf{h}_t^{\mathrm{m}}; \psi(\mathrm{vec}(\mathbf{P}_t))]). \tag{5}$$

A shared projection ϕ then combines the general-selector logits with the market state, raw proposals, and their disagreement:

$$\mathbf{z}_t = \phi([\mathbf{h}_t^{\mathrm{m}}; \mathrm{vec}(\mathbf{P}_t); \mathbf{u}_t; \mathbf{d}(\mathbf{P}_t)])\,. \tag{6}$$

The shared state in Equation 6 describes both the market conditions and the differences among the proposed allocations.

Four prediction tasks. The shared model trains four related tasks:

1. **General selection** produces the default expert logits $\mathbf{u}_t$ over all dates.
2. **Critical-date detection** maps $\mathbf{z}_t$ to a criticality logit g_t predicting y_t^{D}.
3. **Critical-date selection** uses the specialist to map $\mathbf{z}_t$ to selection logits $\mathbf{c}_t$. It is trained to predict $e_t^\star$ only where $y_t^{\mathrm{D}} = 1$.
4. **Relative-return prediction** predicts volatility-normalized cross-asset relative returns as an auxiliary task.

The first three tasks define the base intervention branch. The fourth head supplies an auxiliary training signal.

Joint training. We train the shared backbone with supervised multi-task learning. The general loss $\mathcal{L}_{\mathrm{G}}$ applies a Bradley–Terry comparison between every realized winner and non-winner, treating tied winners equally (Bradley & Terry, 1952). The detector uses class-balanced binary cross-entropy $\mathcal{L}_{\mathrm{D}}$; the specialist uses cross-entropy $\mathcal{L}_{\mathrm{C}}$ weighted by Δ_t; and the auxiliary head uses Huber loss $\mathcal{L}_{\mathrm{R}}$ on relative returns (Huber, 1964). The total loss is

$$\mathcal{L} = \lambda_{\mathrm{G}}\mathcal{L}_{\mathrm{G}} + \lambda_{\mathrm{D}}\mathcal{L}_{\mathrm{D}} + \lambda_{\mathrm{C}}\mathcal{L}_{\mathrm{C}} + \lambda_{\mathrm{R}}\mathcal{L}_{\mathrm{R}}. \tag{7}$$

We fix $\lambda_{\mathrm{G}} = 1$ and choose the remaining weights and checkpoint by preceding-quarter validation.

Forming the base policy. Let $a_t^{\mathrm{G}} = \arg\max_e u_{t,e}$ denote the General choice from Equation 5, and let $a_t^{\mathrm{C}} = \arg\max_e c_{t,e}$ denote the specialist choice. The base branch proposes the specialist when its detector passes a validation-selected numeric threshold. A monthly forest then retains that proposal only if its predicted simple-return advantage over General, minus a validation-selected multiple of between-tree disagreement, is positive. We denote the resulting choice by b_t; it is General when either check fails. This filter evaluates the specialist's proposal. The value learner below instead considers every candidate relative to General.

4.2 Default-Relative Value

The default-relative value model estimates the switching value of a candidate relative to General at deployment. Training uses historical outcomes of all four proposals to construct targets for each ordered reference–candidate pair. Let $\ell_{t,e} = \log(1 + r_{t,e})$. For reference j and candidate e, define

$$B_t(j,e) = [\ell_{t,e} - \ell_{t,j}]_+, \qquad H_t(j,e) = [\ell_{t,j} - \ell_{t,e}]_+\big(1 + \lambda\mathbb{I}\{j \in \mathcal{B}_t\}\big). \tag{8}$$

Here $[x]_+ = \max(x, 0)$, and $\lambda \geq 0$ increases the training penalty for replacing an already optimal choice. Equation 8 uses every matured date, including ordinary dates. The 16 pairwise targets share the same market outcome and are grouped by date.

The value learner is trained separately from the backbone and uses its own input features rather than the shared state $\mathbf{z}_t$. Its features summarize price history, current portfolio weights, and matured expert feedback.

Tree and neural estimators learn the conditional switching value $V(j, e) = \mathbb{E}[B - H \mid \mathbf{f}_t]$; they also predict the top-two log-return gap as a possible gate score. This value is distinct from the backbone's binary critical-date prediction.

Combining historical estimates. We fit matched estimators on long, medium, and recent histories: all available observations, the last 252, and the last 126, with recency half-lives of 126, 63, and 21 observations. For a common architecture and protection weight, we combine their estimates as

$$S_t(j, e) = (1 - \rho)\frac{1}{3}\sum_{h=1}^{3} \widehat{V}_t^{(h)}(j, e) + \rho \min_h \widehat{V}_t^{(h)}(j, e), \quad \rho \in \{0, 0.5, 1\}. \tag{9}$$

Equation 9 discounts replacements whose estimated value is sensitive to the history window, with ρ controlling the weight on the least favorable estimate. Tree models are refitted monthly using outcomes strictly before the first decision of that month. Neural models use only data before the validation quarter and remain fixed through the next test quarter; their feedback features continue to update as outcomes mature.

4.3 Selective Deployment

The final module combines the base choice with the aggregated switching value and calibrates when an additional switch is justified.

Executing the final choice. The value branch considers all four agents, setting $\widehat{e}_t = \arg\max_e S_t(a_t^{\mathrm{G}}, e)$, with $S_t(j, j) = 0$. Its gate score q_t is either this candidate's value or the mean predicted top-two log-return gap. The final policy is

$$a_t = \begin{cases} b_t, & b_t \neq a_t^{\mathrm{G}}, \\ \widehat{e}_t, & b_t = a_t^{\mathrm{G}}, \quad S_t(a_t^{\mathrm{G}}, \widehat{e}_t) > 0, \quad q_t \geq \gamma, \\ a_t^{\mathrm{G}}, & \text{otherwise.} \end{cases} \tag{10}$$

Equation 10 combines the base and value branches: established base interventions are retained, and default-relative value determines additional replacements. An activation means $a_t \neq a_t^{\mathrm{G}}$, irrespective of which branch produces it.

Calibrating additional switches. The preceding quarter chooses the estimator, protection weight, history aggregation, and gate. Eligibility requires positive and stable incremental gains; otherwise the router retains the base branch. The full calibration rules appear in Appendix B.

5 Experiments

5.1 Setup

Markets and Assets. We evaluate the full available universes of 23 U.S. ETFs and 25 China ETFs over 2025 through June 2026, using 367 and 352 valid decision dates. Appendix A lists the assets.

Baselines. We compare PortfolioRouter performance with passive and rule-based portfolios, fixed-agent policies, supervised and reinforcement-learning routers, and the financial systems. Appendix C.1 illustrates their definitions and implementation details.

Training and Validation. On every decision date, the candidate portfolio allocations come from GPT-5.4 (OpenAI, 2026), Qwen 3.6 Plus (Qwen Team, 2026), DeepSeek V4 Flash (DeepSeek-AI et al., 2026), and Gemini 3.5 Flash (Google, 2026). Six quarterly folds use historical data for

training and the immediately preceding quarter for validation. Every training and validation outcome matures before the next split begins. Validation selects the model settings and gates before each test quarter; during deployment, only the prescribed monthly tree refits and updates from newly matured feedback continue online. More reproducibility settings are in Appendix A.

Evaluation and Metrics. All methods follow the same daily walk-forward protocol over 2025 and 2026 H1; we report each subperiod and their concatenation. Portfolio outcomes use close-t to close-$(t+1)$ returns, same-close execution, and zero fees, slippage, market impact, and taxes. The primary metrics are cumulative return and annualized Sharpe (Sharpe, 1994), computed from the sample standard deviation of daily returns with a factor of $\sqrt{252}$. Maximum drawdown (MDD) (Magdon-Ismail & Atiya, 2004), selection accuracy, regret, and switching statistics provide complementary evidence.

5.2 Overall Portfolio Performance

Table 1 evaluates the complete routing policy against the baselines under the common execution protocol. PortfolioRouter provides the strongest risk-adjusted performance across the two markets. Among the 17 external baselines, it has a higher Sharpe ratio in all 34 market–method comparisons and a higher return in 32 of 34 comparisons (94.1%). Relative to the unweighted mean of the 11 supervised and reinforcement-learning routers, it improves return by 6.75 and 13.68 percentage points and Sharpe by 0.24 and 0.51 in the U.S. and China, respectively. Comparisons with routers such as ERM are end-to-end because their representations and model-selection procedures also differ.

The only return exceptions are FinMem and FinGPT-Forecaster in the U.S.; both have lower Sharpe ratios than PortfolioRouter. The drawdown advantage is less uniform: PortfolioRouter's mean MDD ranks fourth in the U.S. and first in China. Table 3 further shows that the gain over General in China is larger in 2025 than in 2026 H1.

5.3 Switching and Mechanism Evidence

The ablations consistently support the shared learning tasks. Every task removal lowers both return and Sharpe in both markets (Table 2). Averaged across the four removals, return falls by 5.91 percentage points in the U.S. and 8.62 points in China, while Sharpe decreases by 0.24 and 0.31. Simplifying the history ensemble, aggregation rule, or deployment threshold also lowers both metrics in every market-level comparison, with return losses ranging from 2.40 to 9.76 points and Sharpe losses from 0.09 to 0.37. Protection and the treatment of base interventions have smaller, market-dependent effects. The matched-control protocol appears in Appendix D.4.

Paired attribution shows that critical-date switches contribute positively in both markets but play different roles. They account for 46.8% of the net log-wealth gain in the U.S. and 83.2% in China. U.S. gains are therefore distributed across critical and ordinary dates, whereas the gain in China is concentrated primarily on critical dates. A fixed-count timing replay places the observed additional-switch gains above its randomized range in both markets. The full attribution and timing protocol appear in Appendix D.5.1.

Proposal disagreement is the mean pairwise allocation distance (half the ℓ_1 distance, including cash) across the four agents and is observable before outcomes; "high" exceeds its historical 75th percentile, fixed before the preceding validation quarter. The net log-wealth contribution of high-disagreement dates is 1.51 times their frequency share in the U.S. and 3.97 times their frequency share in China. Across the two markets, about 55% of overrides beat General, indicating that the gain depends on the magnitude of favorable and harmful switches rather than on win rate alone. Figures 3 and 4 show the frequency and realized contribution of these switches. Their bars are three-run means, and the replays describe historical contributions under fixed decisions rather than causal effects. Further analysis appears in Appendix D.5.2.

5.4 Additional Experiments

Additional controls test whether the gains come from date-specific routing, the supervision objective, or the common switching gate. PortfolioRouter outperforms all five fixed-agent policies on return and Sharpe in both markets. Averaged across these policies, its return advantage is 9.52 and 16.07

Table 1: Comparative performance on the 2025–2026 H1 period at zero cost. Learned routing methods report mean ± population standard deviation over three runs per market; deterministic strategies (denoted *) report one value.

Category	Method	U.S. Market		China Market	
		Return (↑%)	**Sharpe (↑)**	**Return (↑%)**	**Sharpe (↑)**
Rule-based Strategies	Equal-Weight Portfolio	33.89	1.26	30.20	1.17
	Momentum (20 days)	32.97	1.36	43.95	1.49
	Buy-and-Hold	25.80	0.96	29.68	1.21
Financial Systems	FinMem	38.38	1.34	35.50	1.34
	FinGPT-Forecaster	**44.87**	1.49	33.29	1.16
Fixed Agent	Validation-Selected Agent	21.73	1.10	30.53	1.32
Supervised Routers	Boosted Reward*	25.52	1.22	31.06	1.35
	Ridge Reward*	32.51	1.44	29.16	1.27
	LightGBM*	26.62	1.20	30.57	1.26
	Shallow ERM Router	30.00 ± 5.45	1.33 ± 0.22	31.18 ± 7.13	1.34 ± 0.28
	Transformer Accuracy	26.79 ± 2.78	1.19 ± 0.12	38.35 ± 2.35	1.52 ± 0.09
	Neural Reward	26.59 ± 2.14	1.22 ± 0.08	29.71 ± 3.73	1.30 ± 0.14
	RouteLLM MF	30.59 ± 1.48	1.38 ± 0.06	31.01 ± 2.98	1.34 ± 0.11
	Qlib LSTM	31.12 ± 1.08	1.35 ± 0.04	29.80 ± 6.70	1.26 ± 0.26
RL Routers	PPO	28.28 ± 3.04	1.29 ± 0.10	33.76 ± 2.24	1.40 ± 0.13
	A2C	23.75 ± 1.89	1.19 ± 0.09	40.49 ± 6.95	1.64 ± 0.21
	DQN	27.41 ± 4.30	1.22 ± 0.17	29.56 ± 2.26	1.29 ± 0.09
Ours	**PortfolioRouter**	34.86 ± 0.55	**1.51 ± 0.04**	**45.92 ± 9.26**	**1.87 ± 0.31**

Table 2: Functional mechanism ablations on the same runs. Each row evaluates the policy obtained under the stated modification and calibration procedure. Task-removal rows retrain the backbone under a matched training budget.

Category	Method	U.S. Market		China Market	
		Return (↑%)	**Sharpe (↑)**	**Return (↑%)**	**Sharpe (↑)**
Ours	**PortfolioRouter**	34.86 ± 0.55	1.51 ± 0.04	45.92 ± 9.26	1.87 ± 0.31
Matched Ablations	No Protection ($\lambda = 0$)	35.00 ± 0.92	1.52 ± 0.05	44.68 ± 5.34	1.83 ± 0.17
	Long History Only	30.62 ± 1.84	1.38 ± 0.06	42.76 ± 6.83	1.77 ± 0.23
	Mean Aggregation ($\rho = 0$)	31.16 ± 1.43	1.36 ± 0.05	43.52 ± 10.97	1.78 ± 0.36
	Replace Base Interventions	35.13 ± 0.86	1.52 ± 0.05	45.77 ± 8.39	1.86 ± 0.27
Reselected Controls	Mean Value ($\rho = 0$)	33.47 ± 0.89	1.46 ± 0.04	45.61 ± 10.19	1.84 ± 0.33
	Zero Value Threshold	25.10 ± 1.29	1.14 ± 0.05	42.00 ± 5.14	1.75 ± 0.17
Task Removals	w/o General Task	25.73 ± 1.60	1.14 ± 0.05	33.31 ± 2.41	1.45 ± 0.12
	w/o Detection Task	23.65 ± 1.58	1.06 ± 0.06	38.69 ± 7.54	1.56 ± 0.26
	w/o Critical-Selection Task	32.61 ± 2.55	1.41 ± 0.09	36.13 ± 2.45	1.52 ± 0.13
	w/o Return-Prediction Task	33.81 ± 2.17	1.48 ± 0.10	41.09 ± 5.62	1.71 ± 0.20

percentage points and its Sharpe advantage is 0.29 and 0.53 in the U.S. and China, respectively. Its return margins over the best fixed agent are still 4.66 and 9.66 points (Appendix D.3).

PortfolioRouter also has higher return and critical-date contribution than each of the three matched neural objectives in both markets. Compared with their mean, its return margins are 6.72 points in the U.S. and 5.05 in China, while its critical-date contributions are higher by 244.04 and 332.01 log basis points (Appendix D.6.1).

Under a common switching gate, PortfolioRouter has higher return and Sharpe than all three matched predictor controls in both markets. Its mean margins over these controls are 6.23 and 11.02 percentage points of return and 0.19 and 0.40 Sharpe in the U.S. and China, respectively (Appendix D.6.2).

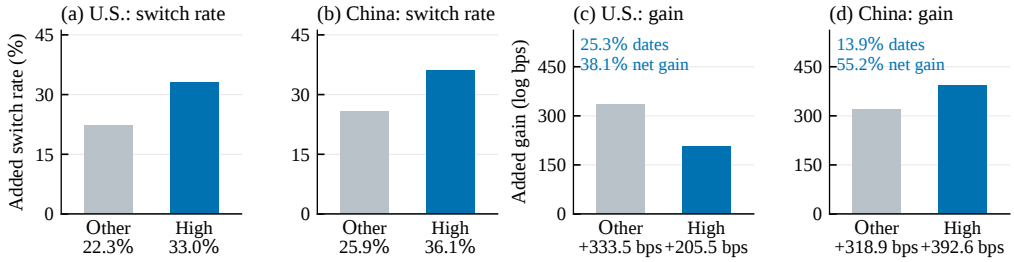


Figure 3: Additional-switch frequency and net log-wealth contribution by proposal disagreement.

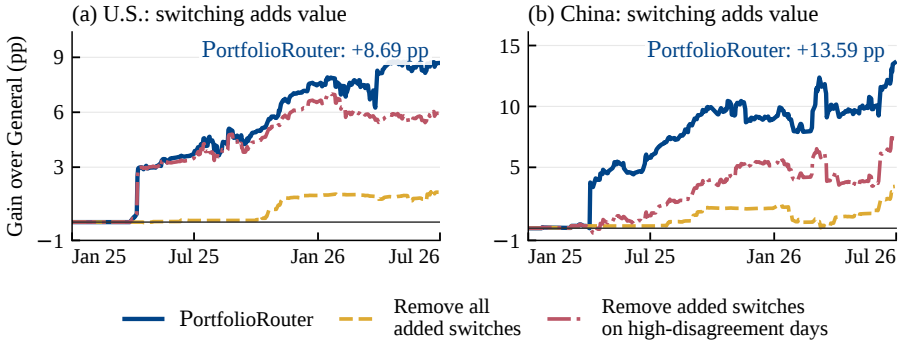


Figure 4: Cumulative-return effects of all added switches and those on high-disagreement dates.

6 Discussion and Limitations

The paired gains depend on the magnitude of favorable and harmful switches: slightly more than half of the overrides beat General. Critical dates contribute positively in both markets, while critical dates explain most of the gain in China. This pattern motivates critical-date specialization alongside value learning across all dates. Figures 3 and 4 link additional switching gains to observable proposal disagreement, with a stronger concentration in China (Appendix D.5.2). The removal replays describe historical contributions under fixed future decisions, rather than causal effects. The evaluation fits a separate router for each market. Performance on independent future periods and transfer without market-specific refitting remain open questions. Same-close execution and zero transaction costs also leave execution frictions unmeasured. Although proposal inputs are filtered by date, the underlying language models may contain knowledge from beyond the simulated decision dates. Future evaluation should combine independent periods, realistic execution, and a broader set of financial systems.

7 Conclusion

PortfolioRouter combines multi-task routing, default-relative value estimation, and selective deployment to account for the economic consequences of agent choice. Its value model learns the benefit and harm of replacing General, aggregates estimates across history lengths, and guides additional switches through a preceding-quarter gate. In the zero-cost, same-close U.S. and China walk-forward evaluations, it improves mean cumulative return and Sharpe over the same runs' General paths. Paired attribution assigns positive contributions to critical-date switches in both markets, with a larger share of the attributed gain in China; these historical replays describe observed paths rather than causal effects.

AI USE STATEMENT

Generative AI tools were used only to identify and summarize literature and improve manuscript readability. They were not used for theory, hypotheses, methodology, experiments, implementation, translation, data processing, analysis, or result interpretation; synthetic data and proofs were not applicable. The authors verified every cited source and all AI-assisted text and take full responsibility. Portfolio agents are experimental components, not authoring tools.

ETHICS STATEMENT

This study uses licensed historical market data, causal news inputs for one baseline, and model-generated portfolio proposals. It involves no human participants, private or sensitive personal data, or real trades. The results are retrospective and do not constitute investment advice. Same-close execution, zero latency, and zero transaction costs may overstate real-world performance; possible post-date knowledge in the underlying language models and the limited markets and evaluation period further restrict generalizability.

REPRODUCIBILITY STATEMENT

Section 5 specifies the walk-forward protocol, temporal splits, execution assumptions, metrics, and random seeds. Appendix A specifies the asset universes, data construction, model-selection space, and consistency checks, while Appendix B gives the objectives and calibration rules. An anonymous implementation accompanies the submission. Upon acceptance, the authors intend to release the implementation and all redistributable experimental materials publicly. The underlying market data cannot be redistributed because of third-party license restrictions; the supplementary material provides the asset identifiers and preprocessing rules needed to reconstruct the experimental panels.

REFERENCES

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.1093/biomet/39.3-4.324.

Thomas M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991. doi: 10.1111/j.1467-9965.1991.tb00002.x.

DeepSeek-AI et al. DeepSeek-V4: Towards highly efficient million-token context intelligence. arXiv:2606.19348, 2026. URL https://arxiv.org/abs/2606.19348.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024.

Priya Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5484–5494, 2017.

Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68 (1):9–26, 2022. doi: 10.1287/mnsc.2020.3922.

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. URL https://jmlr.org/papers/v23/21-0998.html.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4878–4887, 2017.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Google. Gemini 3.5 Flash. Google AI for Developers Documentation, 2026. URL https://ai.google.dev/gemini-api/docs/models/gemini-3.5-flash. Accessed: 2026-09-26.

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. doi: 10.1111/1467-9965.00058.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732.

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. doi: 10.1162/neco.1991.3.1.79.

Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Kuan-Ming Liu and Ming-Chih Lo. LLM-based routing in mixture of experts: A novel framework for trading. arXiv:2501.09636, 2025. URL https://arxiv.org/abs/2501.09636.

David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Malik Magdon-Ismail and Amir F. Atiya. Maximum drawdown. *Risk Magazine*, 17(10):99–102, October 2004. URL https://ssrn.com/abstract=874069.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937. PMLR, 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs with preference data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5503a7c69d48a2f86fc00b3dc09de686-Paper-Conference.pdf.

OpenAI. GPT-5.4 model, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.4. Accessed: 2026-09-26.

Qwen Team. Qwen3.6-Plus: Towards real world agents, 2026. URL https://qwen.ai/blog?id=qwen3.6. Accessed: 2026-09-26.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0.

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

William F. Sharpe. The Sharpe ratio. *The Journal of Portfolio Management*, 21(1):49–58, 1994. doi: 10.3905/jpm.1994.409501.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998. ISBN 9780471030034. URL https://www.wiley-vch.de/en?isbn=9780471030034&option=com_eshop&view=product.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019. doi: 10.1609/aaai.v33i01.33011658.

Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An AI-oriented quantitative investment platform. arXiv:2009.11189, 2020. URL https://arxiv.org/abs/2009.11189.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Jordan W. Suchow, Denghui Zhang, and Khaldoun Khashanah. FinMem: A performance-enhanced LLM trading agent with layered memory and character design. *IEEE Transactions on Big Data*, 11(6):3443–3459, 2025. ISSN 2332-7790. doi: 10.1109/TBDATA.2025.3593370. URL https://doi.ieeecomputersociety.org/10.1109/TBDATA.2025.3593370.

Chang Zhou, Xingtong Yu, Minbin Huang, Zhennan Wu, Yuan Fang, Hong Cheng, and Xinming Zhang. TradingMoE: Routing the right experts in evolving markets. arXiv:2608.11785, 2026. URL https://arxiv.org/abs/2608.11785.

APPENDIX

A REPRODUCIBILITY DETAILS

This appendix specifies the asset universes, model-selection space, and evaluation checks used in the reported experiments. Appendix B provides the corresponding objectives and deployment rules.

Asset universes and evaluation panel. The U.S. universe contains 23 ETFs: AIQ, GLD, IAI, IBB, IGV, IYR, KBE, KIE, MDY, QCLN, QQQ, SMH, SPY, TLT, XHE, XLB, XLC, XLE, XLK, XLP, XLV, XME, and XTL. The China universe contains 25 ETFs: 159819, 159852, 159870, 159915, 159928, 159934, 159945, 159992, 159995, 510310, 510580, 511010, 512010, 512070, 512170, 512200, 512400, 512800, 512880, 512980, 515170, 515220, 515880, 516160, and 588000. Daily OHLCV series were retrieved from Tencent Finance's daily K-line service and span January 2024 through June 2026. We use the provider's forward-adjusted series for China and issuer-verified split adjustments for the U.S. series, then retain the common trading dates across all assets within each market. All methods use the resulting market panels and evaluation dates; routing methods also receive the same agent proposals. The held-out evaluation covers 367 U.S. sessions and 352 sessions in China from January 2025 through June 2026.

Backbone training and selection. The backbone uses four Transformer scales of $\{1, 5, 21, 63\}$ sessions, a hidden dimension of 64, four attention heads, one layer per scale, and a 32-dimensional market embedding. Validation searches over the history window, learning rate, regularization, and task weights. Each quarterly model starts from a fresh initialization. Checkpoint selection begins at epoch 175; training stops after 200 epochs without an improvement of at least 0.005, subject to a 1,600-epoch limit.

Base feedback model. Validation selects among 64-tree random forests with depths in $\{3, 5\}$, minimum leaf sizes in $\{8, 20\}$, and disagreement penalties in $\{0, 0.5, 1\}$. These models use only feedback whose outcomes have matured before the decision date.

Value-model search. The search compares 36 tree configurations and 12 neural configurations, with the neural models evaluated at 80 and 240 update budgets. The candidates form 20 matched families across the three history lengths described in Section 4.3. Tree models use 64 estimators. Neural models use two 128-unit tanh layers, a learning rate of 0.002, and weight decay of 0.10. The direct objective regresses $B - H$; the dual objective combines weighted binary cross-entropy with 0.25 times the magnitude squared error. Both objectives add 0.1 times the top-two-gap squared error after scaling targets on the training data. Validation considers protection weights $\{0, 0.5, 1\}$ and selects the estimator, history aggregation, and switching gate on the immediately preceding quarter. Appendix B gives the selection objective and eligibility conditions.

Reporting and consistency checks. The main results for learned methods aggregate seeds 101, 211, and 307 and report the mean and population standard deviation across the three runs. Within each run, annualized Sharpe is computed from daily returns using the sample standard deviation and a factor of $\sqrt{252}$. Before aggregation, we verify identical evaluation dates within each market, finite daily returns, and strict outcome maturity for every training and validation example. Future-outcome perturbation tests further verify that changing later outcomes does not alter previously constructed features or fitted feedback models. The mechanism, task-removal, objective, and predictor controls reuse the same market panels, walk-forward splits, and reporting conventions.

B METHOD IMPLEMENTATION DETAILS

Input features. For each asset and session, $\mathbf{X}_t$ contains the close-to-close return, open-to-close return, high to low range normalized by the close, cumulative return from the start of the 63-session input window, and normalized log volume.

Auxiliary target. Specifically, for asset a we set $q_{t,a} = \rho_{t\to t+1,a}/\sigma_{t,a}^{(20)}$, where $\sigma_{t,a}^{(20)}$ is trailing 20-session volatility, and predict

$$y_{t,a}^{\mathrm{R}} = \mathrm{clip}\left(q_{t,a} - \frac{1}{A}\sum_{b=1}^{A} q_{t,b}, -10, 10\right). \tag{11}$$

Base selection and deployment schedule. The base network is selected by validation portfolio utility: mean log return minus 0.1 times downside root-mean-square return, plus 0.5 times the mean daily return increment over its general path. The value branch uses the incremental criterion in Section 4.3. Only the base branch operates in 2025 Q1 because value-branch calibration requires a preceding quarter's deployment path. From 2025 Q2 onward, the value branch is calibrated on the preceding quarter's deployment decisions and realized outcomes.

Training schedule. In Equation 7, we fix $\lambda_{\mathrm{G}} = 1$ and select the remaining loss weights on the preceding validation quarter. We train the general task alone for 25 warm-up epochs and linearly introduce the other losses over the next 50 epochs. Checkpoint selection begins at epoch 175 and uses validation portfolio utility.

Value-model features. It receives 196 features in the U.S. and 212 in China: respectively 92/100 price-history summaries, 92/100 current portfolio weights, and 12 feedback features. The feedback comprises each expert's centered-return means over the last 20 and 60 matured observations, plus its 20-observation standard deviation.

Estimator objectives. A tree model directly regresses $V(j,e) = \mathbb{E}[B - H \mid \mathbf{f}_t]$. A neural alternative predicts a benefit share p using B/H-weighted binary cross-entropy and a magnitude m using squared error against $B + H$, and decodes $\widehat{V} = (2p-1)m$. At the population optimum, $p = \mathbb{E}[B]/\mathbb{E}[B+H]$ and $m = \mathbb{E}[B+H]$ conditional on the features, so this decoding recovers the same target when the denominator is positive. Both implementations also predict the top-two log-return gap as a possible gate score. This continuous estimate is separate from the backbone detector's binary critical-date prediction.

Calibration. The previous quarter selects the value estimator, λ, ρ, gate score, and numeric threshold. Let δ_t denote the incremental validation log return over the base decision b_t, let d_t indicate a critical date, and let $\mathcal{B}_t$ be the set of realized best agents. We rank eligible settings by

$$U_{\mathrm{val}} = \frac{1}{N}\sum_{t=1}^{N}\left[\delta_t + 0.5 d_t \delta_t - 0.5\mathbb{I}\{b_t \in \mathcal{B}_t\}[-\delta_t]_+\right].$$

The threshold candidates are zero for value scores and validation quantiles $\{0, .5, .65, .8, .9, .95\}$. Calibration evaluates incremental log return relative to b_t, rewards its contribution on critical dates, and penalizes harming a correct reference choice. Eligible settings require positive total and critical-date gains, at least four additional switches per run on average, positive gains in at least two of the three calibration runs, and nonnegative gains in both chronological halves of the validation quarter. Otherwise, the router retains the base branch. One setting is shared across runs within a market and quarter.

C Additional Experimental Settings

C.1 Baseline Definitions and Implementations

Rule-based investment strategies. daily rebalanced equal weight, 20-day momentum-or-cash, and market ETF buy-and-hold (SPY in the U.S. and 510310 in China).

FinMem. The system uses one original single-symbol agent per ETF, GPT-4-Turbo with temperature 0.3, and text-embedding-ada-002. The agents process causal 2024 observations before the evaluation from 2025 through 2026 H1 and retain their original layered memory, retrieval, reflection, and online feedback. Buy, sell, and hold set, clear, or retain an ETF target; active targets are normalized when their sum is greater than one (Yu et al., 2025).

FinGPT-Forecaster. The system uses the released DOW30 Llama-2-7B LoRA checkpoint in FP16 [1]. Each weekly prediction receives the preceding three weeks of prices and causal news. The author's parser maps positive and negative predictions to active or zero ETF targets, while neutral or unparsed outputs retain the preceding target. Both financial systems use the pre-specified seed 101 run.

Fixed-agent policies. Each of the four agents is used on every date. We also include an agent selected by validation portfolio utility and held fixed during the next quarter. Section D.3 isolates the value of date-level routing.

Common supervised routers. The supervised controls include a Transformer accuracy router (Vaswani et al., 2017), neural and boosted reward routers, a Ridge reward predictor (Hoerl & Kennard, 1970), and RouteLLM MF (Ong et al., 2025). The neural and boosted reward routers use, respectively, a two-hidden-layer multilayer perceptron (Rumelhart et al., 1986) and gradient-boosted regression trees (Friedman, 2001). Both predict each agent's next-session log return, centered across the four proposals, and select the agent with the highest predicted return. We use the estimators and optimizers from the official packages and modify only the ETF data interface, proposal-selection task, and evaluation protocol. Package versions are fixed across runs.

Shallow ERM. Following the empirical risk minimization principle (Vapnik, 1998), we train a 32-unit, one-hidden-layer MLP by cross-entropy. It uses the same 196/212 causal features as the value model, with training-only standardization and up to 400 Adam epochs.

Common portfolio setting. Fixed-agent and router policies are evaluated using the same agent-generated allocations. All policies are long-only, with total risky-asset weight at most one, residual cash, and no separate per-asset allocation cap. Reported wealth compounds daily returns over the stated period.

LightGBM. The model regresses each agent's next-session log return centered across the four agents, scaled by 100. It uses 200 trees, minimum leaf size 20, and L_2 regularization 1. The grid combines 7/15 leaves, learning rates $0.03/0.1$, and all/126-observation histories. Its training is deterministic under these settings, so each configuration produces one distinct path (Ke et al., 2017).

Qlib LSTM. The Qlib implementation (Yang et al., 2020; Hochreiter & Schmidhuber, 1997) receives 63 observations of the five price and volume features for each ETF and predicts its next-session simple return, scaled by 100. A single panel model shares its parameters across the 23 U.S. or 25 China ETFs. Features are standardized with training observations only. The upstream LSTM has two 64-unit layers, no dropout, batch size 32, Adam, at most 200 epochs, and patience 20 on the preceding validation quarter's mean squared error. The learning-rate candidates are $0.0003/0.001$. A proposal's forecast return is the inner product of its allocation with all asset forecasts.

[1] https://huggingface.co/FinGPT/fingpt-forecaster_dow30_llama2-7b_lora

PPO. The policy uses an MLP with two 64-unit layers, four discrete proposal actions, and the same 196/212 causal features (U.S./China) as the value model. Its reward is $100\log(1 + r_{t,a_t})$. Training uses 256-step rollouts, minibatches of 64, ten optimization epochs per rollout, discount 0.99, GAE parameter 0.95, clipping parameter 0.2, and entropy coefficient 0.01. Learning rates are $0.0003/0.001$ and checkpoint budgets are 20,000/60,000 environment steps, rounded up to complete rollouts. Test actions are deterministic. Each training episode replays only the chronological training segment; validation and test outcomes never enter the training environment (Schulman et al., 2017; Raffin et al., 2021).

A2C and DQN routing adaptations. The additional controls use Stable-Baselines3 2.9.0 (Raffin et al., 2021) on the same four discrete proposal actions, 196/212 causal features (U.S./China), and $100\log(1 + r_{t,a_t})$ reward as PPO. Feature scaling is fitted only on the training segment. Episodes contain only chronological training rows whose returns mature before the validation quarter. Validation outcomes mature before the test quarter; the test policy remains fixed and uses deterministic actions.

A2C, a synchronous actor-critic variant (Mnih et al., 2016), uses the default MLP policy and FinRL's released configuration [2]: learning rate 0.0007, five-step rollouts, and entropy coefficient 0.01. The remaining parameters are the upstream defaults, including RMSProp and discount 0.99. DQN (Mnih et al., 2015) uses its default MLP and library configuration: learning rate 0.0001, replay capacity 10^6, batch size 32, training every four steps, one gradient step per update, and target updates every 10,000 steps. Exploration decreases from 1 to 0.05 during the first 10% of the 60,000-step run. Both algorithms use the shared proposal environment.

Each model is trained in one uninterrupted 60,000-step run and evaluated at 20,000- and 60,000-step checkpoints. The first checkpoint precedes the optimizer update for the rollout containing the 20,000-step boundary, whereas the second follows the final training update. Mean preceding-quarter utility across the two calibration runs selects the checkpoint, and this choice is then applied without reselection to the reported seeds. A2C and DQN each use one fixed learning rate, while PPO searches two rates. Training uses one CPU thread per model.

Selection and reporting. The preceding quarter selects configurations by mean validation portfolio utility over two calibration runs. Those choices remain fixed across evaluation seeds. The main tables use seeds 101, 211, and 307 as defined in Appendix C.2. Table 6 reports the resulting risk metrics, and Table 8 reports all three runs. Each quarterly model is evaluated at the prescribed training budgets. Replaying each candidate checkpoint verifies consistency of daily decisions, evaluation dates, and aggregate metrics.

C.2 Reporting Protocol

The main tables report the mean and population standard deviation over seeds 101, 211, and 307 in both markets. All learned baselines use the same seed identifiers. FinMem and FinGPT-Forecaster each use the pre-specified seed 101 run.

The preceding quarter selects value-branch settings using three calibration runs and backbone and baseline configurations using two calibration runs. The selected settings remain fixed across evaluation seeds.

[2] https://github.com/AI4Finance-Foundation/FinRL

D Additional Experimental Details

D.1 Overall Performance, Risk, and Temporal Robustness

Temporal breakdown. Table 1 reports performance over the complete evaluation period from 2025 through 2026 H1. Table 3 decomposes the General and PortfolioRouter paths into the two reported subperiods to assess whether the full-period improvement is confined to one part of the evaluation window.

PortfolioRouter improves on General in both subperiods and markets. The gains are larger in 2025, reaching 7.47 percentage points in the U.S. and 8.93 in China, but remain positive in 2026 H1 at 0.46 and 2.86 points, respectively. The corresponding full-period gains of 8.69 and 13.59 points incorporate compounding across the two subperiods rather than their arithmetic sum.

Complete U.S. results. The main table emphasizes return and Sharpe across markets, whereas Table 4 expands the U.S. comparison with maximum drawdown, the General and base-intervention paths, fixed-agent policies, and the switching-mechanism controls.

PortfolioRouter reaches a 34.86% mean return and a 1.51 Sharpe ratio, improving on General by 8.69 percentage points of return while reducing MDD from -12.89% to -12.06%. Its Sharpe is the highest among the non-ablation methods in the main comparison. No Protection and Replace Base Interventions have slightly higher U.S. return point estimates, but the cross-market comparison below shows that these modifications do not uniformly improve on the complete policy.

Complete China results. Table 5 applies the same expanded comparison to the China market, allowing the risk and ablation patterns in Table 4 to be checked under a different asset universe.

PortfolioRouter obtains a 45.92% mean return and a 1.87 Sharpe ratio, exceeding the rule-based, supervised-routing, and reinforcement-learning comparators shown in the table. Relative to General, it raises return by 13.59 percentage points and improves MDD from -11.01% to -8.67%; the matched switching variants also remain below the complete policy in mean return. Cross-run return variability is higher in China, with a standard deviation of 9.26 percentage points compared with 0.55 in the U.S.

D.2 Official-Code and Financial Baseline Results

Aggregate risk and return. Table 1 gives the broad end-to-end comparison, while Table 6 isolates the official-code routing baselines and financial systems and adds their maximum drawdowns.

Among the official-code routers, Qlib LSTM has the highest U.S. return (31.12%), whereas A2C has the highest return in China (40.49%). PortfolioRouter exceeds these values by 3.74 and 5.43 percentage points and has a higher Sharpe ratio in both markets. FinGPT-Forecaster and FinMem have higher U.S. returns than PortfolioRouter, but their drawdowns are deeper; PortfolioRouter retains the highest U.S. Sharpe and leads the financial systems in return, Sharpe, and MDD in China. FinGPT-Forecaster's original parser accepts 70.3% of its 3,767 forecasts, with the predeclared hold mapping used for the remaining outputs.

Subperiod performance. The aggregate returns above can mask when gains accrue. Table 7 therefore decomposes the same official-code and financial baselines into 2025 and 2026 H1, complementing the decomposition of General and PortfolioRouter in Table 3.

FinGPT-Forecaster's U.S. return remains strong in both subperiods, at 26.14% in 2025 and 14.85% in 2026 H1. In contrast, FinMem's return in China is concentrated in 2025 (31.68%) and falls to 2.90% in 2026 H1. A2C is comparatively strong in both China subperiods, but its run-level dispersion motivates examining the underlying seeds rather than relying only on the aggregate path.

Run-level variation. Tables 6 and 7 summarize multi-run methods by their mean and population standard deviation. Table 8 exposes the individual reported paths, while retaining the deterministic LightGBM path and the pre-specified seed-101 financial-system runs.

Table 3: Subperiod returns for the reported runs, mean ± population standard deviation. Full-period wealth compounds the daily returns; subperiod percentage-point gains therefore do not add to the full-period gain.

Market	Period	General return (↑%)	PortfolioRouter return (↑%)	Gain (↑ pp)
U.S.	2025	17.02 ± 3.70	24.49 ± 1.02	+7.47
U.S.	2026 H1	7.87 ± 1.79	8.33 ± 0.53	+0.46
U.S.	Full Period	26.17 ± 2.01	34.86 ± 0.55	+8.69
China	2025	20.25 ± 4.36	29.17 ± 4.19	+8.93
China	2026 H1	9.99 ± 1.42	12.86 ± 3.57	+2.86
China	Full Period	32.33 ± 6.44	45.92 ± 9.26	+13.59

Table 4: Complete U.S. results from 2025 through 2026 H1. Return and MDD are percentages; learned policies report mean ± population standard deviation over seeds 101/211/307.

Category	Method	Return (↑%)	Sharpe (↑)	MDD (↓%)
Rule-based Strategies	Equal-Weight Daily	33.89	1.26	-18.23
	Momentum (20 days)	32.97	1.36	-11.06
	SPY	25.80	0.96	-19.23
Fixed Agent	Validation-Selected Agent	21.73	1.10	-11.15
Supervised Routers	Shallow ERM Router	30.00 ± 5.45	1.33 ± 0.22	-12.80 ± 1.10
	Transformer Accuracy	26.79 ± 2.78	1.19 ± 0.12	-13.77 ± 0.85
	Neural Reward	26.59 ± 2.14	1.22 ± 0.08	-13.94 ± 0.45
	Boosted Reward	25.52	1.22	-12.70
	Ridge Reward	32.51	1.44	-14.03
	RouteLLM MF	30.59 ± 1.48	1.38 ± 0.06	-11.99 ± 0.42
	LightGBM	26.62	1.20	-12.07
	Qlib LSTM	31.12 ± 1.08	1.35 ± 0.04	-13.68 ± 0.78
RL Routers	PPO	28.28 ± 3.04	1.29 ± 0.10	-12.56 ± 0.41
	A2C	23.75 ± 1.89	1.19 ± 0.09	-12.30 ± 1.59
	DQN	27.41 ± 4.30	1.22 ± 0.17	-12.91 ± 1.34
Fixed-Agent Policies	Always GPT-5.4	30.20	1.47	-11.95
	Always Qwen3.6-Plus	25.56	1.22	-14.61
	Always DeepSeek-V4-Flash	29.41	1.26	-14.02
	Always Gemini-3.5-Flash	19.78	1.06	-11.03
PortfolioRouter	General Only	26.17 ± 2.01	1.24 ± 0.05	-12.89 ± 1.29
	PortfolioRouter (Ours)	34.86 ± 0.55	1.51 ± 0.04	-12.06 ± 0.76
	Base Intervention Branch	27.80 ± 2.40	1.31 ± 0.06	-12.89 ± 1.29
Matched Ablations	No Protection ($\lambda = 0$)	35.00 ± 0.92	1.52 ± 0.05	-11.96 ± 0.88
	Long History Only	30.62 ± 1.84	1.38 ± 0.06	-12.06 ± 0.76
	Mean Aggregation ($\rho = 0$)	31.16 ± 1.43	1.36 ± 0.05	-12.06 ± 0.76
	Replace Base Interventions	35.13 ± 0.86	1.52 ± 0.05	-12.06 ± 0.76
Reselected Controls	Mean Value ($\rho = 0$)	33.47 ± 0.89	1.46 ± 0.04	-12.06 ± 0.76
	Zero Value Threshold	25.10 ± 1.29	1.14 ± 0.05	-13.29 ± 0.48

The individual paths show substantial seed sensitivity for several learned routers. Qlib LSTM in China ranges from 20.80% to 36.85%, A2C in China from 34.87% to 50.28%, and DQN in the U.S. from 21.43% to 31.39%. These ranges explain the corresponding aggregate standard deviations and discourage interpreting a single best run as representative performance.

Common three-run comparison. Table 9 compares routing methods over the common seeds 101, 211, and 307. The single-run financial systems are omitted because they do not share this three-run reporting cohort.

With the common three-run cohort, PortfolioRouter is ahead of LightGBM, LSTM, PPO, A2C, and DQN in mean return and Sharpe in both markets (Table 9). Relative to the strongest official router

Table 5: Complete China results from 2025 through 2026 H1. Return and MDD are percentages; learned policies report mean ± population standard deviation over seeds 101/211/307.

Category	Method	Return (↑%)	Sharpe (↑)	MDD (↓%)
Rule-based Strategies	Equal-Weight Daily	30.20	1.17	-12.40
	Momentum (20 days)	43.95	1.49	-10.06
	510310	29.68	1.21	-11.76
Fixed Agent	Validation-Selected Agent	30.53	1.32	-10.54
Supervised Routers	Shallow ERM Router	31.18 ± 7.13	1.34 ± 0.28	-9.31 ± 1.26
	Transformer Accuracy	38.35 ± 2.35	1.52 ± 0.09	-10.69 ± 1.09
	Neural Reward	29.71 ± 3.73	1.30 ± 0.14	-10.92 ± 0.93
	Boosted Reward	31.06	1.35	-8.80
	Ridge Reward	29.16	1.27	-12.07
	RouteLLM MF	31.01 ± 2.98	1.34 ± 0.11	-10.81 ± 0.90
	LightGBM	30.57	1.26	-11.58
	Qlib LSTM	29.80 ± 6.70	1.26 ± 0.26	-10.77 ± 1.16
RL Routers	PPO	33.76 ± 2.24	1.40 ± 0.13	-9.50 ± 1.49
	A2C	40.49 ± 6.95	1.64 ± 0.21	-10.67 ± 1.73
	DQN	29.56 ± 2.26	1.29 ± 0.09	-10.78 ± 1.84
Fixed-Agent Policies	Always GPT-5.4	36.26	1.51	-10.27
	Always Qwen3.6-Plus	24.98	1.45	-8.95
	Always DeepSeek-V4-Flash	26.98	1.15	-10.54
	Always Gemini-3.5-Flash	30.52	1.25	-12.32
PortfolioRouter	General Only	32.33 ± 6.44	1.38 ± 0.24	-11.01 ± 1.41
	PortfolioRouter (Ours)	45.92 ± 9.26	1.87 ± 0.31	-8.67 ± 0.56
	Base Intervention Branch	35.76 ± 5.84	1.50 ± 0.19	-11.40 ± 0.97
Matched Ablations	No Protection ($\lambda = 0$)	44.68 ± 5.34	1.83 ± 0.17	-8.67 ± 0.56
	Long History Only	42.76 ± 6.83	1.77 ± 0.23	-10.09 ± 0.76
	Mean Aggregation ($\rho = 0$)	43.52 ± 10.97	1.78 ± 0.36	-8.67 ± 0.56
	Replace Base Interventions	45.77 ± 8.39	1.86 ± 0.27	-8.00 ± 0.16
Reselected Controls	Mean Value ($\rho = 0$)	45.61 ± 10.19	1.84 ± 0.33	-8.67 ± 0.56
	Zero Value Threshold	42.00 ± 5.14	1.75 ± 0.17	-10.24 ± 2.19

in each market, its return is 3.74 percentage points above LSTM in the U.S. and 5.43 points above A2C in China; the corresponding Sharpe ratios are 1.51 versus 1.35 and 1.87 versus 1.64.

D.3 Routing versus Fixed-Agent Selection

Fixed-agent controls test whether gains require date-specific selection. Table 10 compares PortfolioRouter with each agent held fixed throughout the period and with an agent selected for each quarter by validation portfolio utility. All use the same proposals, dates, and return alignment. PortfolioRouter has higher mean return and Sharpe than every fixed-agent policy in this comparison. Against the best fixed agent by full-period return, the gain is 4.65 percentage points in the U.S. and 9.66 in China. Drawdown rankings vary across the fixed-agent controls. Every policy uses the same cached proposals, so this comparison measures the value of selection at a common proposal-generation cost.

D.4 Switching-Mechanism Ablations

The ablations examine how protection, history aggregation, threshold calibration, and retention of base interventions affect portfolio performance. The matched no-protection and long-history variants keep the selected model family and architecture, change the indicated component, and recalibrate only the gate within its selected score type. The matched variant with ρ set to zero does the same for conservative aggregation. Two further controls reselect from all mean-value models or require a zero value threshold. All calibration uses the preceding quarter and the same three calibration runs. Allowing replacement of base interventions keeps the main model and gate fixed.

Table 6: Official-code financial baselines. Routing methods report mean ± population standard deviation for seeds 101/211/307; the financial systems use seed 101. Return and MDD are percentages. LightGBM reports one deterministic value.

Market	Category	Method	Return (↑%)	Sharpe (↑)	MDD (↓%)
U.S.	Supervised	LightGBM	26.62	1.20	-12.07
U.S.	Routers	Qlib LSTM	31.12 ± 1.08	1.35 ± 0.04	-13.68 ± 0.78
U.S.		PPO	28.28 ± 3.04	1.29 ± 0.10	-12.56 ± 0.41
U.S.	RL Routers	A2C	23.75 ± 1.89	1.19 ± 0.09	-12.30 ± 1.59
U.S.		DQN	27.41 ± 4.30	1.22 ± 0.17	-12.91 ± 1.34
U.S.	Financial	FinMem	38.38	1.34	-18.54
U.S.	Systems	FinGPT-Forecaster	44.87	1.49	-15.83
China	Supervised	LightGBM	30.57	1.26	-11.58
China	Routers	Qlib LSTM	29.80 ± 6.70	1.26 ± 0.26	-10.77 ± 1.16
China		PPO	33.76 ± 2.24	1.40 ± 0.13	-9.50 ± 1.49
China	RL Routers	A2C	40.49 ± 6.95	1.64 ± 0.21	-10.67 ± 1.73
China		DQN	29.56 ± 2.26	1.29 ± 0.09	-10.78 ± 1.84
China	Financial	FinMem	35.50	1.34	-12.13
China	Systems	FinGPT-Forecaster	33.29	1.16	-13.68

Table 7: Subperiod returns of the official-code baselines. Routing methods report mean ± population standard deviation for seeds 101/211/307; the financial systems use seed 101.

Market	Category	Method	2025 (↑%)	2026 H1 (↑%)	Full Period (↑%)
U.S.	Supervised	LightGBM	19.94	5.57	26.62
U.S.	Routers	Qlib LSTM	19.00 ± 0.95	10.18 ± 0.06	31.12 ± 1.08
U.S.		PPO	15.79 ± 2.60	10.80 ± 2.24	28.28 ± 3.04
U.S.	RL Routers	A2C	13.34 ± 2.52	9.21 ± 0.75	23.75 ± 1.89
U.S.		DQN	19.52 ± 2.52	6.56 ± 1.38	27.41 ± 4.30
U.S.	Financial	FinMem	21.70	13.70	38.38
U.S.	Systems	FinGPT-Forecaster	26.14	14.85	44.87
China	Supervised	LightGBM	13.37	15.18	30.57
China	Routers	Qlib LSTM	14.66 ± 3.29	13.14 ± 3.00	29.80 ± 6.70
China		PPO	23.66 ± 1.23	8.17 ± 1.50	33.76 ± 2.24
China	RL Routers	A2C	25.27 ± 0.31	12.14 ± 5.37	40.49 ± 6.95
China		DQN	17.75 ± 2.36	10.03 ± 0.85	29.56 ± 2.26
China	Financial	FinMem	31.68	2.90	35.50
China	Systems	FinGPT-Forecaster	22.64	8.69	33.29

Using only long history reduces return from 34.86% to 30.62% in the U.S. and from 45.92% to 42.76% in China. Removing protection reduces the return in China to 44.68% but slightly raises U.S. return to 35.00%. The policies with ρ set to zero earn 31.16% and 43.52%, while reselecting the whole mean-value family earns 33.47% and 45.61%. A zero threshold also lowers returns in both markets. Multiple histories, conservative aggregation, and threshold calibration improve mean returns in both markets; protection does so only in China. Allowing replacement of base interventions slightly raises U.S. return to 35.13% and lowers the return in China to 45.77%.

Removing any of the four tasks lowers both mean return and Sharpe in both markets. The return reduction ranges from 1.04 to 11.21 percentage points in the U.S. and from 4.83 to 12.61 percentage points in China (Table 2).

Table 8: All reported runs of the official-code baselines from 2025 through 2026 H1. Return and MDD are percentages. LightGBM produces the same deterministic path for each nominal seed; the financial systems use seed 101.

Market	Category	Method	Seed	Return (↑%)	Sharpe (↑)	MDD (↓%)
China	Supervised Routers	LightGBM	101	30.57	1.26	-11.58
China		LightGBM	211	30.57	1.26	-11.58
China		LightGBM	307	30.57	1.26	-11.58
China		Qlib LSTM	101	36.85	1.52	-9.37
China		Qlib LSTM	211	20.80	0.91	-12.21
China		Qlib LSTM	307	31.76	1.35	-10.74
China	RL Routers	PPO	101	30.61	1.22	-11.08
China		PPO	211	35.62	1.53	-7.51
China		PPO	307	35.03	1.46	-9.89
China		A2C	101	34.87	1.46	-10.54
China		A2C	211	36.31	1.54	-12.85
China		A2C	307	50.28	1.93	-8.63
China		DQN	101	30.01	1.35	-8.41
China		DQN	211	32.06	1.36	-11.02
China		DQN	307	26.59	1.17	-12.91
China	Financial Systems	FinMem	101	35.50	1.34	-12.13
China		FinGPT-Forecaster	101	33.29	1.16	-13.68
U.S.	Supervised Routers	LightGBM	101	26.62	1.20	-12.07
U.S.		LightGBM	211	26.62	1.20	-12.07
U.S.		LightGBM	307	26.62	1.20	-12.07
U.S.		Qlib LSTM	101	30.86	1.36	-12.64
U.S.		Qlib LSTM	211	32.56	1.39	-14.51
U.S.		Qlib LSTM	307	29.94	1.29	-13.90
U.S.	RL Routers	PPO	101	24.21	1.18	-12.63
U.S.		PPO	211	29.12	1.27	-13.03
U.S.		PPO	307	31.50	1.41	-12.03
U.S.		A2C	101	22.93	1.15	-12.57
U.S.		A2C	211	26.37	1.31	-10.24
U.S.		A2C	307	21.96	1.10	-14.09
U.S.		DQN	101	21.43	0.98	-14.71
U.S.		DQN	211	31.39	1.36	-11.48
U.S.		DQN	307	29.40	1.31	-12.56
U.S.	Financial Systems	FinMem	101	38.38	1.34	-18.54
U.S.		FinGPT-Forecaster	101	44.87	1.49	-15.83

Table 9: Common three-run comparison, using seeds 101, 211, and 307. Entries are mean returns (percent) and Sharpe ratios; LightGBM is deterministic.

Category	Method	U.S. Market		China Market	
		Return (↑%)	Sharpe (↑)	Return (↑%)	Sharpe (↑)
Supervised Routers	LightGBM	26.62	1.20	30.57	1.26
	Qlib LSTM	31.12 ± 1.08	1.35 ± 0.04	29.80 ± 6.70	1.26 ± 0.26
RL Routers	PPO	28.28 ± 3.04	1.29 ± 0.10	33.76 ± 2.24	1.40 ± 0.13
	A2C	23.75 ± 1.89	1.19 ± 0.09	40.49 ± 6.95	1.64 ± 0.21
	DQN	27.41 ± 4.30	1.22 ± 0.17	29.56 ± 2.26	1.29 ± 0.09
PortfolioRouter	General Only	26.17 ± 2.01	1.24 ± 0.05	32.33 ± 6.44	1.38 ± 0.24
	PortfolioRouter (Ours)	34.86 ± 0.55	1.51 ± 0.04	45.92 ± 9.26	1.87 ± 0.31

Table 10: Routing versus fixed-agent policies from 2025 through 2026 H1. Return is in percent. Fixed-agent rows are deterministic; PortfolioRouter reports mean ± population standard deviation over the reported runs.

Category	Method	U.S. Market		China Market	
		Return (↑%)	**Sharpe (↑)**	**Return (↑%)**	**Sharpe (↑)**
Fixed-Agent Policies	Always GPT-5.4	30.20	1.47	36.26	1.51
	Always Qwen3.6-Plus	25.56	1.22	24.98	1.45
	Always DeepSeek-V4-Flash	29.41	1.26	26.98	1.15
	Always Gemini-3.5-Flash	19.78	1.06	30.52	1.25
	Validation-Selected Agent	21.73	1.10	30.53	1.32
PortfolioRouter	**PortfolioRouter (Ours)**	34.86 ± 0.55	1.51 ± 0.04	45.92 ± 9.26	1.87 ± 0.31

D.5 Mechanism Evidence

D.5.1 Critical-Date Contributions and Switch Timing

Paired attribution locates the gains over General, and randomized timing examines whether the dates of additional switches matter beyond their count. Both analyses compare policies on identical dates and proposals. For this attribution, a date is critical if its realized top-two simple-return gap is at least the 80th percentile of history available before the preceding validation quarter. Let $\delta_t = \log(1 + r_{t,a_t}) - \log(1 + r_{t,a_t^{\mathrm{G}}})$. Equation 2 gives the exact identity

$$\log W^{\mathrm{router}} - \log W^{\mathrm{G}} = \sum_{t:y_t^{\mathrm{D}}=1} \delta_t + \sum_{t:y_t^{\mathrm{D}}=0} \delta_t. \tag{12}$$

We also replay two hindsight paths: retain only critical-date switches, or retain only ordinary-date switches. These hindsight replays attribute realized gains under the zero-cost daily allocation protocol. Equation 12 adds log contributions exactly; cumulative-return gains compound instead.

Critical-date switches contribute 312.11 log basis points in the U.S. and 807.05 in China, both positive. Keeping only those switches adds 3.99 and 11.10 percentage points of cumulative return over General. Critical-date regret also falls in both markets. These gains depend on the size of wins and losses: override win rates are 55.41% and 54.58%, and overall accuracy in China rises from 30.21% to 30.97%. Critical-date gains account for most of the improvement in China, while ordinary dates account for just over half of the U.S. log gain. This is consistent with the distinction in Section 3: even a small top-two gap can leave room to improve on General.

A second diagnostic holds the number of additional value-branch switches fixed within each quarter and run. We randomly choose that many dates from the same model's positive-value candidates, leaving the candidate on each date unchanged. Across 2,000 replays, the U.S. observed log gain is 539.06 basis points versus 206.93 on average under random timing; the random interval from the 2.5th to the 97.5th percentile is [−23.88, 445.42]. In China the corresponding values are 711.53, 396.67, and [153.58, 642.66]. The observed values lie above the random ranges in both markets. The replay is a descriptive timing diagnostic for additional switches, conditional on the selected models and runs.

D.5.2 Mechanism Visualization

Switching gains concentrate on high-disagreement dates. Figure 3 links additional value-based switches to disagreement among the agents' proposed portfolios. High-disagreement dates comprise only 25.3% of U.S. dates and 13.9% in China, yet account for 38.1% and 55.2% of the added switches' net log-wealth gains. The concentration is stronger in China; additional switching is more frequent on high-disagreement dates in both markets.

These switches contribute to the return improvement. Additional switches raise mean return over the base policy by 7.06 percentage points in the U.S. and 10.16 in China. Removing those on high-disagreement dates reduces final return by 2.74 and 5.68 points, respectively (Figure 4). The full policy remains above General by 8.69 and 13.59 points on this three-run cohort.

Table 11: Paired switching attribution relative to the same General path, mean ± population standard deviation over the reported runs. A log basis point is one ten-thousandth of log wealth. Counts refer to the complete period per run.

Diagnostic	U.S. Market	China Market
Full Cumulative-Return Gain (pp)	8.69 ± 1.72	13.59 ± 8.48
Critical Switches Only: Gain (pp)	3.99 ± 1.19	11.10 ± 6.32
Ordinary Switches Only: Gain (pp)	4.55 ± 0.55	2.23 ± 2.41
Critical Contribution (log bps)	312.11 ± 95.05	807.05 ± 461.43
Ordinary Contribution (log bps)	355.11 ± 47.83	163.38 ± 171.15
Activation Rate (%)	28.52 ± 1.34	33.43 ± 1.17
Override Win Rate (%)	55.41 ± 0.95	54.58 ± 3.55
Critical-Day Regret: General (bps/day)	35.31 ± 2.33	56.02 ± 6.49
Critical-Day Regret: PortfolioRouter (bps/day)	31.44 ± 1.39	46.68 ± 5.09
Correct General Choices Made Incorrect	27.67 ± 3.77	31.00 ± 3.56
Incorrect General Choices Made Correct	33.00 ± 0.82	33.67 ± 4.50

D.6 Objective and Predictor Controls

D.6.1 Learning Correctness or Switching Value

The objective comparison tests whether learning a replacement's economic value improves on learning its correctness on the same full-universe panels as the main results (23 U.S. ETFs and 25 China ETFs). It reports seeds 101/211/307. Three matched 128-unit neural value models use ordinary correctness, critical-date-weighted correctness, or supervision based on economic benefit and harm. The architecture, inputs, initializations, 80-update budget, auxiliary magnitude and gap losses, history aggregation, and deployment calibration are shared; only the direction objective changes. Critical-date weighting multiplies the classification weight by five on critical dates. The full PortfolioRouter is included as a system reference because it can also select tree estimators.

Economic supervision has the highest mean return in China among the matched objectives (42.84% versus 39.88% for both correctness variants), whereas ordinary correctness has the highest U.S. mean return (28.99%). The large cross-seed standard deviations of several critical-gain estimates counsel against treating their signs as stable effects. The full-system reference, which can also select tree estimators, has the highest mean return in both markets.

D.6.2 Alternative Predictors and Matched Switching

Alternative predictors test how estimator choice affects the switching framework on the same full-universe panels and seeds 101/211/307. The corresponding Transformer accuracy and neural reward predictors route directly on the four-agent proposal pool. We also place these predictors and a gradient-boosted reward predictor inside the same General, base-intervention, and validation-gate framework. Predictor configurations are the frozen preceding-quarter choices from the main full-universe baseline study.

PortfolioRouter has the highest mean return and Sharpe among these matched controls in both markets. The strongest alternative is the boosted predictor in the U.S. (30.75% mean return) and the Transformer predictor in China (39.64%); PortfolioRouter reaches 34.86% and 45.92%, respectively. Appendix D.6.3 retains all predictor families, training budgets, and diagnostic variants.

D.6.3 Complete Control Matrix

All controls use the same full-universe panels as the main experiments: 23 U.S. ETFs and 25 China ETFs. The summaries use seeds 101/211/307. All comparisons use the same full PortfolioRouter configuration, with 128-unit hidden layers in its neural value models.

The matched objective study changes only the direction loss: ordinary binary correctness, critical-date-weighted correctness (weight five), or binary cross-entropy weighted by benefit and harm. Every variant also learns the same magnitude and gap targets. The primary budget is 80 updates;

Table 12: Full-universe matched neural direction objectives and the corresponding full-system reference. Entries are mean ± population standard deviation over seeds 101/211/307 for return (percent) and critical-date log gain over General (basis points).

Category	Objective	U.S. Market (23 ETFs)		China Market (25 ETFs)	
		Return (↑%)	Critical gain (↑)	Return (↑%)	Critical gain (↑)
PortfolioRouter	**PortfolioRouter (Ours)**	34.86 ± 0.55	312.11 ± 95.05	45.92 ± 9.26	807.05 ± 461.43
Objective Controls	Ordinary Correctness	28.99 ± 1.20	66.33 ± 116.22	39.88 ± 5.65	509.67 ± 559.62
	Critical-Weighted Correctness	27.77 ± 2.61	108.49 ± 183.41	39.88 ± 4.10	397.46 ± 508.46
	Economic Switching Value	27.66 ± 1.85	29.38 ± 99.10	42.84 ± 8.37	517.99 ± 612.66

Table 13: Full-universe alternative predictors under the same switching framework from 2025 through 2026 H1. Entries are mean ± population standard deviation over seeds 101/211/307 for return (percent) and Sharpe ratio. The boosted-tree control uses scikit-learn HistGradientBoosting.

Category	Predictor	U.S. Market (23 ETFs)		China Market (25 ETFs)	
		Return (↑%)	Sharpe (↑)	Return (↑%)	Sharpe (↑)
PortfolioRouter	**PortfolioRouter (Ours)**	34.86 ± 0.55	1.51 ± 0.04	45.92 ± 9.26	1.87 ± 0.31
Gated Routers	Transformer Accuracy + Gate	26.34 ± 1.30	1.18 ± 0.04	39.64 ± 7.05	1.60 ± 0.19
	Neural Reward + Gate	28.79 ± 3.50	1.35 ± 0.11	32.45 ± 7.00	1.40 ± 0.22
	Boosted Reward + Gate	30.75 ± 0.68	1.42 ± 0.01	32.61 ± 6.30	1.41 ± 0.19

240-update checks and zero-threshold diagnostics provide additional comparisons. The full PortfolioRouter reference also selects among estimator families.

The predictor comparison includes the Transformer classifier, reward MLP, scikit-learn HistGradientBoosting, Ridge, and a financial adaptation of RouteLLM's matrix-factorization scoring rule. The latter replaces the text embedding with numeric market state and compares the four portfolio agents. Each predictor is evaluated both directly and within the matched switching framework, using the same trained model and configuration selected on the preceding quarter for the main baseline comparison.

Three-seed control matrix. Table 14 expands the focused objective and predictor comparisons above to the complete control suite over the primary reporting seeds. It includes all predictor families, longer training budgets, and zero-threshold diagnostics under the same full-universe protocol.

PortfolioRouter retains the highest mean return in both markets. Ridge Reward is the closest U.S. alternative at 32.51%, while Economic Switching Value is the closest objective control in China at 42.84%, compared with PortfolioRouter's 34.86% and 45.92%. Increasing the objective-training budget to 240 updates or fixing the gate threshold at zero does not yield a consistent improvement across markets. The large cross-seed standard deviations of several critical-gain estimates also caution against treating their signs as stable effects.

Table 14: Full-universe complete objective and predictor controls over the primary seeds 101/211/307. Entries are mean ± population standard deviation for return (percent) and critical-date log gain over General (basis points). Additional rows vary the training budget and gate threshold.

Category	**Method**	**U.S. Market (23 ETFs)**		**China Market (25 ETFs)**	
		Return (↑%)	**Critical gain (↑)**	**Return (↑%)**	**Critical gain (↑)**
Supervised Routers	Transformer Accuracy	26.79 ± 2.78	-22.72 ± 171.84	38.35 ± 2.35	539.90 ± 941.27
	Neural Reward	26.59 ± 2.14	-39.35 ± 250.53	29.71 ± 3.73	-74.37 ± 758.64
	Boosted Reward	25.52 ± 0.00	-15.61 ± 189.42	31.06 ± 0.00	199.61 ± 556.11
	RouteLLM MF Adaptation	30.59 ± 1.48	195.61 ± 163.54	31.01 ± 2.98	85.57 ± 616.51
	Ridge Reward	32.51 ± 0.00	35.76 ± 189.42	29.16 ± 0.00	182.98 ± 556.11
PortfolioRouter	General Only	26.17 ± 2.01	0.00 ± 0.00	32.33 ± 6.44	0.00 ± 0.00
	Base Intervention	27.80 ± 2.40	30.45 ± 91.51	35.76 ± 5.84	145.43 ± 241.93
	PortfolioRouter (Ours)	34.86 ± 0.55	312.11 ± 95.05	45.92 ± 9.26	807.05 ± 461.43
Objective Controls	Ordinary Correctness	28.99 ± 1.20	66.33 ± 116.22	39.88 ± 5.65	509.67 ± 559.62
	Critical-Weighted Correctness	27.77 ± 2.61	108.49 ± 183.41	39.88 ± 4.10	397.46 ± 508.46
	Economic Switching Value	27.66 ± 1.85	29.38 ± 99.10	42.84 ± 8.37	517.99 ± 612.66
Budget Controls	Ordinary Correctness (240 Updates)	31.31 ± 2.30	25.19 ± 269.52	38.07 ± 3.86	363.83 ± 415.22
	Critical-Weighted Correctness (240 Updates)	26.83 ± 2.89	-77.32 ± 294.33	37.09 ± 3.35	245.45 ± 327.87
	Economic Switching Value (240 Updates)	30.70 ± 2.53	36.24 ± 350.91	40.03 ± 7.97	428.60 ± 590.00
Gate Controls	Ordinary Correctness (Zero Threshold)	24.52 ± 1.11	-14.16 ± 108.68	28.40 ± 4.94	-62.68 ± 774.34
	Critical-Weighted Correctness (Zero Threshold)	29.63 ± 2.91	334.94 ± 178.48	30.70 ± 4.74	25.34 ± 780.91
	Economic Switching Value (Zero Threshold)	27.89 ± 1.17	48.94 ± 141.21	30.99 ± 6.54	-0.30 ± 861.07
Gated Routers	Transformer Accuracy + Gate	26.34 ± 1.30	17.34 ± 190.82	39.64 ± 7.05	381.73 ± 372.48
	Neural Reward + Gate	28.79 ± 3.50	-14.51 ± 121.87	32.45 ± 7.00	-139.94 ± 359.32
	Boosted Reward + Gate	30.75 ± 0.68	138.84 ± 92.07	32.61 ± 6.30	-94.68 ± 482.82
	RouteLLM MF Adaptation + Gate	29.13 ± 1.91	38.82 ± 250.69	31.26 ± 4.08	68.21 ± 343.49
	Ridge Reward + Gate	26.39 ± 2.16	-157.76 ± 63.07	32.27 ± 5.58	191.65 ± 329.57