When Valid Signals Fail: Regime Boundaries Between LLM Features and RL Trading Policies

Zhengzhe Yang
Independent Researcher
zhengzhe.yang@outlook.com

Abstract

Can large language models (LLMs) generate continuous numerical features that improve reinforcement learning (RL) trading agents? We build a modular pipeline where a frozen LLM serves as a stateless feature extractor, transforming unstructured daily news and filings into a fixed-dimensional vector consumed by a downstream PPO agent. We introduce an automated prompt-optimization loop that treats the extraction prompt as a discrete hyperparameter and tunes it directly against the Information Coefficient—the Spearman rank correlation between predicted and realized returns—rather than text-classification objectives. The optimized prompt discovers genuinely predictive features (IC above 0.15 on held-out data). However, these valid intermediate representations do not automatically translate into downstream task performance: during a distribution shift caused by a macroeconomic shock, LLM-derived features add noise, and the augmented agent under-performs a price-only baseline. In a calmer test regime the agent recovers, yet macroeconomic state variables remain the most robust driver of policy improvement. Our findings highlight a gap between feature-level validity and policy-level robustness that parallels known challenges in transfer learning under distribution shift.

1 Introduction

Recent work increasingly applies large language models to financial decision-making, whether as end-to-end trading agents or as modular sentiment classifiers feeding downstream models. While these modular pipelines separate the language model from the trading algorithm, they frequently suffer from an *objective mismatch*: the LLM is optimized against standard text-classification objectives (e.g., cross-entropy on sentiment polarity or topic labels) rather than downstream financial utility. Consequently, it remains difficult to guarantee that the extracted narratives form a robust state representation for a continuous trading policy.

We address this gap by maintaining strict architectural separation while directly aligning the feature extraction process with a financial objective. The frozen LLM acts as a *stateless feature extractor*: given a bundle of news articles and SEC filings for a ticker on day d, it emits a fixed-length numerical vector (sentiment, impact, conflict flags, etc.). A separate PPO agent then consumes this vector alongside price data and macroeconomic indicators to make portfolio decisions. This design ensures that the intermediate representations are genuinely predictive, allowing us to evaluate the LLM's true contribution in isolation.

Our contributions are:

1. **Prompt-as-hyperparameter optimization.** We introduce a mutation–evaluation–selection loop that treats the LLM extraction prompt as a discrete hyperparameter and optimizes it against the Information Coefficient (IC)—the rank correlation between predicted and realized returns—rather than text-classification objectives like BLEU or accuracy. The winning prompt improves IC from -0.024 to $+0.104$ (Table 2).
2. **Feature-validity-to-policy-utility gap.** We show that valid intermediate LLM representations do not automatically translate into downstream RL performance. The gap is regime-dependent: under distribution shift caused by a macroeconomic shock, news-derived features add noise rather than signal.

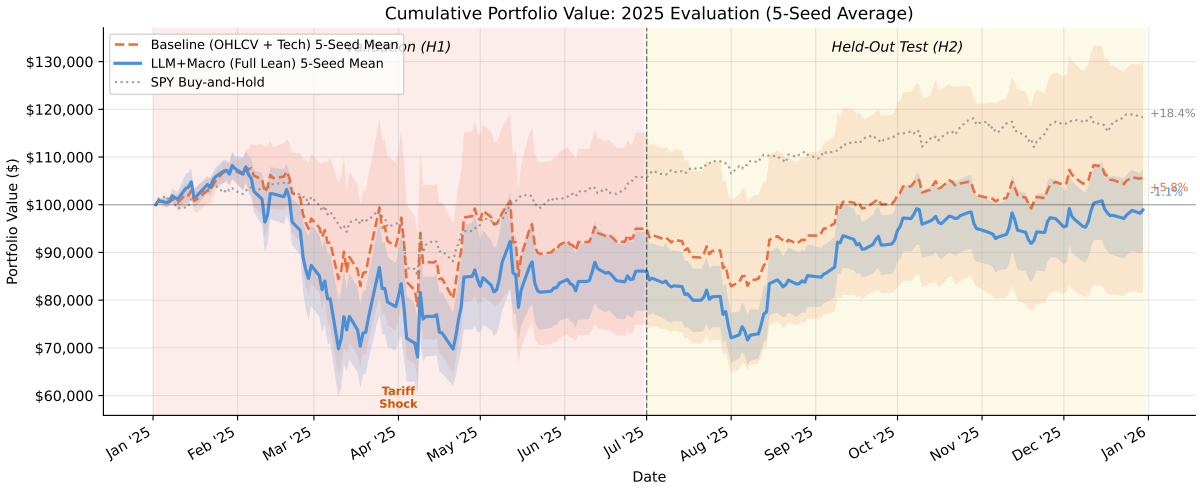

Figure 1: Cumulative portfolio value across 2025. Solid lines: 5-seed mean; shaded: ± 1 std. H1 (red) captures a tariff-driven shock; H2 (yellow) is calmer. The regime split isolates when LLM features succeed and fail. SPY buy-and-hold outperforms all RL configurations during the 2025 bull rally; the contribution is the *relative* comparison across configurations under controlled distribution shift, not absolute excess return.

3. **Multi-regime ablation on held-out data.** A controlled four-configuration ablation (Baseline, LLM-only, Macro-only, LLM+Macro) across a volatile validation period (H1 2025) and a calmer held-out test period (H2 2025) reveals that macroeconomic state variables are the most reliable driver of policy robustness (Figure 1).

2 Related Work

Reinforcement Learning for Trading. Deep RL has been applied to portfolio management and order execution with increasing sophistication (Hambly et al., 2023). FinRL (Liu et al., 2020) provides a standardized library for training PPO, A2C, and DDPG agents on market environments. A persistent challenge is non-stationarity: the data distribution shifts between training and deployment, a problem well-studied in the broader RL literature as distributional shift (Kumar et al., 2020). In finance, these shifts are driven by macroeconomic regime changes (Ang and Bekaert, 2002), and recent work has explored conditioning RL policies on detected regimes (Sun et al., 2023).

LLMs in Finance. BloombergGPT (Wu et al., 2023) demonstrated that domain-specific pretraining improves financial NLP benchmarks. FinGPT (Yang et al., 2023) pursued the same goal with open-source fine-tuning. Lopez-Lira and Tang (2023) demonstrated that frontier models can forecast subsequent-day stock returns using raw headline text, while later approaches instruction-tune open source variants directly against financial tasks (Zhang et al., 2023). However, these lines of work overwhelmingly evaluate the LLM as a standalone classifier (sentiment polarity, NER, QA). By contrast, we treat the LLM as a frozen, zero-shot *numerical* feature extractor—analogous to using a pre-trained vision model as a fixed encoder for downstream tasks—and evaluate its representations against a continuous downstream RL objective rather than NLP classification metrics.

Prompt & Pipeline Optimization. Frameworks like DSPy (Khattab et al., 2023) formally compile language model calls by optimizing prompts against programmatic validation metrics. While these frameworks routinely optimize for exact-match accuracy or retrieval scores, our optimization loop extends this paradigm to a domain-specific continuous metric (rank correlation of predicted returns) prior to RL integration, treating the prompt as a discrete hyperparameter in the same spirit as architecture search.

Information Asymmetry and News Latency. Insider-trading research (Seyhun, 1998) established that information edges decay rapidly as they become public. For free-tier news feeds, institutional desks have already acted on the headline by the time a retail pipeline ingests it. This latency shapes the horizon at which LLM-derived features can carry signal—a constraint we quantify in Section 5.1.

3 System Architecture

3.1 Data Ingestion Pipeline

Reproducible feature extraction requires a deterministic historical record. We built a concurrent Go pipeline that ingests: (1) news from Alpaca's Benzinga feed, (2) RSS aggregations from financial outlets, and (3) SEC EDGAR filings (Form 4 insider trades and 8-K disclosures). Raw text is bundled per ticker per trading day and persisted to a SQLite database (the "backfill layer").

This backfill-first design prevents look-ahead bias: the LLM always reads from a frozen snapshot whose information boundary is strictly $\leq$ day d. It also allows the prompt-optimization loop (Section 4) to re-extract features from identical text without re-scraping.

Because our pipeline enforces strict daily information boundaries (aggregating all feeds at the close of day d), it is intentionally blind to intra-day volatility and high-frequency market microstructure. Consequently, the RL agent operates on smoothed, day-over-day narrative shifts rather than instantaneous headline shocks. We measure the effect of this daily resolution in Section 5.1.

3.2 Feature Schema

A frontier LLM (`Qwen3 235B A22B Instruct 2507`) processes each ticker's daily bundle and outputs a structured JSON mapping into continuous RL observation bounds:

1. **Stock-level LLM features (4 dims):**
   - `sentiment` $\in [-1, 1]$: Directional conviction from the daily news flow (-1: very bearish, $+1$: very bullish).
   - `impact` $\in [0, 1]$: Financial materiality of the bundle (e.g., CEO resignation vs. routine marketing).
   - `conflicting_signals` $\in [0, 1]$: Evidentiary contradiction across competing sources within the same bundle.
   - `news_novelty` $\in [0, 1]$: Divergence of the current day's narrative from historical baselines.

2. **Macroeconomic features (5 dims):** `vix` (market anxiety), `treasury_10y` (discount rate proxy), and `credit_spread` (corporate default risk), sourced from FRED. Two additional LLM-inferred regime flags (`market_sentiment`, `macro_event_flag`) complement the systematic landscape.

This JSON constraint grounds high-dimensional linguistic narratives into an explicit 9-dimensional vector digestible by the downstream MLP. Table 1 details the summary statistics.

3.3 RL Policy Agent

We use FinRL (Liu et al., 2020) to construct the trading environment and PPO agent. The composite state vector is:

$$S_t = [P_t \parallel \mathcal{E}_t \parallel M_t \parallel B_t] \tag{1}$$

where P_t is OHLCV (open, high, low, close, volume) bars plus technical indicators, $\mathcal{E}_t$ is ticker-level LLM features, M_t is macro features, and B_t is portfolio state. Observations are normalized via `VecNormalize` (`clip_obs` $= 10$).

The agent is a PPO with an MLP policy, trained for 500k timesteps on 2023–2024 data—a cutoff supported by the convergence analysis in Figure 3. The RL framework trades a 21-ticker universe ($\sim$10,500 rows).

All results are averaged across five seeds $\{0, 1, 2, 3, 42\}$ with deterministic hierarchical seeding (PyTorch `manual_seed` + SB3 environment seeds) to ensure reproducibility.

4 Prompt Optimization

Standard prompt engineering does not scale when the downstream task is continuous-valued RL rather than classification. Standard text-quality metrics like BLEU or classification accuracy do not measure whether extracted features rank future returns correctly. We designed an automated optimization loop that treats the extraction prompt as a discrete hyperparameter and tunes it against the Information Coefficient (IC).

The pipeline operates via a feedback loop with Anthropic's Claude API as meta-optimizer. The workflow, illustrated in Figure 4, follows six steps:

1. **Initialize** a baseline chain-of-thought prompt (`v0`).
2. **Define gates:** IC (rank correlation stability), Hit% (directional accuracy), Quintile Spread (monotonicity of ranked portfolios). These measure whether the LLM's numerical outputs predict future returns, unlike text-quality metrics (e.g., BLEU, classification accuracy) which only assess linguistic fidelity.

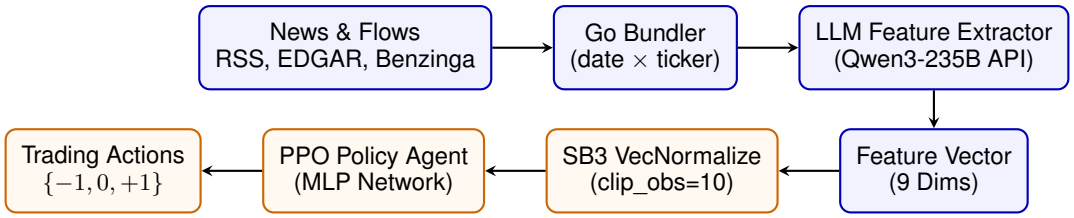

Figure 2: System overview. A Go-based ingestion pipeline collects news, filings, and macro data into a relational store. The LLM produces per-ticker feature vectors, which the PPO agent consumes alongside OHLCV (open, high, low, close, volume) bars and technical indicators.

Table 1: Feature Value Distributions (2023-2024 training period)

Feature	Group	Mean	Std	Min	Max	% Non-zero	Scale
sentiment	LLM News	-0.007	0.290	-1.000	0.900	71.8%	[-1, 1]
impact	LLM News	0.292	0.233	0.000	1.000	71.9%	[0, 1]
`conflicting_signals`	LLM News	0.076	0.182	0.000	1.000	18.5%	[0, 1]
`news_novelty`	LLM News	0.737	0.424	0.000	1.000	77.9%	[0, 1]
`market_sentiment`	Market Regime	0.175	0.405	-1.000	0.800	98.7%	[-1, 1]
`macro_event_flag`	Market Regime	0.481	0.500	0.000	1.000	48.1%	{0, 1}
`treasury_10y`	Macro	4.085	0.370	3.300	4.980	100.0%	[3.5, 5.5]
vix	Macro	16.120	3.262	11.860	38.570	100.0%	[12, 80]
`credit_spread`	Macro	3.675	0.620	2.600	5.220	100.0%	[0.5, 3.0]

VecNormalize (clip_obs=10) applied during training normalizes these distributions.

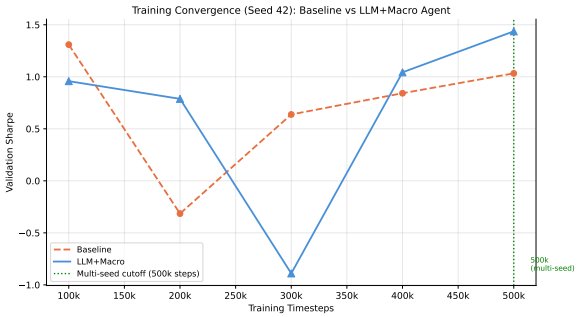

Figure 3: Training convergence (seed 42). Validation Sharpe at each 100k-step checkpoint on H1 2025. Both agents plateau by 400–500k steps; the vertical line marks the multi-seed cutoff.

3. **Meta-optimize:** Claude suggests a discrete structural mutation (e.g., adding few-shot examples, redefining output ranges).
4. **Extract:** Features for a one-month subset (January 2025, 769 bundles, 38 tickers) using the mutated prompt via Qwen3.
5. **Evaluate:** Compute IC gates.
6. **Iterate:** If gates fail, send metrics back to Claude as feedback; if passed, freeze the prompt.

Each of the five mutations targets a distinct failure mode observed in pilot runs on the baseline prompt: semantic ambiguity (mut1), numerical mis-calibration (mut2), signal interference between news and flow modalities (mut3), recency-vs-surprise framing (mut4), and a composition test of whether gains stack (mut5). The taxonomy spans standard prompt-engineering levers—specification, calibration, decomposition, framing, composition—so the loop searches a structured mutation space rather than free-form rewrites. As shown in Table 2:

1. **mut1 (Impact-Surprise):** Redefines `impact` as the magnitude of market surprise relative to consensus.
2. **mut2 (Few-Shot):** Adds three concrete calibration examples (priced-in beat, genuine surprise, and conflicting flow).
3. **mut3 (Separate Reasoning):** Decouples reasoning into distinct news and flow signal analysis blocks.
4. **mut4 (Counterfactual):** Uses a "what-if" counterfactual test to anchor sentiment scores against market defaults.
5. **mut5 (Combined):** Merges mut1, mut2, and mut4 into a single prompt.

The results demonstrate that explicit few-shot

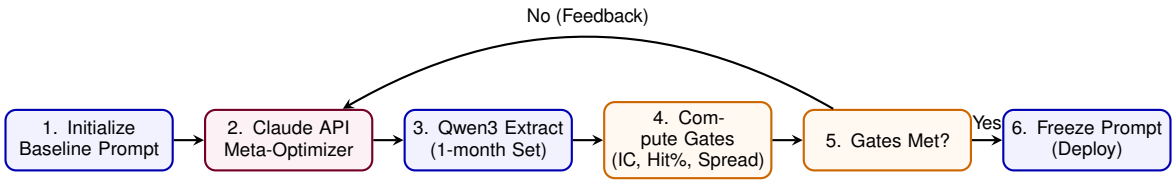

Figure 4: Prompt optimization workflow. Claude iteratively refines the extraction prompt based on downstream financial metric feedback.

Table 2: Prompt mutation results. Top: January 2025 optimization set (769 bundles, 38 tickers). Bottom: February 2025 out-of-sample validation.

(a) Jan 2025 — Optimization Set

Candidate	Hypothesis	IC IR	Hit%	Spread	Brier	Comp.
v3-baseline	Chain-of-thought	−0.024	45.5	−1.07%	0.305	−0.144
mut1	Impact = surprise	−0.075	22.2	−1.66%	0.291	−0.325
mut2-few-shot†	Few-shot examples	**+0.104**	71.4	+0.22%	0.288	+0.191
mut3-separate	Separate reasoning	+0.029	60.0	+0.75%	0.300	+0.134
mut4	Counterfactual	−0.098	57.1	−0.29%	0.289	+0.052
mut5-combined	mut1+2+4	−0.175	55.6	−2.06%	0.290	−0.243

†Frozen as `v4-stable-core`.

(b) Feb 2025 — OOS Validation

Candidate	IC IR	Hit%	Spread	Brier	Comp.
v3-baseline	−0.212	50.0	N/A	0.293	−0.210
mut2-few-shot	−0.044	25.0	N/A	0.281	−0.230

calibration (`mut2-few-shot`) was the critical driver of performance, improving IC IR to +0.104 compared to the baseline's −0.024. In contrast, combining multiple structural changes (`mut5`) caused a performance collapse, likely due to instruction bloat and conflicting reasoning anchors. The prompt was frozen as `v4-stable-core` (Figure 5) and used for all subsequent extraction.

4.1 Adequacy Gates

Before committing to RL training, we evaluated the extracted features against four predictive adequacy gates (Table 3). The prompt cleared all four thresholds, notably exceeding the IC requirement by a substantial margin. Because our optimization loop explicitly targets downstream predictive utility, this strong rank correlation confirms the prompt's validity in generating a continuous observation space for the downstream RL agent.

Table 3: Adequacy gate assessment (`mut2-few-shot`, Jan 2025).

Metric	Gate Threshold	Value	Status
signal_coverage	≥ 0.25	0.408	**PASS**
ic_ir_5d	≥ 0.05	+0.104	**PASS**
quintile_spread	> 0	+0.002	**PASS**
hit_rate	≥ 0.52	0.714	**PASS**

†Structural issue: `pred_prob` uses impact as confidence, but impact measures materiality, not prediction certainty.

5 Empirical Evaluation

Training spans 2023–2024. Validation covers 2025 H1 (January–June, 120 trading days), a period of tariff-driven macro volatility. The held-out test covers 2025 H2 (July–December), a materially calmer regime.

5.1 Feature Validity

Before examining downstream RL performance, we verify that the optimized prompt produces features with genuine predictive signal. Table 4 reports the Information Coefficient (IC)—the Spearman rank correlation between each feature's daily values and subsequent 5-day returns, averaged

Prompt `v1-stable-core` Feature Extraction Template

System: You are a quantitative feature extraction engine for an institutional trading system. Given an event bundle about a specific ticker, output ONLY valid JSON with numerical features...
User: Extract trading features for ticker {{.Ticker}} from the following event bundle on {{.Date}}. Consider ALL signal types together — news articles, insider trades, and options flow. Output ONLY valid JSON matching this schema.
Field definitions (fill in this order):

- `reasoning`: Write ONE sentence summarizing the key signal and WHY it moves the stock.
- `sentiment`: Predicted FUTURE PRICE TRAJECTORY over the next 1-5 trading days. [-1.0 strongly bearish, 0.0 no edge, +1.0 strongly bullish] Focus on SURPRISE vs CONSENSUS, not absolute tone.
- `impact`: Materiality of the news. [0.0 trivial, 1.0 highly market-moving].
- `conflicting_signals`: Do the signals point in contradictory directions? [0.0 aligned, 1.0 strongly contradictory].
- `insider_trading`: Insider BUYS are a moderately bullish signal. Insider SELLS are WEAK (usually scheduled 10b5-1 plans).

Calibration Example A — Priced-in beat: *"AAPL reports Q4 earnings beating consensus by 2%, in line with whisper numbers."* → `sentiment: +0.1, impact: 0.2, conflicting_signals: 0.0`
Calibration Example B — Genuine surprise: *"NVDA unexpectedly raises full-year guidance 40% above Street estimates..."* → `sentiment: +0.8, impact: 0.9, conflicting_signals: 0.1`

Figure 5: Abbreviated visualization of `v1-stable-core`. Few-shot calibration examples anchoring numerical output ranges proved necessary to maximize IC.

across trading days. IC normalizes by the standard deviation of daily ICs, analogous to a signal-to-noise ratio. Among the LLM-derived features, `conflicting_signals` (IC $= 0.233$, $t = 2.52$) and `impact` (0.177, $t = 1.91$) carry the strongest signal.

The macro features (VIX, Treasury, credit spread) register IC ≈ 0 by construction: they are identical across all tickers on a given day, so cross-sectional rank correlation is undefined. Their value is purely time-serial—they tell the RL agent *when* to trade cautiously, not *which* ticker to favor. Figure 6 confirms this asymmetry using a gradient-boosted tree trained on 5-day forward returns as a model-agnostic surrogate for the observation space (the deployed MLP policy itself does not expose comparable feature attributions): the tree assigns 58% cumulative split-importance to macro features despite their zero cross-sectional IC.

Figure 7 shows sentiment IC as a function of forecast horizon. The signal is near zero at one day, peaks at 3–10 days, and decays by day 20—consistent with the delayed-news constraint: the 1-day edge has been captured by faster participants, leaving only medium-term narrative drift.

5.2 Validation: Macro-Shock Regime (H1 2025)

Cross-sectional feature evaluation is conducted on a broader 38-ticker US large-cap signal universe, while the downstream RL environment restricts execution to a 21-ticker liquid trading subset plus SPY; the exact ticker lists are provided in Appendix A.

Given that the LLM features carry genuine signal, we now ask whether this translates into downstream RL performance. Table 5 presents the four-configuration ablation on H1 2025. No configuration significantly outperforms the price-only baseline (all paired t-test $p > 0.1$).

The pattern is informative. LLM-only is the worst configuration (Sharpe -0.411): trading on idiosyncratic stock news during a systemic shock amounts to ignoring the dominant risk factor. Macro-only (-0.007) tracks the baseline closely. LLM+Macro (-0.267) is better than LLM-only because the macro features provide a "regime brake," but the noisy LLM signals still drag it below baseline. Figure 8 visualizes this: during elevated VIX, the LLM-augmented agent systematically underperforms.

5.3 Held-Out Test: Calm Regime (H2 2025)

If idiosyncratic news fails only when macro risk dominates, a calmer regime should restore its value. We locked all model parameters and evaluated once on H2 2025 (Table 6).

All three augmented configurations now exceed the baseline (Sharpe 0.809). LLM-only recovers to 1.001, suggesting that stock-level narratives can capture signal when systemic risk subsides. However, Macro-only remains strongest (1.099, $\Delta = +0.290$), and LLM+Macro (1.038, $p = 0.49$ vs. baseline) does not reach statistical significance at $N = 5$ seeds. SPY buy-and-hold (1.756) outper-

Table 4: Feature IC analysis (5-day forward return, 2023–2024 training period). Cross-sectional IC for macro features is zero by construction: they are constant across tickers on any given day and carry regime information detectable only by the RL policy, not by cross-sectional ranking.

Feature	Group	IC Mean	IC IR	t-stat	% Pos	N
sentiment	LLM News	0.016	**0.093**	1.00	0.5	117
impact	LLM News	0.029	**0.177**	1.91	0.6	117
conflicting_signals	LLM News	0.040	**0.233**	2.52	0.6	117
news_novelty	LLM News	0.011	**0.065**	0.70	0.5	117

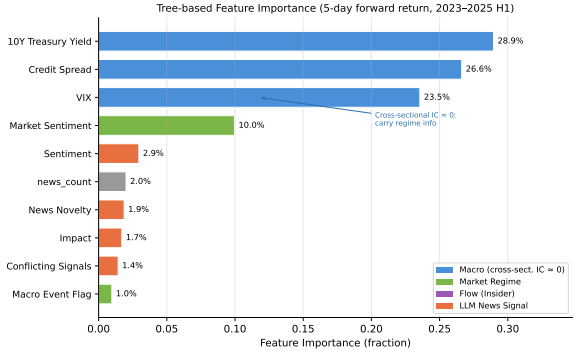

Figure 6: Feature importance from a gradient-boosted tree on 5-day forward returns. Macro features dominate (credit_spread 28%, VIX 16%, treasury 14%) despite zero cross-sectional IC.

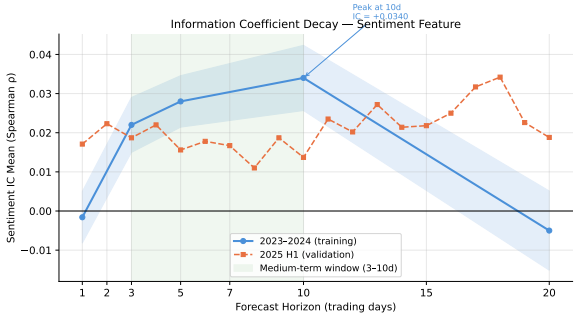

Figure 7: IC decay for sentiment across horizons. Signal peaks at 3–10 days, consistent with the delayed-news constraint.

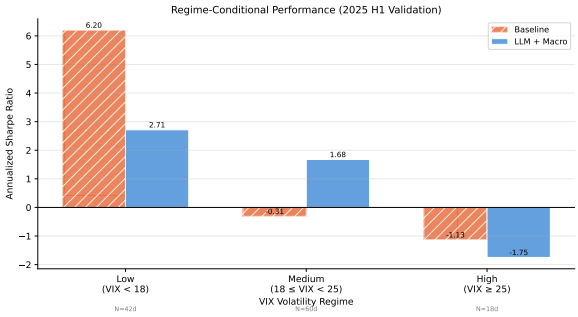

Figure 8: Regime-conditional performance. During elevated VIX (shaded), the LLM-augmented agent underperforms the baseline. In low-volatility windows the gap narrows or reverses.

forms all RL configurations during the late-2025 bull rally; our contribution is the *relative* ablation, not absolute excess return.

5.4 Robustness Checks

Transaction cost sensitivity. Table 7 varies transaction costs from 0 to 50 bp for seed 42 on H1 2025. The baseline's advantage over LLM+Macro is stable across all cost levels, confirming that the H1 null result is not an artifact of unrealistic friction assumptions.

6 Discussion

The feature-to-policy gap. Our results expose a gap between intermediate representation quality and downstream task performance—a phenomenon familiar from transfer learning, where strong pretrained features do not guarantee fine-tuning success under distribution shift. The prompt optimization loop successfully produces features with high rank correlation to future returns. However, this signal only translates into RL policy improvement when the test distribution resembles training conditions. When a macroeconomic shock shifts the data distribution (H1 2025), the RL agent cannot exploit features whose predictive structure has changed.

Macro features as a regime brake. VIX, Treasury yields, and credit spreads do not predict *which* stocks will outperform—their cross-sectional IC is zero (Table 4). Instead, they tell the RL agent when the market environment has shifted, allowing it to reduce exposure. Using a gradient-boosted tree on the same observation space as a model-agnostic proxy (the MLP policy does not expose comparable attributions), macro inputs receive 58% of split-importance (Figure 6); the ordering is consistent with the ablations in Tables 5–6. This time-serial

Table 5: H1 2025 validation ablation ($N = 5$ seeds, 120 trading days). Mean±std across seeds. ΔSharpe vs. Baseline.

Config	N Feats	Sharpe	Return%	Max DD%	ΔSharpe
Baseline (OHLCV + Tech)	0	0.010 ± 0.618	-5.03 ± 18.00	34.91 ± 6.01	—
LLM Signals Only†	6	-0.411 ± 0.690	-16.42 ± 18.81	42.12 ± 8.11	-0.421
Macro Only‡	5	-0.007 ± 0.355	-6.16 ± 9.74	34.78 ± 9.28	-0.017
LLM + Macro (Full)	10	-0.267 ± 0.284	-13.91 ± 7.41	38.66 ± 6.34	-0.276

† LLM-only: sentiment, impact, conflicting_signals, news_novelty + 2 regime flags. ‡ Macro-only: VIX, treasury_10y, credit_spread + 2 regime flags. No config beats Baseline ($p > 0.1$, paired t, $N = 5$).

Table 6: H2 2025 held-out test ($N = 5$ seeds). Mean±std. ΔSharpe vs. Baseline.

Config	N Feats	Sharpe	Return%	Max DD%	ΔSharpe
Baseline (OHLCV + Tech)	0	0.809 ± 0.333	11.29 ± 6.21	14.96 ± 4.04	—
LLM Signals Only†	5	1.001 ± 0.853	12.98 ± 8.93	16.27 ± 6.59	+0.192
Macro Only‡	5	1.099 ± 0.695	16.04 ± 9.53	15.30 ± 3.15	+0.290
LLM + Macro (Full)	10	1.038 ± 0.424	15.14 ± 6.19	17.27 ± 2.18	+0.229
SPY buy-and-hold§	—	1.756§	11.87§	5.07§	+0.947
Paired t-test (LLM+Macro vs. Baseline): $t(4) = 0.76$, $p = 0.4873$					

†/‡ Feature groups as in Table 5. § SPY buy-and-hold (single value). Models fixed before unlocking test set.

conditioning is invisible to cross-sectional IC.

Information latency as a feature constraint. Our reliance on free-tier news feeds bounds the temporal resolution of extractable signal. The IC-decay curve (Figure 7) shows that predictive power peaks at 3–10 days. From an ML perspective, this is an input-quality constraint analogous to training on low-resolution images—the representation is valid but resolution-limited.

Limitations.

- **Statistical power.** Five seeds on a 120-day window yield low power (<50% to detect ΔSharpe=0.3 at $\sigma = 0.4$).
- **Absolute performance.** All agents underperform SPY buy-and-hold. The contribution is the relative ablation, not absolute return.
- **Narrow universe.** 21 large-cap US equities. Generalization is an open question.
- **Input resolution.** A low-latency news feed may restore short-horizon signal.

7 Conclusion

We present an automated prompt-optimization pipeline that tunes a frozen LLM to produce predictive numerical features for an RL trading agent. Valid intermediate representations do not automatically yield downstream improvement: the gap between feature quality and policy robustness is regime-dependent—LLM-derived features help when the test distribution is stable but degrade under macroeconomic shocks. Evaluating LLM-generated features on intermediate metrics alone is insufficient; multi-regime out-of-sample testing should become standard for any pipeline feeding LLM representations into a downstream learner.

A Ticker Universes

Signal universe (38 tickers). LLM feature extraction and prompt-optimization experiments use the following 38-ticker signal universe:

> AAPL, ABBV, ADBE, AMD, AMZN, AVGO, BA, BAC, CAT, COST, CRM, CVX, GE, GOOGL, GS, HD, INTC, IWM, JNJ, JPM, LLY, MA, MCD, META, MSFT, NFLX, NKE, NVDA, ORCL, QCOM, QQQ, RTX, SPY, TSLA, UNH, V, WMT, XOM.

Tradable RL universe (21 tickers). The downstream RL environment restricts execution to the following 21-ticker liquid trading universe:

> AAPL, MSFT, AMZN, NVDA, META, TSLA, AMD, NFLX, ADBE, QCOM, JPM, V, MA, GS, UNH, LLY, XOM, WMT, BA, CAT, SPY.

Why the universes differ. The broader 38-ticker universe improves cross-sectional feature evaluation and includes contextual instruments such as

Table 7: Transaction cost sensitivity (2025 H1, seed 42). The baseline advantage over LLM+Macro is robust across all cost levels, confirming the H1 null result is not a friction artifact.

Cost (bp)	Cost (%)	Baseline Sharpe	LLM+Macro Sharpe	Δ Sharpe	LLM Win?
0	0.00%	1.231	0.753	-0.478	×
5	0.05%	1.202	0.730	-0.472	×
10	0.10%	1.156	0.674	-0.482	×
20	0.20%	1.108	0.549	-0.559	×
50	0.50%	0.929	0.580	-0.349	×

Δ Sharpe = LLM Sharpe - Baseline Sharpe. Standard cost (0.10%) highlighted.

QQQ, IWM, and GOOGL. The RL agent trades only the 21-ticker liquid subset. GOOGL is excluded from trading for compliance reasons, although its signals remain available in the upstream feature store.

References

Andrew Ang and Geert Bekaert. 2002. International asset allocation with regime shifts. *The Review of Financial Studies*, 15(4):1137–1187.

Ben Hambly, Ruiwei Xu, and Huining Yang. 2023. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, and 1 others. 2023. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. *arXiv preprint arXiv:2310.03714*.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191.

Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Linyi Yang, Bowen Xiao, and William Wang. 2020. Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance. *Deep RL Workshop, NeurIPS 2020*.

Alejandro Lopez-Lira and Yuehua Tang. 2023. Can chatgpt forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619*.

H Nejat Seyhun. 1998. *Investment intelligence from insider trading*. MIT press.

Hao Sun and 1 others. 2023. Market regime aware reinforcement learning for quantitative trading. *Proceedings of the ICAIF*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Drozdov, Brad Mullis, Chenyu Yue, Steve Ostrum, and 1 others. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*.

Boyu Zhang and 1 others. 2023. Pixtral & finma: Instruct-finllm for financial domain. *arXiv preprint arXiv:2306.06031*.