LEARNING WHEN AND HOW TO TRADE ON GOVERNMENT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Government policies influence financial markets, but supportive language does not necessarily imply profitable trading opportunities. We study when policy information adds value to existing trading strategies and how this value evolves across trading days. We construct PolicyValue-Bench, a point-in-time benchmark aligning 5,125 government policy documents from 2021–2025 with policy histories and market data for five Chinese A-share sector ETFs. We present PolicyState, a stateful gating framework that assesses policy text and LLM-generated judgments in the context of market conditions and retained policy effects. It learns policy admission, opportunity retention, and daily risk release through separate opportunity and risk states that guide exposure increases and cash exits. Training combines matched-control return supervision with sequential trading optimization. In a full-year 2025 out-of-sample evaluation across four base strategies, our method achieves an average net cumulative return of 16.25%, representing relative improvements of 140.4% over matched bases and 54.0% over the strongest comparison method. It improves returns in 19 of 20 sector-strategy combinations.

1 INTRODUCTION

Government policies are an important source of information in the Chinese A-share market. Industrial plans, regulatory measures, fiscal support, and sector-specific guidance reshape expectations about cash flows, risk, and capital allocation, affecting prices in related industries (Megginson et al., 2022; Liu et al., 2021; Cao et al., 2024; Crowley et al., 2019). Advances in financial language models and LLM-based trading agents (Lopez-Lira & Tang, 2026; Yu et al., 2025; Zhang et al., 2024) offer opportunities to interpret these documents and inform trading decisions. We study when policy information adds trading value to an existing strategy and how that value guides exposure over time.

A policy's supportive or restrictive stance alone does not determine its trading value. Supportive measures may respond to existing industry difficulties, while repeated or anticipated announcements may add little new information. This distinction is consistent with evidence that markets respond to unexpected policy changes (Kuttner, 2001; Bernanke & Kuttner, 2005) and react less strongly to known information (Tetlock, 2011). As illustrated schematically in Fig. 1, supportive announcements are able to warrant avoiding additional exposure when benefits are already priced in, reducing exposure as a temporary rally fades, or exiting to cash when downside risk persists. A model therefore uses current market conditions and the remaining effects of earlier policies to infer *when to act, how long to retain policy influence, and when risk should take priority*.

Existing resources cover policy retrieval (Wang et al., 2024), monetary policy stance and trading (Shah et al., 2023), financial-agent investment decisions (Li et al., 2025a; Ma et al., 2025), and news-based macro and sector allocation (Chen et al., 2026). However, they do not provide a dedicated setting for evaluating whether industry-policy information improves existing trading strategies under matched execution rules and costs. Such an evaluation requires aligning policy announcements with their histories and contemporaneous market data, while comparing strategies with and without policy.

To support this evaluation, we construct **PolicyValue-Bench**, a point-in-time benchmark of government policies in the Chinese A-share market. It contains 5,125 policy documents released during 2021-2025, aligned with policy histories and market data for five sector ETFs. Four policy-independent base strategies provide references for measuring net return gains from policy information

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.

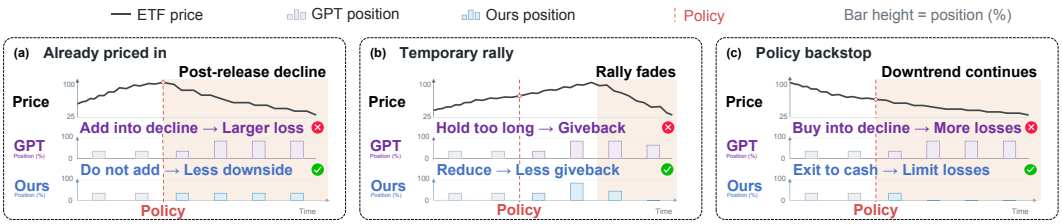


Figure 1: Schematic comparison of policy-driven trading under three policy-market scenarios.

under identical execution rules and costs. Daily targets are generated after the close and submitted for execution at the next trading day's open.

We present **PolicyState**, a stateful gating framework that learns policy admission, opportunity retention, and daily risk release. It assesses new policies by combining policy content and LLM-generated assessments with current market features and the remaining influence of earlier policies. These estimates update separate opportunity and risk states across trading days. The opportunity state guides exposure increases over the base strategy, while the risk state triggers a cash exit even when positive policy signals remain. Both states inform the assessment of subsequent policies. Training first initializes policy-value estimates through matched-control return supervision, then jointly refines value estimation, state updates, and exposure decisions by optimizing net trading returns relative to the base strategy.

In a full-year 2025 out-of-sample evaluation across five sector ETFs and four base strategies, our method raises average net cumulative return from 6.76% to 16.25%, a relative improvement of 140.4%. It improves returns in 19 of 20 sector-strategy combinations and reduces average maximum drawdown from 11.44% to 6.38%. Component and training-stage ablations support the contributions of policy information, market context, recurrent states, and two-stage optimization.

Our contributions are summarized as follows:

- We build PolicyValue-Bench, comprising 5,125 government policy documents from 2021-2025, aligned with policy histories and market data for five A-share sector ETFs, with matched base strategies for evaluating incremental trading value.
- We design PolicyState, a stateful gating framework that uses separate opportunity and risk states to learn when to admit policy signals, how long to retain opportunities, and when to prioritize risk control.
- Experiments show relative improvements in average net cumulative return of 140.4% over matched bases and 54.0% over the strongest comparison method. Our method improves returns in 19 of 20 sector-strategy combinations and lowers average maximum drawdown relative to matched bases.

2 Related Work

2.1 Financial Datasets and Benchmarks

Financial-language resources cover news, filings, and policy documents, supporting tasks from text understanding to investment evaluation(Boudoukh et al., 2013; Loughran & McDonald, 2011; Wang et al., 2024; Li et al., 2025a). Policy-focused resources study Chinese policy retrieval (Wang et al., 2024) and monetary policy stance classification, including stance-based trading (Shah et al., 2023). Recent benchmarks extend evaluation to sequential financial decisions. InvestorBench offers multiple financial decision tasks for LLM agents (Li et al., 2025a). Agent Trading Arena tests numerical reasoning through market interaction (Ma et al., 2025). StockBench (Chen et al., 2025) evaluates sequential trading over multiple months, showing that strong financial reasoning does not consistently translate into profitable trading. SocioDojo (Cheng & Chin, 2024) provides a lifelong learning environment combining real-world text and time series for analytical decision making. Closer to our setting, CN-Buzz2Portfolio (Chen et al., 2026) studies macro and sector allocation in the Chinese market using daily financial news. However, these resources give limited attention to how policy can guide adjustments to existing trading strategies. This gap is particularly important in China's A-share

market, where policies can significantly affect stock prices and downside risk (Megginson et al., 2022; Xie et al., 2022). Our benchmark aligns government policy releases with policy histories, sector ETFs, and point-in-time market data, and measures net return gains over matched policy-independent strategies under a shared execution protocol.

2.2 Financial Prediction and Trading Methods

Financial text complements market data (Boudoukh et al., 2013), with dictionaries and pretrained models supporting semantic analysis (Loughran & McDonald, 2011; Zhang et al., 2021) and news-driven systems optimizing portfolios and risk-sensitive objectives (Yang et al., 2023; Liang et al., 2021). CausalStock (Li et al., 2024) combines news encoding with temporal causal discovery for stock movement prediction. LLMs extract return-related signals (Lopez-Lira & Tang, 2026), while LLM as a Risk Manager (Kim et al., 2026) filters lead-lag relationships and evaluates gains over a statistical baseline under a fixed trading protocol. VTA (Koa et al., 2026) combines verbal and latent reasoning over price histories, while STABLE (Soun et al., 2026) generates context-conditioned return distributions for portfolio allocation. Recently, trading agents add memory, tools, and feedback. FinMem (Yu et al., 2025) uses layered memory, decay, and reflection; FinAgent (Zhang et al., 2024) combines multimodal information with retrieval, tools, and multi-horizon reflection. FLAG-Trader (Xiong et al., 2025) learns an LLM trading policy through reinforcement learning. QuantAgents (Li et al., 2025b) coordinates market analysis, risk control, and strategy evaluation, while TiMi (Song et al., 2026)separates multi-agent strategy development from minute-level deployment. Decision-focused learning further aligns predictions with downstream objectives through decision ranking (Mandi et al., 2022) or learned surrogate losses, including applications to portfolio optimization (Zharmagambetov et al., 2023). These approaches address information filtering, temporal memory, and decision quality, but do not specifically study how to jointly learn industry-policy admission and its persistent opportunity and risk effects within an existing strategy. Our framework adopts matched-control return supervision and sequential trading optimization to learn policy admission, opportunity retention, and daily risk release through separate states that guide exposure increases and cash exits relative to a base strategy.

3 Our Dataset

To evaluate the trading value of government policies, we construct **PolicyValue-Bench**, aligning 5,125 policy documents from 2021-2025 with policy histories, market data, and five A-share sector ETFs. The task is to decide when to adjust a base strategy's exposure, how long to retain the adjustment, and when to exit to cash, with long-only exposure constrained to $[0, 1]$. We measure the added value as $\Delta R = R^{\text{with policy}} - R^{\text{without policy}}$, the difference in net cumulative returns on the same assets and periods under identical transaction costs and execution constraints. We average ΔR across 20 sector-strategy combinations using four policy-independent bases: buy-and-hold (B&H), mean reversion (MR), 20-day trend (TREND), and DEFENSIVE.

3.1 Dataset Construction and Statistics

Data collection and sector selection. We collect policy documents from the Chinese government's policy database and official government and regulatory websites, covering January 1^{st}, 2021 to December 31^{st}, 2025. Since these policies often target industries, we use sector ETFs to represent their trading exposure and reduce firm-specific fluctuations. We rank sectors by policy density using only the 2021–2023 training-period corpus. The five highest-density sectors are automobiles, computers, healthcare, securities, and transportation, each represented by a corresponding ETF; documents from 2024–2025 play no role in this selection.

Policy processing. For each document, we retain its publication date, original text, and source link, and extract its summary, key measures, policy direction, and affected sectors. Documents covering multiple sectors are assigned to each relevant sector while maintaining links to the original source. Documents with the same sector and date are grouped into a single sector-level policy event day.

Quality review. Three reviewers manually check a random sample of 1,000 sector-level records. The review covers policy summaries, sector assignments, and policy direction. Errors identified during the review are corrected in the dataset.

Table 1: **PolicyValue-Bench examples and statistics.** Policy descriptions are abbreviated English translations. Prices are listed as previous close / open / close in RMB per ETF unit. Policy days count sector-date pairs with at least one policy.

(a) Examples	**Automobiles / 516110.SH**	**Computers / 512720.SH**
Policy	2025 automobile trade-in program	Q&A on facial-recognition security rules
Source	MOFCOM and seven departments	Chinese Government Website
Policy date	2025-01-17	2025-03-22
Market date	2025-01-20	2025-03-24
Key measures	Scrappage-renewal subsidies: RMB 20,000 for eligible new-energy vehicles; RMB 15,000 for fuel vehicles.	Personal information protection, registration, and limits on compulsory facial verification.
Prices	1.256 / 1.266 / 1.267	1.140 / 1.139 / 1.132
(b) Dataset overview		
Coverage	2021-2025; 5,125 documents; 2,150 policy days; 5 sector ETFs	
Sectors	Automobiles, computers, healthcare, securities, transportation	
Policy fields	Publication date, source, title, summary, key measures	
Market fields	ETF market data aligned by date	
Quality review	1,000 sector-level records; 3 reviewers; identified errors corrected	

Table 2: Policy semantics and five-day excess returns relative to the CSI 300 ETF. 95% confidence intervals use date-clustered bootstrap resampling.

(a) Positive-policy outcomes

Five-day excess return	n	Share (%)
Positive (> 0)	845	45.95
Non-positive (≤ 0)	994	54.05
Total	1,839	100.00

(b) Returns by policy semantics

Semantics	n	Mean (%)	95% CI (%)
Negative	66	-0.10	[-0.60, 0.38]
Neutral	177	-0.17	[-0.52, 0.18]
Positive	1,839	-0.04	[-0.18, 0.09]
Mixed	68	-0.48	[-1.13, 0.13]

Temporal alignment. We align policy events with market data to form a daily decision stream for each sector. New policies enter as current inputs, while earlier policies remain available as historical context. Policy availability follows the recorded publication date, with policies released on non-trading days entering the stream on the next trading day. On day t, inputs include policies available by that date and market observations through the day-t close. Orders based on these inputs are eligible for execution at the next trading day's open.

Dataset statistics. The benchmark contains 2,150 policy event days across five sectors, each representing a sector-date pair with at least one policy. Table 1 summarizes the dataset's coverage, fields, and quality review, alongside two examples: an automobile trade-in program and a Q&A on facial-recognition security rules. Each example pairs the policy's source, publication date, and key measures with the corresponding sector ETF and dated market prices.

3.2 Benchmark Analysis

Policy direction and market returns. We examine five-day sector ETF returns following policy events in 2021-2025, measured relative to the CSI 300 ETF. Among 1,839 samples containing supportive policies, 54.05% do not outperform the market benchmark over the following five trading days (Table 2 (a)). The mean five-day excess return is -0.044% for supportive policies and approximately -0.10% for restrictive policies, with substantially overlapping 95% confidence intervals (Table 2 (b)). These results suggest that policy direction alone is insufficient for identifying trading opportunities.

Policy information and trading value. Policy information can help identify opportunities or avoid losses. For example, a restrictive policy may signal downside risk and support an early exit. Its trading value therefore depends on the resulting decisions, which we evaluate through return improvements over the matched base strategies. In the 2025 evaluation across 20 sector-strategy accounts, Dictionary, Chinese financial PLM, and DeepSeek-V4-Flash improve returns by 3.63, 3.79, and 2.78 percentage

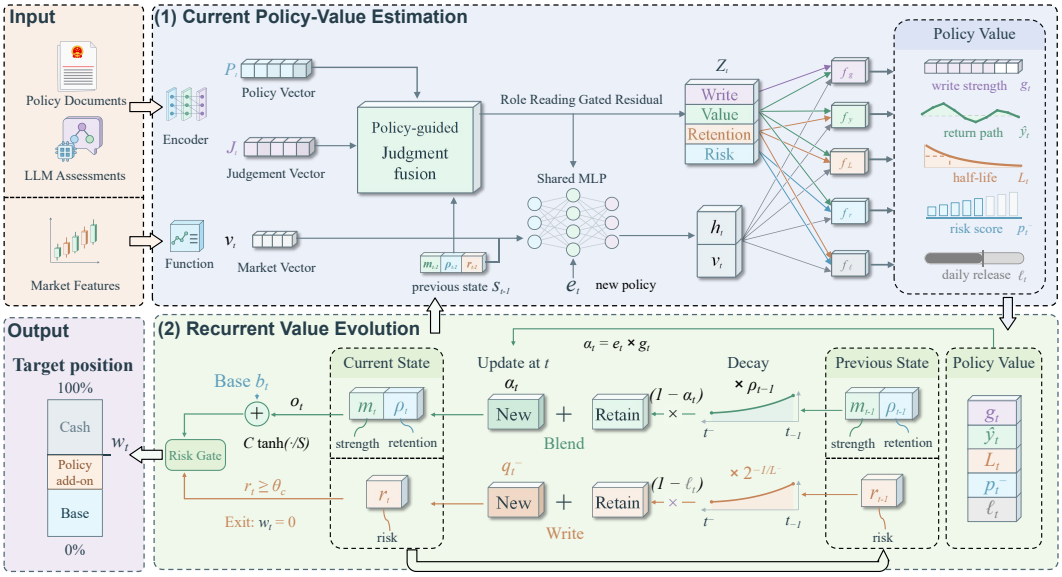


Figure 2: Overview of PolicyState. (1) Current policy-value estimation combines policy text, LLM judgments, market context, and prior states to predict opportunity and risk signals. (2) Recurrent value evolution updates separate opportunity and risk states through admission, decay, and risk release. These states guide exposure increases and cash exits, and inform subsequent policy assessment.

points, respectively FinMem achieves a smaller gain of 1.49 points, while Text-to-return and FinAgent reduce returns by 7.91 and 1.26 points, respectively (Table 3). The variation across methods highlights the need to learn how policy information should affect exposure and how its influence should change over time.

4 Methodology

Trading on policy direction alone overlooks whether information is already reflected in prices and how its value changes over time. We present PolicyState, a stateful gating framework that jointly learns when policy information should change a base strategy and how its influence should evolve, using market conditions and prior policy effects to guide selective policy admission, opportunity retention, and risk-triggered exits.

As shown in Fig. 2, PolicyState maps policy documents, LLM assessments, and market features to a target position $w_t \in [0, 1]$ for each sector ETF. First, a frozen LLM encoder separately encodes policy text and assessments, and policy-guided judgment fusion forms role-specific features. Together with shared market and state context, these features predict write strength g_t, return path $\hat{y}_{t,1:10}$, opportunity half-life L_t, and downside risk p_t^-; a separate head predicts daily risk release ℓ_t. Second, these predictions update opportunity strength m_t, retention rate ρ_t, and accumulated risk r_t. Opportunities decay daily and blend with newly admitted value on policy-event days, with L_t guiding retention. Risk decays and is selectively released each day, while new risk is added only when a new policy meets the risk-write condition. Third, the opportunity state adds exposure o_t to the base position b_t, with the sum clipped to $[0, 1]$. The risk state overrides this target to zero when the exit threshold is reached. Updated states inform the next day's estimation, while targets set after the close are submitted for the next open. *Note that the framework is able to be optimized in an end-to-end manner.*

In the following subsections, we first describe policy-value estimation, then explain recurrent state updates and exposure decisions, and finally describe the two-stage training procedure.

4.1 Current Policy Value Estimation

The trading value of a new policy depends on its content, current market conditions, and the opportunities and risks retained from earlier policies. As shown in Fig. 2(1), PolicyState first uses policy

content to select relevant LLM interpretations for different prediction roles, then combines these representations with market and state context to guide subsequent trading.

Policy-guided judgment fusion. Policy text provides the primary source, while LLM judgments offer supplementary interpretations whose relevance may differ across prediction tasks. On policy-event days, a shared frozen Qwen encoder separately encodes current policies with historical context and their LLM judgments into P_t and J_t, followed by separate projections. First, four learnable queries retrieve policy features for writing, value, retention, and risk. Second, each role uses its policy feature to retrieve relevant judgment information. Third, a gated residual connection adds a learned judgment increment to the policy feature. The gate depends on the policy feature, retrieved judgment feature, their elementwise absolute difference, and the previous state s_{t-1}. This allows each role to selectively adopt LLM interpretations according to policy content and retained context. The fusion gate controls judgment adoption, whereas g_t controls subsequent opportunity writing. The resulting role vectors form $Z_t = [z_{t,1}; z_{t,2}; z_{t,3}; z_{t,4}]$.

Context-conditioned value prediction. To assess these features in the current trading context, we encode market information available by the day-t close as $v_t = f_x(x_t)$ and use $e_t \in \{0, 1\}$ to indicate new policy arrivals. The previous state $s_{t-1} = (m_{t-1}, \rho_{t-1}, r_{t-1})$ stores opportunity strength, retention rate, and accumulated risk. A shared MLP integrates these inputs:

$$h_t = f_{\text{shared}}([\text{vec}(Z_t); v_t; s_{t-1}; e_t]), \tag{1}$$

where vec flattens the role features and semicolons denote concatenation. Each prediction head combines this shared context with market features and the role vectors relevant to its task. Let $u_{ij} = [h_t; z_{t,i}; z_{t,j}; v_t]$, and

$$\begin{aligned} &g_t = \sigma(f_g(u_{12})), \quad \hat{y}_{t,1:10} = R_{\max} \tanh(f_y(u_{23})), \quad p_t^- = \sigma(f_r(u_{42})), \\ &L_t = L_{\min} + (L_{\max} - L_{\min})\sigma(f_L(u_{32})), \quad \ell_t = \sigma(f_\ell(u_{34})). \end{aligned} \tag{2}$$

Here, σ is the sigmoid function, $R_{\max}$ bounds return predictions, and $[L_{\min}, L_{\max}]$ bounds the opportunity half-life in trading days.

These outputs serve complementary purposes. Write strength g_t controls new opportunity admission, while L_t determines its daily retention factor $2^{-1/L_t}$. The return path $\hat{y}_{t,h}$ estimates cumulative log returns from the next trading day's open to the close of the hth subsequent trading day, adjusted using the market benchmark and matched non-policy dates. These matched dates define supervision targets; the base strategies serve as references for evaluating trading performance. The risk score p_t^- is supervised using downside magnitudes at horizons $\{1, 5, 10\}$. The daily release proportion ℓ_t is learned through sequential trading optimization and controls how much existing risk is removed before new risk is added. Target construction is detailed in the training section. Together, $V_t = (\hat{y}_{t,1:10}, g_t, L_t, p_t^-, \ell_t)$ guides recurrent state updates.

On days without new policies, we set $P_t = J_t = 0$ and $e_t = 0$, while retaining market and state inputs. Opportunity writing uses $\alpha_t = e_t g_t$, and new risk writes are also masked by e_t. Thus, new writing is disabled between announcements, while existing opportunities and risks continue to decay and the release head can still reduce accumulated risk.

4.2 Recurrent Value Evolution

Policy information remains useful between announcements, while opportunities and risks may evolve at different rates. We therefore maintain separate opportunity and risk states, allowing accumulated risk to trigger a cash exit without discarding retained opportunity information. As shown in Fig. 2(2), both states are updated daily. We first extract positive and downside magnitudes from the predicted return path at horizons $\mathcal{K} = \{1, 5, 10\}$.

$$v_t^+ = \max\left(0, \max_{h \in \mathcal{K}} \hat{y}_{t,h}\right), \qquad v_t^- = \max\left(0, -\min_{h \in \mathcal{K}} \hat{y}_{t,h}\right). \tag{3}$$

Opportunity decay and blending. An admitted policy updates both the strength of retained opportunities and how long they persist. We obtain a new opportunity value $c_t^+ \in [0, 1]$ by shifting, scaling, and clipping v_t^+. With admission weight $\alpha_t = e_t g_t$, we first decay the previous opportunity strength, then blend in the new value and update its retention rate.

$$m_t = (1 - \alpha_t)\rho_{t-1} m_{t-1} + \alpha_t c_t^+, \qquad \rho_t = (1 - \alpha_t)\rho_{t-1} + \alpha_t 2^{-1/L_t}. \tag{4}$$

The same admission weight determines how strongly the new policy changes opportunity strength and retention. The current update uses ρ_{t-1}, while the updated rate ρ_t governs decay from the next day onward. Without a new policy, $\alpha_t = 0$, so opportunity strength decays and the retention rate remains unchanged.

Risk decay, release, and writing. Risk should be reassessed between announcements, while new downside signals may add to existing concerns. We therefore combine gradual decay with learned daily release and conditional risk writing. When $e_t = 1$ and $p_t^- \geq \theta_w$, shifting, scaling, and clipping v_t^- to $[0, 1]$ produces new risk q_t^-; otherwise, $q_t^- = 0$. Using a learnable global risk half-life L^-, we decay existing risk, release the predicted fraction ℓ_t, and add new risk.

$$r_t = \mathrm{clip}\left(2^{-1/L^-}(1-\ell_t)r_{t-1} + q_t^-, 0, 2\right). \tag{5}$$

Thus, new risk accumulates on top of retained risk. Release acts on existing risk before new risk is added and remains active even on days without policy announcements.

Exposure and state feedback. For a single PolicyState model, the opportunity state determines additional exposure o_t over the base position $b_t \in [0, 1]$. The risk state takes priority when it reaches the cash-exit threshold θ_c.

$$o_t = C\tanh(m_t/S), \qquad w_t = (1 - \mathbf{1}\{r_t \geq \theta_c\})\,\mathrm{clip}(b_t + o_t, 0, 1). \tag{6}$$

Here, C caps additional exposure, and the learnable scale S controls saturation. Targets are set after the day-t close, with orders eligible for execution at the next trading day's open. A cash exit preserves the evolving opportunity state. The full state s_t feeds into next-day judgment fusion and the shared MLP, enabling policy assessment alongside retained opportunities and risks. For ensemble inference, we average opportunity overlays from three independently trained PolicyState models and trigger a cash exit when at least two reach the risk-exit threshold. Otherwise, the averaged overlay is added to the base position and clipped to $0, 1$; each model maintains its own recurrent state.

4.3 Training Strategy

We train PolicyState in two stages, keeping the Qwen2.5-7B-Instruct encoder (Yang et al., 2024) frozen throughout. We train our method on 2021-2023 and select hyperparameters and random seeds using 2024 validation returns. The final performance is evaluated on 2025.

Stage I: Policy-value supervision. We initialize Gate through supervised learning. First, we match each policy-event day to ten similar days without new policies from the same ETF and year, using market features standardized with training-period statistics. Benchmark-adjusted event log returns minus matched medians form clipped 1-10-day target paths. Second, these paths supervise returns, while gains and downside at horizons $\{1, 5, 10\}$ provide soft writing and risk targets. Positive-increment timing provides a retention prior. Each event starts from the initial state; the four heads and their upstream modules are trained jointly.

Stage II: Sequential trading optimization. Starting from Stage I, we perform 40 training iterations. First, we simulate complete sector-year sequences, initializing states at the start and carrying them across days. Second, for each sector-year-base-strategy account, we compute its net cumulative log-return improvement over the base under identical execution rules and costs, normalized to 242 trading days. The objective rewards average improvement and gives extra weight to the quarter of accounts with the lowest relative gains. An anchor penalty averages mean squared deviations from fixed Stage I parameters across tensors. Each iteration performs one parameter update by backpropagating through trading computations and recurrent states, using straight-through estimators for risk-write and risk-exit thresholds and trade-quantity rounding.

Training configurations. Both stages use AdamW with constant learning rates and gradient clipping at norm 1.0. Stage I uses 40 epochs, batch size 32, learning rate 2×10^{-4}, and weight decay 0.05. Stage II uses learning rate 5×10^{-5} and weight decay 0.001 for network parameters; global risk half-life and exposure saturation parameters use learning rate 5×10^{-4} without weight decay. With frozen Qwen features precomputed and cached, training on a single 32 GB NVIDIA RTX 5090D takes approximately 12 seconds for Stage I and 106-112 seconds for Stage II per seed.

Table 3: Full-year 2025 out-of-sample results, macro-averaged over 20 sector-base accounts. Edge denotes return gain over the matched base; win rate denotes the fraction of accounts with positive edge. The full-input GRU uses the same inputs and training budget as PolicyState.

Method	Return (%) ↑	Sharpe ↑	Sortino ↑	MaxDD (%) ↓	Turnover (×) ↓	Win rate (%) ↑	Edge (pp) ↑
Matched bases	6.76	0.439	0.664	11.44	16.66	-	0.00
Text-based Models							
Dictionary	10.39	0.643	0.916	13.16	13.48	80	3.63
Chinese financial PLM	10.55	0.654	0.942	13.16	15.77	75	3.79
Text-to-return	-1.15	0.047	0.100	13.18	30.38	20	-7.91
LLM							
Deepseekv4 flash	9.54	0.607	0.890	13.24	19.67	65	2.78
Qwen2.5-7B	7.89	0.517	0.746	12.00	11.48	45	1.13
GPT-5	7.20	0.537	0.778	10.55	6.97	65	0.45
Claude Sonnet 5	4.65	0.460	0.678	7.87	8.45	40	-2.10
Gemini 3.8 Flash	4.40	0.401	0.596	8.43	11.96	30	-2.36
Financial Agents							
FinMem	8.25	0.535	0.783	13.06	21.10	65	1.49
FinAgent	5.50	0.371	0.560	12.32	18.15	15	-1.26
Full-input GRU	11.03	0.628	0.952	11.82	52.43	60 (12/20)	4.27
PolicyState	**16.25**	**1.472**	**1.579**	**6.38**	**35.59**	**95**	**9.49**

5 Experimental Results

5.1 Experimental Setup

Evaluation metrics. We evaluate five sector ETFs (automobile, computer, healthcare, securities, and transportation) with four base strategies (B&H, MR, TREND, and DEFENSIVE), yielding 20 sector-base accounts. We report cumulative net return, annualized Sharpe ratio, Sortino ratio, maximum drawdown (MaxDD), turnover, return edge, and account-level win rate. Return edge is the gain in cumulative net return over the matched base, measured in percentage points (pp); win rate is the share of accounts with a positive edge. MaxDD is reported as a positive loss magnitude.

Compared methods. Baselines include (i) text-based models, *s.t.*, dictionary sentiment (Loughran & McDonald, 2011), a Chinese financial PLM (Zhang et al., 2021), and a five-day text-to-return model; (ii) LLM baselines, *s.t.*, GPT-5, Qwen2.5-7B-Instruct, DeepSeek-V4-Flash, Claude Sonnet 5, and Gemini 3.8 Flash; and (iii) financial agents, *s.t.*, FinMem (Yu et al., 2025) and FinAgent (Zhang et al., 2024), retaining native memory and price or technical context. All methods share 20 PolicyValue-Bench accounts and next-open execution for 2025 testing, with 2021-2023 fitting and 2024 validation where applicable. LLMs predict exposures in $[0, 1]$ from identical policy, market, base-position, and ten-record history inputs.

5.2 Comparison with the State-of-the-art Methods

To evaluate whether policy-value modeling improves trading performance, we compare PolicyState with the baselines on 20 sector-base accounts in the 2025 out-of-sample period under the shared execution protocol. Table 3 reports macro-averaged performance. Key findings include: (i) PolicyState achieves the highest average net cumulative return of 16.25%, improving on the matched bases and the full-input GRU by 9.49 and 5.22 pp, respectively; (ii) PolicyState achieves the highest Sharpe and Sortino ratios (1.472 and 1.579 vs. 0.439 and 0.664 for the matched bases) and the lowest MaxDD (6.38% vs. 11.44%); (iii) these gains are net of transaction costs despite turnover of 35.59×, compared with 16.66× for the matched bases and 52.43× for the full-input GRU.

To assess the consistency of these gains across sectors and base strategies, we examine the return edge of PolicyState over each matched base. Table 4 presents results for all five sector ETFs and four base strategies. Key findings include: (i) PolicyState improves net cumulative returns in 19 of 20 accounts, yielding a 95% win rate, with positive average gains for all four base strategies; (ii) the computer ETF shows strong gains across all four strategies, ranging from 17.9 to 25.8 pp; (iii) healthcare with MR is the only combination with a negative return edge (-5.7 pp).

Table 4: 2025 return edge of PolicyState across sectors and base strategies (pp).

Base	Automobile	Computer	Healthcare	Securities	Transportation	Mean
B&H	3.9	25.8	0.7	5.5	9.6	9.1
MR	13.5	17.9	-5.7	1.1	4.0	6.2
TREND	7.0	22.8	7.3	13.9	10.4	12.3
DEFENSIVE	15.6	21.8	2.9	5.5	6.3	10.4

Table 5: Component and training-stage ablations. Variants train independently for 40 epochs per stage; win rate is the share outperforming matched bases.

Intervention	Return (%)	Sharpe	Sortino	MaxDD (%)	Turnover (×)	Win rate (%)	Edge (pp)
Full	**16.25**	**1.472**	**1.579**	6.38	35.59	**95**	**9.49**
Component ablations							
Policy semantic inputs zeroed	7.05	0.696	0.746	8.50	18.61	55	0.29
Judgment fusion disabled	12.61	1.105	1.155	7.53	32.11	70	5.85
Market inputs zeroed	10.67	0.911	0.906	9.30	33.80	65	3.91
Opportunity-state reset	9.45	1.057	1.010	**6.09**	37.59	65	2.69
Risk-state reset	12.83	1.151	1.126	6.82	46.18	65	6.07
Risk release disabled	12.87	1.108	1.187	8.26	33.39	75	6.11
Risk control disabled	9.44	0.790	0.859	10.38	17.84	60	2.68
Training-stage ablations							
Stage I only	10.82	0.865	0.924	10.77	21.33	75	4.06
Stage II only	7.83	0.655	0.699	10.53	**16.83**	55	1.07

5.3 Ablation Study

Current policy-value estimation. To assess the contribution of each information source, we separately zero policy semantic inputs, disable LLM judgment fusion, and zero market inputs. Component variants are retrained under identical training and execution settings. Table 5 shows that: (i) removing policy semantics causes the largest return drop among these variants, from 16.25% to 7.05%, leaving only a 4.3% relative return improvement over matched bases; (ii) disabling judgment fusion reduces return to 12.61%, while removing market inputs lowers it to 10.67% and increases MaxDD from 6.38% to 9.30%; (iii) win rates decline from 95% to 55%, 70%, and 65% when semantic inputs, judgment fusion, and market inputs are removed, respectively.

Recurrent value evolution. To assess how policy effects evolve across days, we separately reset the opportunity and risk states. We also disable risk release and risk control to examine how accumulated risk is updated and used for cash exits. Table 5 shows that: (i) resetting the opportunity state reduces return to 9.45%, despite slightly lowering MaxDD to 6.09%, suggesting that opportunity retention helps capture subsequent gains; (ii) resetting the risk state lowers return to 12.83% and raises MaxDD to 6.82%, while disabling risk release yields a MaxDD of 8.26%, supporting the retention and updating of accumulated risk; (iii) removing risk control reduces return to 9.44% and increases MaxDD to 10.38%, highlighting its contribution to limiting downside.

Two-stage optimization. To examine the training procedure, we compare the full model with Stage I only and Stage II only. Stage I only uses the supervised model; Stage II only optimizes trading returns from random initialization without the Stage I anchor. Single-stage variants receive 40 epochs, while the full model receives 40 epochs per stage. Table 5 shows that: (i) Stage I only and Stage II only achieve returns of 10.82% and 7.83%, trailing the full model by 33.4% and 51.8% in relative terms; (ii) both have higher MaxDD (10.77% and 10.53% vs. 6.38%); (iii) their win rates fall to 75% and 55%, compared with 95% for the full model. Hence, combining supervised initialization with sequential trading optimization improves returns, drawdowns, and account-level consistency.

6 Conclusion

We present PolicyValue-Bench and PolicyState to evaluate and learn the added trading value of government policies over matched policy-independent strategies. PolicyState selectively integrates policy and market information through separate opportunity and risk states for policy admission, persistence, and risk release. Held-out 2025 results show substantial net return gains across sectors and strategies. Future work will assess broader markets and longer periods.

AI USE STATEMENT

Generative AI tools were used to support the preparation of this manuscript, primarily for improving writing clarity, maintaining consistency in mathematical notation and equation formatting, and assisting with code testing and evaluation. The research ideas, model design, experimental methodology, and scientific conclusions were developed by the authors. All AI-assisted content was carefully reviewed and revised by the authors, and all mathematical expressions, implementation details, and experimental results were independently checked and verified. The authors take full responsibility for the content, accuracy, and integrity of the final manuscript.

ETHICS STATEMENT

This work studies policy-informed trading using publicly available government policy documents and historical ETF market data. It is intended for research evaluation, not investment advice or a ready-to-deploy trading system.

REPRODUCIBILITY STATEMENT

We provide the source code in the supplementary materials to facilitate reproducibility.

REFERENCES

Ben S Bernanke and Kenneth N Kuttner. What explains the stock market's reaction to federal reserve policy? *The Journal of finance*, 60(3):1221–1257, 2005.

Jacob Boudoukh, Ronen Feldman, Shimon Kogan, and Matthew Richardson. Which news moves stock prices? a textual analysis. Technical report, National Bureau of Economic Research, 2013.

Jidi Cao, Yile Wang, and Haiyue Liu. Stock market reaction to new energy vehicle industry development plan: an event study in china. *Climate Policy*, 24(7):934–948, 2024.

Liyuan Chen, Shilong Li, Jiangpeng Yan, Shuoling Liu, Qiang Yang, and Xiu Li. Cn-buzz2portfolio: A chinese-market dataset and benchmark for llm-based macro and sector asset allocation from daily trending financial news. *arXiv preprint arXiv:2603.22305*, 2026.

Yanxu Chen, Zijun Yao, Yantao Liu, Amy Xin, Jin Ye, Jianing Yu, Lei Hou, and Juanzi Li. Stockbench: Can llm agents trade stocks profitably in real-world markets? *arXiv preprint arXiv:2510.02209*, 2025.

Junyan Cheng and Peter Chin. Sociodojo: Building lifelong analytical agents with real-world text and time series. In *International Conference on Learning Representations*, volume 2024, pp. 22153–22187, 2024.

Meredith A Crowley, Ning Meng, and Huasheng Song. Policy shocks and stock market returns: Evidence from chinese solar panels. *Journal of the Japanese and International Economies*, 51: 148–169, 2019.

Sumin Kim, Minjae Kim, Jihoon Kwon, Yoon Kim, Oscar Levy, Alejandro Lopez-Lira, Yongjae Lee, and Chanyeol Choi. Llm as a risk manager: Llm semantic filtering for lead–lag trading in prediction markets. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 979–989, 2026.

Kelvin Koa, Jan Chen, Yunshan Ma, Zheng Huanhuan, and Tat-Seng Chua. Reasoning on time-series for financial technical analysis. In *International Conference on Learning Representations*, volume 2026, pp. 64059–64079, 2026.

Kenneth N Kuttner. Monetary policy surprises and interest rates: Evidence from the fed funds futures market. *Journal of monetary economics*, 47(3):523–544, 2001.

Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, Kp Subbalakshmi, Jimin Huang, et al. Investorbench: A benchmark for financial decision-making tasks with llm-based agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2509–2525, 2025a.

Shuqi Li, Yuebo Sun, Yuxin Lin, Xin Gao, Shuo Shang, and Rui Yan. Causalstock: Deep end-to-end causal discovery for news-driven multi-stock movement prediction. *Advances in neural information processing systems*, 37:47432–47454, 2024.

Xiangyu Li, Yawen Zeng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. Quantagents: Towards multi-agent financial system via simulated trading. *arXiv preprint arXiv:2510.04643*, 2025b.

Qianqiao Liang, Mengying Zhu, Xiaolin Zheng, and Yan Wang. An adaptive news-driven method for cvar-sensitive online portfolio selection in non-stationary financial markets. In *IJCAI*, pp. 2708–2715, 2021.

Chang Liu, Linlin Liu, Dayong Zhang, and Jiasha Fu. How does the capital market respond to policy shocks? evidence from listed solar photovoltaic companies in china. *Energy policy*, 151:112054, 2021.

Alejandro Lopez-Lira and Yuehua Tang. Can ChatGPT forecast stock price movements? return predictability and large language models. *Journal of Financial Economics*, 184:104335, 2026. doi: 10.1016/j.jfineco.2026.104335.

Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1):35–65, 2011. doi: 10.1111/j.1540-6261.2010.01625.x.

Tianmi Ma, Jiawei Du, Wenxin Huang, Wenjie Wang, Liang Xie, Xian Zhong, and Joey Tianyi Zhou. Agent trading arena: A study on numerical understanding in LLM-based agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5496–5514, 2025. doi: 10.18653/v1/2025.findings-emnlp.294.

Jayanta Mandi, Vıctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In *International conference on machine learning*, pp. 14935–14947, 2022.

William L Megginson, Junjie Xia, et al. Industrial policy and asset prices: Evidence from the made in china 2025 policy. *Journal of banking & finance*, 142:106554, 2022.

Agam Shah, Suvan Paturi, and Sudheer Chava. Trillion dollar words: A new financial dataset, task & market analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6664–6679, 2023.

Zifan Song, Kaitao Song, Guosheng Hu, Ding Qi, Junyao Gao, Xiaohua Wang, Dongsheng Li, and Cai Zhao. Trade in minutes! rationality-driven agentic system for quantitative financial trading. In *International Conference on Learning Representations*, volume 2026, pp. 27140–27156, 2026.

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.

Paul C Tetlock. All the news that's fit to reprint: Do investors react to stale information? *The Review of Financial Studies*, 24(5):1481–1512, 2011.

Jinyuan Wang, Zhong Wang, Zeyang Zhu, Jinhao Xie, Yong Yu, Yongjian Fei, Yue Huang, Dawei Cheng, and Hai Zhao. Csprd: A financial policy retrieval dataset for chinese stock market. In *International Conference on Database and Expert Systems Applications*, pp. 3–17. Springer, 2024.

Weimin Xie, Hengxin Zhang, Jialu Guo, and Miao He. Does a national industrial policy promote financial market stability? a study based on stock price crash risk. *China Journal of Accounting Research*, 15(4):100269, 2022.

Guojun Xiong, Zhiyang Deng, Keyi Wang, Yupeng Cao, Haohang Li, Yangyang Yu, Xueqing Peng, Mingquan Lin, Kaleb E Smith, Xiao-Yang Liu, et al. Flag-trader: Fusion llm-agent with gradient-based reinforcement learning for financial trading. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13921–13934, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Mengyuan Yang, Mengying Zhu, Qianqiao Liang, Xiaolin Zheng, and MengHan Wang. Spotlight news driven quantitative trading based on trajectory optimization. In *IJCAI*, pp. 4930–4939, 2023.

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*, 2025.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining*, pp. 4314–4325, 2024.

Zhuosheng Zhang, Hanqing Zhang, Keming Chen, Yuhang Guo, Jingyun Hua, Yulong Wang, and Ming Zhou. Mengzi: Towards lightweight yet ingenious pre-trained models for chinese. *arXiv preprint arXiv:2110.06696*, 2021.

Arman Zharmagambetov, Brandon Amos, Aaron Ferber, Taoan Huang, Bistra Dilkina, and Yuandong Tian. Landscape surrogate: Learning decision losses for mathematical optimization under partial information. *Advances in Neural Information Processing Systems*, 36:27332–27350, 2023.

A Implementation Details

A.1 Market Inputs and Base Strategies

Assets and prices. The five sector ETFs are automobiles (516110.SH), computers (512720.SH), healthcare (512010.SH), securities (512880.SH), and transportation (159666.SZ). Daily open, close, and previous-close prices are used for execution. The saved price series places healthcare ETF prices before its June 25, 2021 1:4 split in post-split units. Market features use observations through the decision-day close. The CSI 300 ETF (510300.SH) supplies the benchmark for the relative-return input features.

Feature definitions. Let p_t denote the sector ETF close, $M_k(t)$ its trailing k-session simple moving average, and $u_t = p_t/p_{t-1} - 1$ its daily simple return. Regime features are computed from the available close-price history in the 200-calendar-day interval ending at t. Write $\mathcal{H}_t$ for this history and $\sigma_{20}(t)$ for the population standard deviation of the last 20 daily returns. The ordered input vector contains the following 14 features. The two drawdown entries are retained as distinct input columns to reproduce the saved feature interface.

Table 6: Definitions of the 14 market input features before normalization.

No.	Input field	Definition before normalization
1–2	Sector returns, 7 and 20 days	$p_t/p_{t-k} - 1$, for $k \in \{7, 20\}$.
3–4	Relative returns, 7 and 20 days	Sector return minus the corresponding CSI 300 ETF return.
5	20-day volatility	$\sigma_{20}(t)$; daily, non-annualized population standard deviation.
6	Drawdown	$p_t/\max_{d\in\mathcal{H}_t} p_d - 1$.
7–8	Above MA20 and MA15	$\mathbf{1}\{p_t \geq M_k(t)\}$, for $k \in \{20, 15\}$.
9	Regime MA20 gap	$p_t/M_{20}(t) - 1$.
10	Regime MA20 slope	$M_{20}(t)/M_{20}(t-20) - 1$.
11	Relative volatility regime	Current 20-day volatility divided by the median of the rolling 20-day volatilities computed within $\mathcal{H}_t$; the ratio defaults to 1 when its denominator is zero.
12	Regime drawdown	The drawdown in row 6, retained in the regime-feature block.
13	MA20 crossings	Number of sign changes in $p_d - M_{20}(d)$ over the last 40 observations with sufficient moving-average history; equality is assigned the nonnegative sign.
14	Regime availability	Indicator that at least 60 close observations are available to compute regime features.

Boolean features are converted to 0 or 1. Missing, nonnumeric, or nonfinite feature values are replaced by zero before normalization. Each coordinate is standardized as $(x_j - \mu_j)/\max(s_j, 10^{-6})$, where μ_j and s_j are the mean and population standard deviation over the 3,045 saved sector-day training observations from 2021–2023. Each sector-day contributes once, independently of the four base accounts. These statistics remain fixed for evaluation; the exact vectors and field order accompany this supplement in `implementation_config.json`.

Base strategies. All bases use only the available price history. Buy-and-hold sets $b_t = 1$. TREND sets $b_t = \mathbf{1}\{p_t \geq M_{20}(t)\}$, and DEFENSIVE sets $b_t = 0.5\mathbf{1}\{p_t \geq M_{60}(t)\}$. Mean reversion uses

$$z_t^{\mathrm{MR}} = \frac{p_t - M_{20}(t)}{s_{20}^{\mathrm{price}}(t)}, \qquad b_t = \mathrm{clip}\big(0.5 - 0.4 z_t^{\mathrm{MR}}, 0, 1\big), \tag{A.1}$$

where $s_{20}^{\mathrm{price}}(t)$ is the population standard deviation of the last 20 closing prices. Its target is rounded to four decimal places; zero price standard deviation gives $b_t = 0.5$. The strategy implementation retains its native feature rounding, including six-decimal close and MA values in the mean-reversion calculation.

A.2 Policy Context, Judgments, and Frozen Representations

Information supplied to the language model. An event-day input contains the decision date, target sector and ETF, the titles, publication dates, issuing organizations, and saved summaries and measures of current policies, and up to five earlier policies from the same sector. History documents have publication dates strictly earlier than the decision date; current documents have dates no later than it. Current documents are excluded from the historical list, and document IDs are deduplicated before selecting the latest five history items. Thus, the text encoder receives the stored summaries and measures, while the original documents remain part of the benchmark corpus.

The daily input cache records publication dates at calendar-date resolution. Policies released on non-trading dates enter the next trading-day decision stream. The cache does not contain an intraday publication-time field; same-date availability is therefore a date-level convention. Market inputs extend through the decision-day close, and the resulting order is eligible for the next trading-day open.

Judgment generation. We use a frozen local Qwen2.5-7B-Instruct model to generate six fields: direction, affected object, transmission mechanism, novelty relative to the provided history, implementation horizon, and policy-grounded risks. The user message contains the policy payload described above. It contains no subsequent prices, realized returns, or trading-performance labels. Generation uses greedy decoding (sampling disabled), a maximum of 384 new tokens, repetition penalty 1.05, and a 4,096-token context budget. The resumed generation batch size is 8. Each output must terminate with EOS and contain all six field labels; invalid outputs fail validation. The system prompt requires judgments to use only the supplied documents and to avoid recalling later events or market outcomes. This controls the provided context; the historical coverage of the pretrained model's knowledge is not established by these input checks. The original Chinese system prompt and encoder templates are supplied verbatim in `prompts_zh.json`; Section I gives an English translation for reading.

A.3 Policy Reading and Value Prediction

Each encoding prompt ends with five textual slot markers, ordered as direction, novelty, value, horizon, and risk. The prompt body and marker suffix are tokenized separately and concatenated. For each slot, the extraction position is the final token of the cumulative suffix ending at that marker. A frozen Qwen forward pass supplies the final-layer output hidden state at each of these five positions. The resulting policy and assessment representations $P_t, J_t \in \mathbb{R}^{5\times 3584}$ are separately projected to $\bar{P}_t, \bar{J}_t \in \mathbb{R}^{5\times d}$, with $d = 32$. The market encoder maps 14 features to a 16-dimensional embedding.

Policy and judgment prompts are encoded separately with a 4,096-token limit and no truncation. BF16 encoder computation produces FP16 cached features. The stored 5×7168 tensor concatenates the two branches along the hidden axis and is split before projection. On non-event days, both branches are zero and $e_t = 0$.

Four queries form $Q \in \mathbb{R}^{4\times d}$, corresponding to writing, returns, persistence, and risk:

$$\begin{aligned} U_t &= \mathrm{LN}\big(Q + \mathrm{MHA}(Q, \bar{P}_t, \bar{P}_t)\big), \\ V_t &= \mathrm{LN}\big(\mathrm{MHA}(U_t, \bar{J}_t, \bar{J}_t)\big). \end{aligned} \tag{A.2}$$

Here, MHA denotes multi-head attention and LN denotes layer normalization. Let $u_{t,k}$ and $v_{t,k}$ denote the kth rows of U_t and V_t. Fusion is computed as

$$\begin{aligned} a_{t,k} &= e_t \eta_t \, \sigma\big(f_a([u_{t,k}; v_{t,k}; |u_{t,k} - v_{t,k}|; s_{t-1}])\big), \\ z_{t,k} &= u_{t,k} + a_{t,k} \tanh\big(W_\Delta [u_{t,k}; v_{t,k}] + b_\Delta\big). \end{aligned} \tag{A.3}$$

Here, $e_t \in \{0, 1\}$ indicates whether a new policy arrives that day, $\eta_t \in \{0, 1\}$ indicates whether an LLM judgment is available, and σ is the sigmoid function. The network f_a concatenates the two feature vectors, their elementwise absolute difference, and the previous day's state to compute the fusion weight; $\tanh(W_\Delta [u_{t,k}; v_{t,k}] + b_\Delta)$ is the supplementary information. W_Δ and b_Δ are linear-layer parameters initialized to zero. Stacking the four fused features as rows gives Z_t.

Both attention modules use four heads of width 8 with scaled dot-product attention and learned query, key, value, and output projections. Each input branch applies LayerNorm, a $3584 \to 32$ linear

projection, GELU, and LayerNorm. Policy-attention dropout is 0.2 during Stage I; judgment-attention dropout is zero. The adoption network f_a uses widths $99 \to 32 \to 1$, with GELU between the linear layers. Its output weights start at zero and its bias at -2.

The market encoder f_x maps 14-dimensional features to 16 dimensions, with output v_t^x. The shared representation in the main text is computed as

$$h_t = f_{\text{shared}}([\text{vec}(Z_t); v_t^x; s_{t-1}; e_t]). \tag{A.4}$$

Here, vec flattens the four policy features into a vector, and semicolons denote vector concatenation. Each prediction head uses h_t, v_t^x, and two relevant policy features: writing uses writing and return features; returns and half-life use return and persistence features; risk uses risk and return features; and risk release uses persistence and risk features. Denoting their raw outputs by ξ, we have

$$\begin{aligned} g_t &= \sigma(\xi_t^g), & \hat{y}_{t,h} &= 0.20 \tanh(\xi_{t,h}^y), \quad h = 1, \dots, 10, \\ L_t &= 0.5 + 9.5\sigma(\xi_t^L), & \rho_t^{\text{new}} &= 2^{-1/L_t}, \\ p_t^- &= \sigma(\xi_t^-), & \ell_t &= \sigma(\xi_t^\ell). \end{aligned} \tag{A.5}$$

The market encoder uses a $14 \to 16$ linear layer, GELU, and LayerNorm. The shared MLP has widths $148 \to 64 \to 64$, with GELU after each linear layer and dropout 0.2 followed by LayerNorm between them. Each output head uses a $144 \to 32$ linear layer, GELU, and a final linear layer. The return head has ten outputs and each other head has one. The ordered role pairs are writing–value, value–retention, retention–value, risk–value, and retention–risk for the write, return, half-life, risk, and release heads, respectively. The release output layer has zero initial weights and bias -3, giving initial release $\sigma(-3) \simeq 0.047426$.

Here, ρ_t^{new} is the daily retention rate of the new opportunity. The main text uses $R_{\max} = 0.20$, $L_{\min} = 0.5$, and $L_{\max} = 10$, bounding the return path to $[-0.20, 0.20]$ and the half-life to 0.5–10 trading days. With $\mathcal{K} = \{1, 5, 10\}$, opportunity and downside magnitudes are

$$v_t^+ = \max\left(0, \max_{h \in \mathcal{K}} \hat{y}_{t,h}\right), \qquad v_t^- = \max\left(0, -\min_{h \in \mathcal{K}} \hat{y}_{t,h}\right). \tag{A.6}$$

A.4 State Writing and Trading Execution

Opportunity magnitude is transformed into the new opportunity value, and the actual write weight is constrained by the new-policy indicator:

$$c_t^+ = \text{clip}\left(\frac{v_t^+ - 0.005}{0.08}, 0, 1\right), \qquad \alpha_t = e_t g_t. \tag{A.7}$$

These quantities give the opportunity-strength and daily-retention updates in the main text, with initial state $s_0 = (0, 2^{-1/5}, 0)$. Risk writing uses the downside magnitude and risk score:

$$d_t^- = \text{clip}\left(\frac{v_t^- - \delta^-}{0.06}, 0, 1\right), \qquad q_t^- = e_t \mathbf{1}\{p_t^- \geq \theta_w\} d_t^-. \tag{A.8}$$

Here, δ^- is the downside-magnitude offset, θ_w is the risk-write threshold, and $\mathbf{1}\{\cdot\}$ is the indicator function. The global risk half-life L^- in the main text is learnable.

The saturation scale and global risk half-life use learned log-parameters α_S and α_-:

$$S = \text{clip}(\exp(\alpha_S), 0.02, 2.0), \qquad L^- = \text{clip}(\exp(\alpha_-), 0.5, 30.0). \tag{A.9}$$

They are initialized to $S = 0.20$ and $L^- = 2.0$.

For a single Gate, additional exposure and target exposure are

$$o_t = C \tanh(m_t / S), \qquad w_t = (1 - \mathbf{1}\{r_t \geq \theta_c\}) \,\text{clip}(b_t + o_t, 0, 1), \tag{A.10}$$

where $C = 0.60$ is the cap on additional exposure, S is a learnable scaling factor, and θ_c is the exit threshold. Each of the three members maintains its own recurrent state. For member k, let $o_t^{(k)}$ be its

positive overlay and $r_t^{(k)}$ its risk state. The ensemble target is

$$\bar{o}_t = \frac{1}{3}\sum_{k=1}^{3} o_t^{(k)}, \tag{A.11}$$

$$G_t^- = \mathbf{1}\left\{\sum_{k=1}^{3} \mathbf{1}\{r_t^{(k)} \geq \theta_c\} \geq 2\right\}, \tag{A.12}$$

$$w_t^{(b)} = (1 - G_t^-)\,\mathrm{clip}\Big(b_t^{(b)} + \bar{o}_t, 0, 1\Big). \tag{A.13}$$

The same sector-level overlay and exit decision modulate each of the four base strategies. The evaluated ensemble uses seeds 43, 44, and 47. Each member's state is initialized at the start of the sector-year sequence and carried forward within it. For B&H, $b_t = 1$ and $o_t \geq 0$, so the positive target is capped at one and direct exposure changes occur through risk exits. Opportunity states can still affect subsequent risk predictions through state feedback.

A.5 Execution, Costs, and Account Metrics

Order formation and fills. Each account starts with RMB 1,000,000 in cash and zero shares. Positions are long-only and unlevered. After close t, a target is queued only if its absolute difference from the marked current exposure is at least 0.03. A queued order is considered at open $t+1$. The single-side commission is $\kappa = 0.00005$ (0.5 basis points) of executed notional. The ledger adds no separate slippage, bid–ask spread, market-impact, tax, or minimum-commission term, and imposes no volume participation cap. Eligible orders fill at the recorded open.

At an execution step, let c, n, p^- be the preceding cash, share count, and marked close, let O be the current open, and let $\bar{w}$ be the pending target. The order value uses the previous mark:

$$E^- = c + np^-, \qquad \Delta = \bar{w}E^- - np^-. \tag{A.14}$$

For $\Delta > 0$, the proposed purchase quantity is

$$q^+ = 100\left\lfloor \frac{\Delta(1 - 1.5\kappa)}{100O} \right\rfloor, \tag{A.15}$$

and the fill requires $q^+O(1+\kappa) \leq c + 10^{-6}$. For a sale, with $n_{\mathrm{free}} = \max(n - n_{\mathrm{locked}}, 0)$,

$$q^- = \begin{cases} n_{\mathrm{free}}, & \bar{w} \leq 0, \\ \min\{n_{\mathrm{free}}, \lfloor \max(-\Delta, 0)/O \rfloor\}, & \bar{w} > 0. \end{cases} \tag{A.16}$$

Purchases require 100-share lots; partial sales use integer shares. Purchases are blocked when $O/p_{\mathrm{prev.close}} - 1 \geq 0.099$, and sales when this ratio is at most -0.099. We enforce T+1 trading: shares purchased on a trading day cannot be sold until the next trading day. At each close, the next pending order is recomputed from the new target and marked exposure.

Cash, NAV, and the terminal step. For actual filled quantities q^+, q^-, accounting is

$$c' = c - q^+O(1+\kappa) + q^-O(1-\kappa), \qquad n' = n + q^+ - q^-, \qquad E_t = c' + n'p_t. \tag{A.17}$$

At the last close, unlocked shares are liquidated at that close with commission; the terminal liquidation bypasses the ordinary daily-limit checks. Locked shares remain marked at the final close. The last pending decision has no subsequent open and receives no fill. Terminal NAV therefore includes any marked residual holding. The same accounting is used for PolicyState and the paired base replay.

Metric definitions. For account a, let $E_{a,0} = 10^6$ and $r_{a,t} = E_{a,t}/E_{a,t-1} - 1$. Cumulative net return is $R_a = E_{a,T}/E_{a,0} - 1$. Sharpe is $\sqrt{252}\,\bar{r}_a / \mathrm{sd}_{\mathrm{sample}}(r_a)$; Sortino uses denominator $\sqrt{T^{-1}\sum_t \min(r_{a,t}, 0)^2}$ and the same numerator. Both use zero as the return benchmark. Maximum drawdown is $\max_t[1 - E_{a,t}/\max_{0\leq u\leq t} E_{a,u}]$. Annual two-sided turnover is

$$\mathrm{Turnover}_a = \sum_{t=1}^{T} \frac{\text{buy notional}_{a,t} + \text{sell notional}_{a,t}}{E_{a,t-1}}, \tag{A.18}$$

including entry trades and terminal liquidation. Account macro-averages are computed after each metric is calculated separately. Return edge is $100N^{-1}\sum_a (R_a - R_a^b)$ in percentage points, and win rate is $N^{-1}\sum_a \mathbf{1}\{R_a > R_a^b\}$. Portfolio metrics, when reported, are computed from $\sum_a E_{a,t}$ and identified separately.

A.6 Stage I Supervisory Targets and Losses

Matched control dates. Each training event is matched to ten non-event dates of the same ETF and calendar year using squared Euclidean distance over the 14 normalized market features. Candidate dates must have a complete ten-session outcome path. The preferred pool excludes dates preceded by an event in the last five valid decision steps. If fewer than ten candidates remain, all valid non-event dates form the fallback pool. Matching takes the ten nearest dates without a distance caliper. The same-year pool can contain dates later than the event; it constructs offline training labels.

Return paths and boundaries. For event or control date d, let d_1 be the next trading session and d_h the hth subsequent session. The frozen supervision price file uses the CSI 300 index (000300.SH) as its benchmark; this is distinct from the CSI 300 ETF used in the relative-return input features. Define

$$A_{d,h} = \operatorname{clip}\left(\log \frac{p_{d_h}}{O_{d_1}} - \log \frac{p^B_{d_h}}{O^B_{d_1}}, -0.2, 0.2\right), \tag{A.19}$$

$$y_{d,h} = \operatorname{clip}\left(A_{d,h} - \operatorname{median}_{c\in\mathcal{C}_d} A_{c,h}, -0.2, 0.2\right), \qquad h = 1, \ldots, 10. \tag{A.20}$$

Both event and control paths must remain within the calendar year of date d and finish no later than December 31, 2023. Incomplete paths and paths crossing either a calendar-year boundary or the training-period endpoint are excluded rather than shortened or zero-padded. The saved target set contains 796 events, and its latest event-label endpoint is December 29, 2023.

Write and risk targets are constructed from the maximum return and maximum downside magnitude over horizons $\mathcal{K} = \{1, 5, 10\}$:

$$\begin{aligned} \tilde{g}_t &= \sigma\left(\frac{\max_{h\in\mathcal{K}} y_{t,h} - 0.02}{0.01}\right), \\ \tilde{p}^-_t &= \sigma\left(\frac{\max(0, -\min_{h\in\mathcal{K}} y_{t,h}) - 0.03}{0.01}\right). \end{aligned} \tag{A.21}$$

Persistence uses the timing of positive return increments. Let $y_{t,0} = 0$. For events whose maximum target return within ten days is at least 0.01, we construct temporal distributions:

$$\begin{aligned} &\Delta^+_{t,h} = \max(0, y_{t,h} - y_{t,h-1}), \\ &\pi_{t,h} = \frac{\Delta^+_{t,h}}{\sum_{j=1}^{10} \Delta^+_{t,j}}, \qquad \hat{\pi}_{t,h} = \frac{2^{-(h-1)/L_t}}{\sum_{j=1}^{10} 2^{-(j-1)/L_t}}. \end{aligned} \tag{A.22}$$

Here, $\pi_{t,h}$ is the target distribution obtained by normalizing positive increments, and $\hat{\pi}_{t,h}$ is the predicted distribution corresponding to the half-life. The persistence loss is

$$\mathcal{L}_{\text{retention}} = \frac{1}{|\mathcal{T}_+|} \sum_{t\in\mathcal{T}_+} \frac{1}{2} \sum_{h=1}^{10} \left(\sqrt{\pi_{t,h}} - \sqrt{\hat{\pi}_{t,h}}\right)^2. \tag{A.23}$$

Here, $\mathcal{T}_+$ is the set of events satisfying the return condition. This term is zero when the set is empty. The complete Stage I objective is

$$\mathcal{L}_1 = 2\mathcal{L}_{\text{path}} + 0.5\mathcal{L}_{\text{write}} + \mathcal{L}_{\text{risk}} + \mathcal{L}_{\text{retention}} + 0.01\mathcal{L}_{\text{role}}. \tag{A.24}$$

$\mathcal{L}_{\text{path}}$ uses mean smooth-L1 loss with unit transition, with both predictions and targets divided by 0.05 before computation. The write and risk losses are mean binary cross-entropy against the corresponding soft targets. The role regularizer is $\mathcal{L}_{\text{role}} = \|\bar{Q}\bar{Q}^\top - I_4\|_F^2/16$, where $\bar{Q}$ contains the row-normalized queries, I_4 is the identity matrix, and $\|\cdot\|_F$ is the Frobenius norm.

Each event starts from the initial recurrent state. Stage I uses AdamW for 40 epochs, batch size 32, learning rate 2×10^{-4}, weight decay 0.05, and gradient-norm clipping at 1.0. Event order is shuffled each epoch. The frozen semantic caches are reused throughout training, and no semantic-contrast term is enabled.

A.7 Stage II Training and Execution Settings

Account a in the main text is jointly defined by sector, year, and base strategy. Let R_a^{Gate} and R_a^{base} be the cumulative returns of Gate and the base strategy, and let T_a be the number of trading days. We first floor cumulative returns at $\tilde{R} = \max(R, -0.95)$ and then convert the log-return difference to a common 242-trading-day scale:

$$E_a = \frac{242}{T_a}\left[\log(1+\tilde{R}_a^{\mathrm{Gate}}) - \log(1+\tilde{R}_a^{\mathrm{base}})\right]. \tag{A.25}$$

The set $\mathcal{W}$ contains the $\max(1, \lfloor A/4 \rfloor)$ accounts with the lowest E_a.

The Stage II objective is

$$\mathcal{L}_2 = -100\left(\frac{1}{A}\sum_{a=1}^{A} E_a + \frac{0.25}{|\mathcal{W}|}\sum_{a\in\mathcal{W}} E_a\right) + 1000\mathcal{L}_{\mathrm{anchor}}. \tag{A.26}$$

To compute $\mathcal{L}_{\mathrm{anchor}}$, we first take the mean squared difference from the Stage I parameters within each parameter tensor, then average over all parameter tensors. Stage II starts from the Stage I parameters and updates all Gate parameters; risk release is learned from this stage's trading objective. Threshold decisions and trading discretization use straight-through estimators to propagate approximate gradients. Qwen remains frozen in both stages.

Both stages are trained on 2021–2023 data for 40 epochs each. Training uses $\theta_w = 0.50$, $\delta^- = 0.01$, and $\theta_c = 0.50$; the ensemble execution used for the reported results sets these three values to 0.60, 0.005, and 0.65, respectively.

Each Stage II iteration performs one update after the full sequence rollout. Network parameters use AdamW with learning rate 5×10^{-5} and weight decay 0.001; the two global log-parameters use learning rate 5×10^{-4} and zero weight decay. Learning rates remain constant and gradient-norm clipping is 1.0. The rollout uses evaluation mode to disable dropout while retaining gradients; cash and share accounting use FP64.

Risk writing uses the straight-through expression $p + \mathrm{stopgrad}(\mathbf{1}\{p \geq \theta_w\} - p)$. Risk exit similarly uses $s + \mathrm{stopgrad}(\mathbf{1}\{r \geq 0.5\} - s)$ with $s = \sigma(8(r - 0.5))$. Quantity rounding uses $x + \mathrm{stopgrad}(\lfloor x \rfloor - x)$. These expressions retain hard forward decisions and supply surrogate derivatives for optimization.

A.8 Data and Code Availability

We plan to publicly release PolicyValue-Bench and the complete PolicyState implementation. The release will include policy documents and source links, processed policy events and histories, market-data alignment and split definitions, preprocessing and judgment-generation prompts, model and training configurations, checkpoints, and evaluation scripts. We will also provide the frozen feature caches, target-construction code, and account traces needed to reproduce the reported execution and metric calculations.

B Supplementary Evaluation Protocol and Controls

B.1 Common Evaluation Protocol

Unless otherwise stated, fixed-test results in this section use T+1 execution and the complete 2025 daily panel, with five ETFs and four base strategies. The expanding-window evaluation uses the test years and data partitions stated in that section. The starting capital, rounding, position-change threshold, terminal handling, and costs are those specified in Appendix A. We verify every reported default-cost account return against three implementations: a scalar account simulator, the tensor training broker, and an independent ledger aligned by calendar date; their maximum absolute discrepancy is below 10^{-7}. Return is an account macro-average. Sharpe and maximum drawdown are also averaged only after being computed for each account; separately identified portfolio statistics use the sum of equally funded account equities, without rebalancing capital between accounts.

The primary 2025 fixed-test results use the ensemble selected on 2024 validation returns. Each two-stage method receives 40 supervised epochs and 40 sequential epochs; final checkpoints are used without further selection. PolicyState inference uses $(\theta_w, \delta^-, \theta_c) = (0.60, 0.005, 0.65)$, and GRU controls use 0.65 as the probability threshold for cash exit. These distinct model outputs need not be calibrated identically. In the expanding-window evaluation, each fold selects the trained model or ensemble together with its cash-exit threshold by maximizing account-macro return in its stated validation year.

B.2 Input and Architecture Controls

The full-input GRU reuses the same frozen policy and judgment vectors, market encoder, fusion module, auxiliary supervision, and trading objective. A 16-dimensional GRU state, a learned context projection, and a state readout replace the explicit opportunity/risk recurrence. It has 296,081 trainable parameters, compared with 295,681 for PolicyState. The GRU updates on every trading day, adds an opportunity adjustment to the base exposure, and uses its risk-head probability for cash exit.

The *market-input GRU* uses this same network and training budget, but zeros both semantic branches and the event indicator during training and evaluation. It can act daily without a policy-triggered write. It retains the common policy-event-derived auxiliary training targets, so it is a control of the available decision inputs rather than a claim of policy-independent supervision. In contrast, *both policy and judgment semantics zeroed* sets both semantic branches to zero in PolicyState while retaining event arrival and its original state-update structure. Thus the two controls answer different questions.

For *past-policy replacement*, each policy-day embedding bundle is replaced with a randomly sampled bundle from a strictly earlier day of the same sector. Policy/history and judgment representations move together, while arrival dates, market inputs, base exposures, labels, and training budgets remain unchanged. Sampling is with replacement, using one frozen map per model seed (random seed $100000 + s$); events without an earlier donor receive zeros. All donor dates are verified. This tests correct event–content correspondence without inserting future documents into earlier decisions. It replaces the whole cached text context, not only the current document, and changes the age distribution of that context; it is not a pure permutation of current-policy wording.

Two untrained market filters provide simple risk-control comparisons. The trend filter sets $w_t = b_t \mathbf{1}\{p_t \geq M_{20}(t)\}$. The defensive cap sets $w_t = \min(b_t, 0.5\mathbf{1}\{p_t \geq M_{60}(t)\})$. They use the same execution broker and require no policy input or search.

Table 7: Controlled 2025 comparisons. Return and drawdown are percentages; edge is in percentage points; turnover is two-sided.

Method	Return	Edge	Sharpe	MaxDD	Turnover
Matched bases	6.76	0.00	0.439	11.44	16.66
PolicyState	16.25	9.49	1.472	6.38	35.59
Full-input GRU	11.03	4.27	0.628	11.82	52.43
Market-input GRU	7.08	0.32	0.501	11.04	17.93
Both policy and judgment semantics zeroed	6.41	-0.35	0.542	10.95	18.02
Past-policy replacement	9.07	2.31	0.558	11.92	23.82
Trend filter	3.27	-3.49	0.293	8.44	22.30
Defensive cap	1.04	-5.72	0.089	6.11	10.23

Table 8: Individual model returns and the fixed three-model ensemble (percent). The mean and sample standard deviation describe separately executed single models, not an average of trading signals.

Method	43	44	47	Mean $\pm$ SD	Ensemble
PolicyState	12.64	12.09	13.08	12.60$\pm$0.49	16.25
Full-input GRU	10.15	15.04	9.40	11.53$\pm$3.06	11.03
Market-input GRU	8.16	8.54	7.71	8.14$\pm$0.41	7.08
Both policy and judgment semantics zeroed	8.45	9.01	7.66	8.37$\pm$0.68	6.41
Past-policy replacement	9.49	3.92	10.57	8.00$\pm$3.57	9.07

The comparisons above distinguish semantic inputs, event timing, and daily learned market control. They do not by themselves identify a causal policy effect. The GRU is a matched structural control; comparisons with text classifiers or agent adaptations additionally depend on those methods' information sets and action adapters.

B.3 Baseline Inputs and Decision Adapters

Dictionary sentiment uses a fixed lexicon and is not a trained predictor. The archived financial PLM freezes Mengzi and fits a direction head on records from 2021–2023. The retrained text-to-return model uses only frozen policy-text embeddings and 879 training records from 2021–2023; it predicts five-session ETF excess return using Ridge regression. Its regularization is chosen from $\{0.01, 0.1, 1, 10, 100\}$ by expanding validation within the training period (selected value 100), with end-of- partition labels purged. These procedures differ from two-stage PolicyState training and should not be described as an identical optimization or tuning budget.

For the discrete text/agent comparisons, the recorded adapter has $H = 10$ sessions and $\rho = 0.6$. A buy sets a persistent positive state, a sell sets a negative state, and an explicit hold resets it; without a new signal, the state persists until its ten-session expiry. Targets are $b_t + \rho(1 - b_t)$, $(1 - \rho)b_t$, or b_t, respectively. The negative adapter reduces a positive base exposure rather than guaranteeing a full cash exit. Thus sharing execution rules does not make its action space equivalent to PolicyState's learned overlay and binary cash exit.

The FinMem adaptation retains layered memory, retrieval, reflection, and LLM decisions with policy and historical close inputs. The FinAgent adaptation retains market-intelligence memory, retrieval, multi-horizon reflection, and close-derived technical context. Agent memory is initialized using only 2021–2023 training-period information; 2024 is used for validation. Action-outcome reflections become available only after their 20-session windows mature. These are input-restricted adaptations, not exact reproductions of the original multimodal systems, and neither requires LLM fine-tuning. The GPT-5, Qwen2.5-7B-Instruct, DeepSeek-V4-Flash, Claude Sonnet 5, and Gemini 3.8 Flash account-history baselines receive the same policy summaries, current market inputs, base position, and previous ten records of their own account's targets, realized exposures, and reasons. They directly output a target in $[0, 1]$ and may exit to cash; they do not consume PolicyState's

hidden states or its separately generated judgment channel. The account-history system prompts, input schema, and recorded fitting/adapter settings are supplied in `baseline_prompts.json` and `baseline_protocol.json`. Equal-input and equal-new-search-budget architectural conclusions are based on the GRU controls above, rather than assuming all these heterogeneous methods have identical capabilities.

C Selected Ensemble and Statistical Uncertainty

The reported PolicyState result uses the three independently trained models with random seeds 43, 44, and 47. Hyperparameters and the ensemble members are selected using 2024 validation returns, and the selected ensemble is then evaluated once on the held-out 2025 period. Under the primary protocol, the ensemble achieves an account-macro average net cumulative return of 16.25%, compared with 6.76% for the matched bases, for an improvement of 9.49 percentage points. It outperforms the matched base in 19 of 20 accounts.

The 20 accounts share prices, events, and market shocks. We therefore use a paired moving-block bootstrap that resamples synchronized five-trading-day blocks across all accounts and both methods. Within each draw, the resampled realized daily net returns are compounded before averaging account-level return differences. Trading is not re-executed on the resampled paths. The reported analysis uses 10,000 bootstrap draws and random seed 20260901. The two-sided p value uses null centering and the plus-one correction,

$$p = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\left\{ |\Delta_b^* - \overline{\Delta^*}| \geq |\widehat{\Delta}| \right\}}{B + 1}, \tag{C.1}$$

where $B = 10{,}000$, $\widehat{\Delta}$ is the observed return difference, and Δ_b^* is the difference in bootstrap draw b. For the selected ensemble, the 9.49-point improvement is statistically significant at the 5% level under a two-sided test ($p = 0.0382$), with a 95% percentile confidence interval of $[2.40, 20.10]$ percentage points. The interval is conditional on the selected fitted ensemble, assets, and observed test year.

Table 9: Paired five-trading-day block-bootstrap result for the primary 2025 comparison. The difference and confidence interval are in percentage points.

Comparison	Difference	95% CI	Two-sided p
PolicyState – matched bases	9.49	[2.40, 20.10]	0.0382

Table 10: Sensitivity of the paired moving-block bootstrap test to block length. The observed difference is 9.49 percentage points under both specifications.

Block length (trading days)	Difference (pp)	Two-sided p
5	9.49	0.0382
10	9.49	0.0441

D Expanding-Window Evaluation

For test years 2024 and 2025, training ends in 2022 and 2023, respectively; the intervening year is used only for validation. Each fold fits its normalizer and matched targets from its own training partition. Both folds require complete outcome paths to remain within the event's calendar year and exclude paths crossing the training-partition endpoint. For each fold, the trained model or ensemble and its cash-exit threshold are selected by maximizing account-macro return in the immediately preceding validation year: 2023 for the 2024 evaluation and 2024 for the 2025 evaluation. PolicyState and the GRU control each receive 40 epochs per stage. Their cash thresholds are independently chosen from $\{0.50, 0.65, 0.80\}$ by validation macro return, with ties resolved in that order, and saved

before evaluating that fold's test period. For the 2025 fold, the PolicyState checkpoint and evaluation threshold are the same as those used for the primary result in the main text.

Coverage follows the available price panels: training in 2021–2022 contains four sectors, and transportation enters the 2023 panel on February 13. For calendar-aligned portfolio calculations its account capital remains in cash before the first available observation. These are retrospective chronological tests on the chosen asset set; they do not establish that the same sectors would have been selected using only the historical training corpus.

Table 11: Expanding-window account-macro returns (percent). The last column gives validation-selected exit thresholds for PolicyState / GRU.

Test	Train	Valid.	Bases	PolicyState	GRU	Exit thresholds
2024	2021–2022	2023	9.65	17.28	11.76	0.80/0.80
2025	2021–2023	2024	6.76	16.25	11.03	0.65/0.65

PolicyState improves on the matched bases in both 2024 and 2025 and also outperforms the GRU control in both years. The positive results across two consecutive evaluation years provide evidence that the effectiveness of the PolicyState architecture is not driven by selecting a single favorable year.

E Training-Stage and Budget Analysis

The supervised-only control uses the same 40-epoch Stage I checkpoints as the full model. The Stage II-only control starts from random initialization and optimizes the sequential trading objective without the Stage I anchor. The primary comparisons use 40 epochs for each individual stage, matching the ablation results in the main text. We additionally report the previously recorded 80-epoch Stage II-only return to test whether doubling the number of full-sequence optimization passes closes the gap to the two-stage model. Supervised and sequential epochs have different computational costs, so 40+40 epochs and 80 sequential epochs are not claimed to have identical FLOPs.

Table 12: Training-stage and budget comparisons. Budgets are Stage I + Stage II epochs. The 40-epoch rows follow the primary protocol; for the 80-epoch Stage II-only checkpoint, edge is computed from the recorded return and the matched-base return, while MaxDD is unavailable.

Training	Epochs	Return (%)	Edge (pp)	MaxDD (%)
Two stages	40 + 40	**16.25**	**9.49**	**6.38**
Stage I only	40 + 0	10.82	4.06	10.77
Stage II only	0 + 40	7.83	1.07	10.53
Stage II only	0 + 80	10.25	3.49	–

The full two-stage procedure outperforms either 40-epoch single-stage variant. Extending Stage II-only training to 80 epochs raises its return to 10.25%, but it remains 6.00 percentage points below the two-stage model. This comparison indicates that the gain is not explained solely by the number of sequential optimization passes.

F Cost Sensitivity

We replay the selected PolicyState ensemble and the matched base strategies under higher effective one-way proportional costs, while keeping model outputs fixed. Share rounding, cash constraints, and order acceptance are recomputed at each cost level. Thus, the results reflect changes in executed trades rather than a constant turnover-based deduction.

Table 13: 2025 account-macro net returns under higher effective costs per side. Edge is the PolicyState return minus the matched-base return. Values are percentages or percentage points; edge is computed before display rounding.

Cost (bp)	PolicyState	Matched bases	Edge (pp)
0.5	**16.25**	6.76	**9.49**
2	15.62	6.50	9.12
5	14.39	5.98	8.41
10	**12.37**	5.12	**7.26**
20	8.46	3.42	5.04
30	4.71	1.78	2.93
50	-2.35	-1.39	-0.96
75	-10.38	-5.10	-5.28
100	-17.62	-8.57	-9.05

The selected ensemble retains a positive return edge through 30 bp per side. At 10 bp per side, it earns 12.37% and exceeds the matched bases by 7.26 percentage points. The edge becomes negative at the reported 50 bp grid point, indicating that the advantage does not persist under extreme transaction-cost assumptions.

G Account Results and Sector Concentration

Table 14: Return edge of the selected PolicyState ensemble over each matched base in 2025 (percentage points). The ensemble improves 19 of 20 accounts.

Sector	Base	Edge
Automobiles	B&H	3.9
Automobiles	MR	13.5
Automobiles	TREND	7.0
Automobiles	DEFENSIVE	15.6
Computers	B&H	25.8
Computers	MR	17.9
Computers	TREND	22.8
Computers	DEFENSIVE	21.8
Healthcare	B&H	0.7
Healthcare	MR	-5.7
Healthcare	TREND	7.3
Healthcare	DEFENSIVE	2.9
Securities	B&H	5.5
Securities	MR	1.1
Securities	TREND	13.9
Securities	DEFENSIVE	5.5
Transportation	B&H	9.6
Transportation	MR	4.0
Transportation	TREND	10.4
Transportation	DEFENSIVE	6.3

Table 15: Mean edge over the remaining 16 accounts after omitting one sector from aggregation (pp). Models are not retrained; this is a concentration diagnostic, not an unseen-sector test.

Omitted sector	Remaining mean edge
Automobiles	9.36
Computers	6.34
Healthcare	11.54
Securities	10.24
Transportation	9.97

For the selected ensemble, the account-macro average net cumulative return is 16.25%, compared with 6.76% for the matched bases. Its account-macro Sharpe ratio, Sortino ratio, maximum drawdown, and two-sided turnover are 1.472, 1.579, 6.38%, and 35.59, respectively. The corresponding matched-base values are 0.439, 0.664, 11.44%, and 16.66.

H Data-Time Audit and Interpretation

We rechecked all 2,150 sector-level policy event days across 2021–2025. Current-document dates never exceed the decision date, all historical-document dates are strictly earlier, and the history contains at most five documents. All 796 supervised training paths end within the training partition; the latest endpoint is December 29, 2023. These checks validate the saved input convention at calendar-date resolution.

The input payloads do not establish a historical intraday publication cutoff, first-seen archival time, or version history for later revisions. Restricting the judgment prompt to supplied documents does not establish the pretrained model's historical knowledge boundary. The five-sector universe is selected by policy density using only the 2021–2023 training-period corpus, as specified in the main text. The reported scope is these Chinese sector ETFs, not a broad cross-market claim.

Two architectural properties also delimit interpretation. With a fully invested B&H base, opportunity additions are clipped, so exposure changes occur through risk exits; that account alone does not validate direct opportunity-driven leverage. Without a new policy, the model's risk recurrence can only decay, with market inputs modulating release. The daily-updating GRU controls provide an alternative, but the present results do not establish that policy-triggered risk creation is necessary. Risk and half-life remain learned internal variables; predictive calibration and a causal interpretation of policy-effect duration do not follow from trading returns alone.

On non-event days the raw policy and judgment vectors are zero. Learned queries, projection biases, and residual transformations can nevertheless produce nonzero role representations; event indicators suppress new policy writes. The writing head and the ten-output return-path head are separate networks, despite sharing upstream representations. Ensemble members are trained independently; the averaging and majority-exit rule is applied only at inference.

Code and a subset of the data are provided in the supplementary material.

I PROMPT TEMPLATES

The following is the English translation of the Chinese judgment prompt. The executable Chinese text and encoder templates accompany the submission.

Prompt Template for Policy Judgment

You are a policy analyst. Analyze the effect on the target sector using only the supplied current policies and historical policies strictly earlier than today. First verify whether the direct beneficiaries or restricted entities belong to the target sector, then judge direction. General support for a field does not imply support for the target sector. Mark indirect connections explicitly; use neutral or uncertain when evidence is missing. Judge today's new policies; use history only for comparison. Do not use events outside the material, subsequent market prices, stock performance, or remembered historical outcomes. Policy support does not imply rising stock prices.
Output exactly six labeled lines, totaling at most 250 Chinese characters: Direction: support, restriction, mixed, neutral, or uncertain for the target sector, stating direct or indirect. Object: the entities actually affected and their relation to the target sector. Mechanism: how concrete measures affect the sector, stating when evidence is absent. Novelty: substantive changes relative to the provided history, including repetition or continuation. Horizon: explicitly stated dates or implementation conditions; otherwise uncertain. Risk: restrictions or implementation uncertainty supported by the material; otherwise uncertain.
Do not output trading instructions, numerical returns, extra paragraphs, or merely a general favorable-sector conclusion. Failure to mention the target sector does not mean restricting it. Support for another field does not establish crowding out or harm to the target sector. Do not invent a transmission chain from general economic associations. Support or restriction must have specific evidence; otherwise use neutral or uncertain. Do not infer immediate effectiveness without an explicit date. State evidence-based risks or say they are unspecified instead of adding generic implementation risks.

Input and Judgment-Encoding Template

The user message is a JSON object containing the decision date, target ETF, target sector, current-policy summaries and measures, and strictly earlier same-sector summaries and measures. Each document includes publication date, issuing organization, title, and summary/measures. The judgment encoder subsequently receives the target sector and generated six-line judgment, followed by the same five semantic-slot markers used for policy encoding.