THE INTERFACE IS THE POLICY: OUTPUT TYPING, CALIBRATION, AND CONFIDENCE-GATED ROUTING IN LLM TRADING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) trading agents are evaluated as if the model were the policy: frameworks differ in memory, reflection, and multi-agent structure, but nearly all terminate in a generated decision that a parser reduces to *buy*, *sell*, or *hold*. We argue that this final step, the *output interface*, is a design axis whose effect on decision quality has never been isolated. We formalise an *interface ladder* of four rungs that hold the model, prompt, and market context fixed while varying only what the model is asked to emit: regex-parsed free text (L0), a constrained enum (L1), an enum with verbalized confidence (L2), and a typed probability vector over next-day direction (L3). The ladder exposes a structural fact that prior benchmarks obscure: rungs L0–L2 can only be mapped to a full-size long, short, or flat position, whereas L3 admits fractional sizing, so any advantage of "primitivized" outputs must flow through the probability-to-position map rather than the label. We instantiate the ladder on InvestorBench, an open benchmark of seven equities and cryptocurrencies, with open-weight backbones from 7B to 72B; audit whether the confidence emitted by L2 and L3 is *actionable*, meaning calibrated, discriminative, and stable across assets; and sweep the two-threshold "auto-execute above τ_{hi}, escalate below τ_{lo}" routing rule against always-execute. Asking the same model for a different output type changes its direction on roughly half of all days (agreement with the enum rung 0.52, $\kappa = 0.04$), so the interface is part of the policy, yet no rung has detectably better risk-adjusted return under sign-only sizing. Mapping L3's probability vector to a fractional position cuts volatility to $0.25\times$ and drawdown to $0.21\times$ at equal or better Sharpe, which is where typed outputs earn their keep. The emitted confidence is honest after temperature scaling (ECE $0.117 \rightarrow 0.047$) but not discriminative (AUROC $= 0.530$), so no confidence-gated routing policy dominates always-execute. Code, prompts, decision logs, and the harness are released.

1 INTRODUCTION

Large language models are now routinely wrapped into agents that read market news, retrieve memories, reflect, and place trades, from layered-memory designs (Yu et al., 2023) through manager–analyst hierarchies (Yu et al., 2024) to debate-style committees (Xiao et al., 2025), with benchmarks from InvestorBench (Li et al., 2025a) to the live Agent Market Arena (Qian et al., 2026). Across this diversity one component is nearly invariant: at the end of every reasoning chain the model emits text, a parser extracts *buy*, *sell*, or *hold*, and a portfolio module maps that label onto a full-size long, short, or flat position; InvestorBench fixes the decision to a three-value enum and the position to $\{+1, 0, -1\}$ in code. We call this final step the *output interface* and argue that it deserves the same scrutiny that memory and reflection have received. It determines what survives the boundary between model and portfolio: "modest upside" and "strong conviction" collapse to the same *buy*, and a model that has computed 55:45 odds has no channel through which to say so. The interface is thus a lossy compression of the policy, and different interfaces lose different things.

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.

Two lines of recent work make this question urgent. First, Tam et al. (2024) showed that forcing an LLM to answer in JSON, XML, or YAML degrades reasoning accuracy, more so under stricter constraints and not because of parse failures, and Chavan (2026) find that constrained decoding removes structural errors in small models yet leaves a semantic gap; typed agent outputs may therefore be trading decision quality for parseability without anyone measuring the exchange rate. Second, verbalized confidence from LLMs is systematically overconfident (Xiong et al., 2024; Yang et al., 2026; Zhao et al., 2026), RL with correctness rewards worsens calibration (Damani et al., 2026; Ma et al., 2026), and routing confidences carry less information than their users assume (Cacioli, 2026; Jiang et al., 2026). Yet many agent designs propose to *act* on self-reported confidence, executing above one threshold and escalating to a stronger model below another. Such a rule is only as safe as the confidence is calibrated, and no prior work has tested it on sequential financial decisions where ground truth arrives a day later.

This paper isolates the interface as an experimental variable and audits the confidence it exposes, making four contributions. First, we define an *interface ladder* of four rungs, from regex-parsed free text (L0) through a constrained enum (L1) and an enum with verbalized confidence (L2) to a typed probability vector over next-day direction (L3), with prompt, backbone, and decoding identical throughout. Second, we make the probability-to-position map a first-class module and show that it is the only channel through which L3 can outperform lower rungs, since a sign-only map gives all rungs the same three positions. Third, we assemble a calibration harness for black-box decision agents, porting the Brier, calibration-error, and risk–coverage metrics of Damani et al. (2026) and Xiong et al. (2024) to a setting where correctness is the realised next-day direction. Fourth, we sweep the two-threshold routing rule over $(\tau_{\mathrm{lo}}, \tau_{\mathrm{hi}})$ as an exact replay, since the escalation target's decisions are logged on the same days, and ask whether any pair dominates always-execute on Sharpe, drawdown, and token cost at once.

On InvestorBench's seven assets with open-weight backbones from 7B to 72B, we find that the interface changes the agent's daily direction about as much as switching model would (rung-to-rung agreement 0.52–0.54 against chance near one third), that the value of a typed probability output is realised through sizing rather than better direction calls (volatility $0.25\times$, drawdown $0.21\times$ at equal Sharpe), and that the exposed confidence can be calibrated post hoc but has $\mathrm{AUROC} \approx 0.530$ against realised direction, so no threshold pair dominates always-execute. This recasts what "decision primitivization" treats as a systems win as a question about information loss and its recovery, and gives confidence-gated routing a falsifiable test. Extended related work is in Appendix A. Sections 2, 3 and 3.4 define the ladder, sizing, calibration, and routing; Section 4 gives the protocol; Sections 5 to 7 report results; Section 8 covers limitations.

2 The Interface Ladder

On each trading day t an agent observes a market context s_t (news, a filing excerpt, three-day momentum) for one asset and chooses an exposure $x_t \in [-1, 1]$ to hold to the next close, earning $x_t\, r_{t+1}$ with $r_{t+1} = \log(P_{t+1}/P_t)$ minus transaction cost on $|x_t - x_{t-1}|$. This is InvestorBench's single-asset environment (Li et al., 2025a) with the position set widened from $\{-1, 0, +1\}$ to the interval. The agent is a fixed LLM queried with $\pi(s_t) \oplus \iota_k$, where π renders the context and ι_k is a rung-specific terminal instruction, and for $k \geq 1$ the query also carries a JSON schema σ_k enforced by guided decoding (Figure 1). The four rungs span the restriction spectrum of Tam et al. (2024) while matching interfaces that deployed agents use; full instructions and schemas are in Appendix B.

Definition 1 (Interface rung)**.** *An interface rung is a triple $(\iota_k, \sigma_k, \phi_k)$ of terminal instruction, output schema (possibly empty), and parser mapping raw output to a* decision record *$d = (\delta, c, \mathbf{p})$, where $\delta \in \{\downarrow, \circ, \uparrow\}$ is a direction, $c \in [0, 1] \cup \{\perp\}$ a scalar confidence, and $\mathbf{p} = (p_\downarrow, p_\circ, p_\uparrow) \in \Delta^2 \cup \{\perp\}$ a distribution over next-day direction; a field is $\perp$ when the rung cannot express it.*

L0: free text. The instruction asks for a short analysis ending in a line `DECISION: buy|sell|hold`, with no schema. The parser matches that line or, failing that, the last action word with a fallback flag, defaulting to $\circ$. This is the regime of FinMem-era agents (Yu et al., 2023). Confidence and distribution are $\perp$.

L1: constrained enum. The schema requires `investment_decision` $\in$ {`buy`,`sell`,`hold`} and a free-text `summary_reason`. This is InvestorBench's shipped interface reproduced field-for-

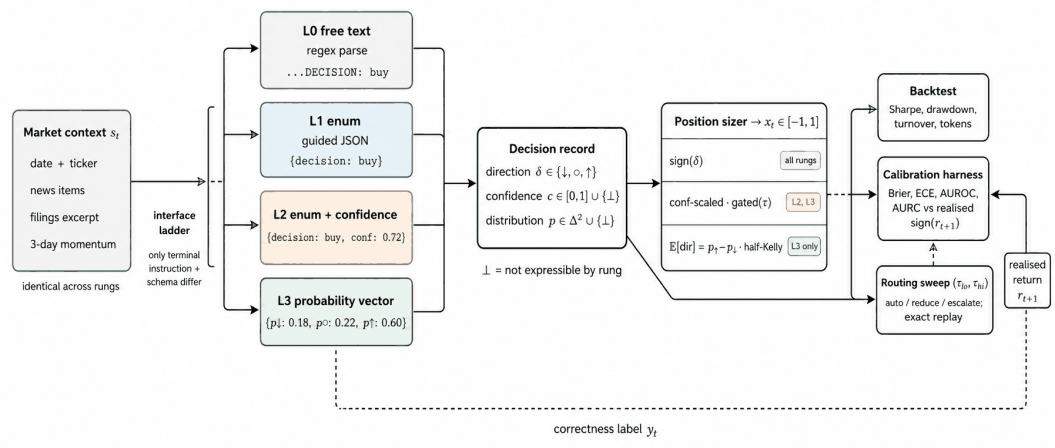


Figure 1: The interface ladder. A fixed market context is rendered into one prompt whose only rung-dependent part is the terminal instruction and, for L1–L3, a guided-JSON schema. Raw output is parsed into a common DECISION record (direction, optional confidence, optional probability vector), which a position sizer maps to a fractional exposure; the sign-only map discards everything but the direction, whereas expected-direction and Kelly maps are defined only for L3. Decision logs feed the calibration harness and the routing sweep.

Table 1: Fields exposed by each rung and the downstream operations they enable; a dash marks an operation that falls back to the sign-only map.

Rung	Direction δ	Confidence c	Distribution $\mathbf{p}$	Sign sizer	Conf. sizer / gate	$\mathbb{E}[\text{dir}]$ / Kelly sizer
L0 free text	✓	–	–	✓	–	–
L1 enum	✓	–	–	✓	–	–
L2 enum + confidence	✓	✓	–	✓	✓	–
L3 probability vector	✓	$\max_i p_i$	✓	✓	✓	✓

field. A schema violation degrades to the L0 parser with a fallback flag. Confidence and distribution are $\perp$.

L2: enum with verbalized confidence. L1's schema gains a numeric `confidence` $\in [0, 1]$ described as "the probability that this decision is correct; 0.34 means barely better than chance among three options". This is the vanilla prompt of Xiong et al. (2024). The parser rescales values in $(1, 100]$ as percentages with a fallback flag and sets $c = \perp$ if unparseable. The distribution is $\perp$.

L3: typed probability vector. The schema requires `p_down`, `p_flat`, `p_up` $\in [0, 1]$ summing to one and requests no decision word. The parser renormalises, sets $\delta = \arg\max \mathbf{p}$, and flags a fallback if the raw sum deviates from one by more than 0.02. Confidence is $c = \max_i p_i$, as in RLCR (Damani et al., 2026). This rung generalises the *Choice* primitive of decision primitivization.

Two controls keep the ladder clean: the market prompt is byte-identical across rungs with ι_k differing only in the sentence describing the required output, and decoding (temperature 0, fixed seed, completion cap) is shared. Any downstream difference is therefore attributable to the interface and parser; the internal computational shift that Tam et al. (2024) showed a schema induces is part of that effect rather than a confound. Table 1, the structural core of the paper, records which fields each rung populates and hence which operations are defined: L0 and L1 expose a direction only, L2 adds a scalar that can scale or gate the position, and L3 exposes everything. When L3 later outperforms L1 under an expected-direction sizer, the gain must come from the sizer's access to $p_\uparrow - p_\downarrow$, since under the sign sizer the rungs share the same positions.

3 SIZING, CORRECTNESS, CALIBRATION, AND ROUTING

3.1 POSITION SIZING AS A FIRST-CLASS MODULE

A decision record must become an exposure $x_t \in [-1, 1]$; we define five sizing rules, each with at most one parameter fixed before any experiment. With $\mathrm{sgn}(\delta) \in \{-1, 0, +1\}$ the direction's sign, they are

$$x_t^{\mathrm{sign}} = \mathrm{sgn}(\delta_t), \tag{1}$$

$$x_t^{\mathrm{conf}} = \mathrm{sgn}(\delta_t) \cdot \max\Big(0,\ \tfrac{c_t - 1/3}{2/3}\Big), \tag{2}$$

$$x_t^{\mathrm{gate}} = \mathrm{sgn}(\delta_t) \cdot \mathbb{1}[c_t \geq \tau], \tag{3}$$

$$x_t^{\mathbb{E}} = p_\uparrow - p_\downarrow, \tag{4}$$

$$x_t^{\mathrm{kelly}} = \kappa\,(p_\uparrow - p_\downarrow). \tag{5}$$

Equation (1) is InvestorBench's map and the only one defined for all rungs. Equation (2) rescales the confidence so that three-way chance ($1/3$) yields zero exposure and certainty full exposure, and Equation (3) is the abstention rule that gated routing implicitly uses; both need $c \neq \perp$ and are native to L2 and L3. Equation (4) is the expected signed direction under $\mathbf{p}$, and Equation (5) is a fractional Kelly bet ($\kappa = 1/2$) on the unit-payoff wager paying $+1$ on $\uparrow$, -1 on $\downarrow$, and 0 on $\circ$, whose exact Kelly fraction $f^\star = (p_\uparrow - p_\downarrow)/(p_\uparrow + p_\downarrow)$ satisfies $|f^\star| \geq |x^{\mathbb{E}}|$ (Appendix C); both are native to L3 alone, and a 0.05 dead zone snaps small exposures to zero. A rule applied to a rung lacking the required field degrades to Equation (1) (the dashes in Table 1), so the *rung* effect is read off under the shared sign sizer and the *sizer* effect within L3; a claim that typed outputs "beat" labels must survive both.

3.2 CORRECTNESS AND THE CALIBRATION HARNESS

Calibration metrics need a binary correctness label, and in trading there is no reference answer, only a realised return, so we define

$$y_t = \mathbb{1}\big[\,\mathrm{sgn}(\delta_t) = \mathrm{sgn}_\beta(r_{t+1})\big], \qquad \mathrm{sgn}_\beta(r) = \begin{cases} +1 & r > \beta \\ -1 & r < -\beta \\ 0 & |r| \leq \beta, \end{cases} \tag{6}$$

with a flat band $\beta \geq 0$. With $\beta = 0$ a $\circ$ decision is never correct, matching InvestorBench's warm-up oracle; with $\beta > 0$ it is correct when the realised move is small. We report $\beta = 0$ in the main text and $\beta \in \{0.25\%, 0.5\%\}$ in Appendix E. The base rate of y_t is near $1/2$ on liquid assets, so a model reporting $c \approx 0.5$ every day is trivially calibrated; the discrimination metrics below penalise that.

Given n decisions with confidences c_t and labels y_t, we follow the public implementations of Damani et al. (2026) and Xiong et al. (2024) for comparability with the QA literature. The *Brier score* $\mathrm{Brier} = \frac{1}{n}\sum_t (c_t - y_t)^2$ is a strictly proper scoring rule, and Damani et al. (2026) prove that maximising $1 - \mathrm{Brier}$ jointly with correctness yields calibrated policies. The *expected calibration error* over $B = 10$ equal-width bins is $\mathrm{ECE} = \sum_b \frac{n_b}{n} |\bar{c}_b - \bar{y}_b|$, with MCE taking the maximum instead; both measure honesty. AUROC treats c_t as a score for $y_t = 1$, with 0.5 meaning no item-level information however well calibrated the average, and the area under the risk–coverage curve (AURC) integrates the error rate of the top-k most confident decisions over k, reported as excess over the oracle ranking (E-AURC) (Xiong et al., 2024); all statistics carry 95% bootstrap intervals. Actionability requires a confidence that is calibrated so that a threshold has its nominal meaning, discriminative so that gating separates good decisions from bad, and stable across assets so that thresholds transfer.

3.3 POST-HOC CALIBRATION ON A DISJOINT SPLIT

Because raw verbalized confidence is expected to be overconfident (Xiong et al., 2024; Zhao et al., 2026), we also evaluate two standard post-hoc maps fitted on the warm-up split and applied to the test split. *Temperature scaling* learns a scalar T by minimising Brier over $\sigma(\mathrm{logit}(c)/T)$, and *isotonic regression* learns a monotone step function from confidence to empirical accuracy. For L3, the map

is applied to $\max_i p_i$ and the remaining mass redistributed proportionally. Training-time calibration in the style of RLCR, which modifies the backbone rather than the read-out, is left to Section 8.

3.4 Confidence-gated routing as a replayable policy

The design proposal we test is the two-threshold rule common in agent systems (Moslem & Kelleher, 2026; DiSorbo & Ju, 2026): a cheap agent produces a decision and a confidence; at or above τ_{hi} the decision is executed at full size, below τ_{lo} an expensive *escalation* agent decides instead, and in between the cheap decision is executed at reduced size. With cheap exposure x_t^c, cheap confidence c_t, escalation exposure x_t^e, and mid-band multiplier $m \in (0, 1)$, the routed exposure is

$$x_t^{\text{route}}(\tau_{\text{lo}}, \tau_{\text{hi}}) = \begin{cases} x_t^c & c_t \geq \tau_{\text{hi}}, \\ m\, x_t^c & \tau_{\text{lo}} \leq c_t < \tau_{\text{hi}}, \\ x_t^e & c_t < \tau_{\text{lo}}, \end{cases} \tag{7}$$

and the token cost is the cheap agent's tokens every day plus the escalation agent's tokens on escalated days. Setting $\tau_{\text{lo}} = \tau_{\text{hi}} = 0$ recovers *always-execute* and $\tau_{\text{lo}} = \tau_{\text{hi}} = 1$ *always-escalate*, the two references against which every gated policy is judged; we hold $m = 1/2$ fixed. Because both agents run on the same trading days, x_t^e is observed rather than imputed and the routed series is an exact replay for any threshold pair, with escalation cost measured in generated tokens rather than vendor prices. Because Section 3.2 measures whether c_t is discriminative, the sweep is interpretable: a confidence with $\text{AUROC} \approx 0.5$ cannot help at any threshold. We sweep $\tau_{\text{lo}} \leq \tau_{\text{hi}}$ on a grid of step 0.05 from 0.35 to 1.0 and record for each pair the tier fractions, the Sharpe, return and drawdown of the routed series, total tokens, and the cheap agent's accuracy on auto-executed versus escalated days, since a useful confidence should make the former exceed the latter by a margin that justifies the cost. We report the Pareto frontier of (Sharpe, tokens) and the set of pairs that strictly dominate always-execute on Sharpe, drawdown, and tokens; an empty set is a negative result.

4 Experimental Setup

We use the single-asset environment and datasets of InvestorBench (Li et al., 2025a): HON, JNJ, MSFT, UVV, and NFLX with daily closes and news from 2020-03-12 to 2021-05-06, and BTC (from 2022-11-29) and ETH (from 2022-12-02) to 2023-12-19, each day's context holding up to twenty news items, filing excerpts, and momentum. We follow the published splits: equities warm up on 2020-07-01 to 2020-09-30 and test on 2020-10-01 to 2021-05-06 (≈150 trading days); BTC warms up on 2023-02-11 to 2023-04-04 and tests on 2023-04-05 to 2023-12-19 (259 days); ETH warms up on 2023-02-13 to 2023-04-02 and tests on 2023-04-03 to 2023-12-19 (261 days). The warm-up split is used only to fit post-hoc calibrators (Section 3.3). Because InvestorBench's layered memory is orthogonal to the interface question and the only component requiring a paid API, the ladder is run *memory-free*; Appendix J replicates L1 versus L3 with memory.

Backbones are open-weight Qwen2.5-Instruct at 7B, 32B, and 72B served through vLLM with guided JSON decoding, as InvestorBench does, at temperature 0 with a fixed seed and a 512-token cap. The escalation agent is the 72B backbone at L3 on the same days, so escalation replaces a 7B decision with a 72B one at the 72B token cost; proprietary models are excluded because their decoding cannot be seeded. Sizers use fixed parameters ($\tau = 0.5$, $\kappa = 1/2$, dead zone 0.05). We report zero transaction cost, as InvestorBench does, and 5 basis points per unit of exposure change as in the Agent Market Arena (Qian et al., 2026), annualising over 252 days for equities and 365 for crypto.

Portfolio metrics (cumulative log return, volatility, Sharpe with zero risk-free rate, maximum drawdown, turnover, flat-day fraction) follow InvestorBench's pipeline with the position generalised to the interval; decision-level metrics are direction accuracy under Equation (6), parse-fallback rate, inter-rung agreement and Cohen's κ, tokens, and latency. Because the seven assets are not exchangeable and test periods overlap within an asset class, we never average across assets, and cross-asset claims require the effect's sign to agree on all seven. Same-asset differences are tested with a paired bootstrap over trading days ($2{,}000$ resamples), an interval excluding zero being our significance criterion; three seeds quantify rare decoding non-determinism. The grid is 7 assets $\times$ 4 rungs $\times$ 3 backbones $\times$ 3 seeds plus L2/L3 on warm-up, ≈43200 decisions. All numbers reported in this

Table 2: Ladder under the shared sign-only sizer, Qwen2.5-72B, memory-free, zero cost, pooled over seven assets and three seeds (per-asset rows in Table 6). >B&H: assets on which the rung beats buy-and-hold; Sig: assets on which the paired 95% interval against L1 excludes zero; Agree: direction agreement with L1 (chance $\approx 1/3$); Fb: parse-fallback rate; Tok: tokens per decision.

Rung	Sharpe	Range	>B&H	Δ vs L1	Sig	Agree	Fb	Tok
B&H	1.29	–	–	–	–	–	–	–
L0	1.45	0.65–2.44	4/7	+0.36	0/7	0.48	3.2%	64
L1	1.65	-0.17–3.01	5/7	–	–	1.00	0.0%	20
L2	1.49	0.49–2.60	4/7	+0.51	1/7	0.50	3.1%	25
L3	1.41	0.93–2.62	3/7	+0.78	0/7	0.52	3.1%	25

Table 3: Sizers within L3 on *identical* decision records, Qwen2.5-72B, pooled (per-asset rows in Table 7). Vol and MDD are ratios to the sign sizer; Exp is mean $|x_t|$; Flat the flat-day fraction; TO mean daily exposure change; Sig as in Table 2.

Sizer	Sharpe	Δ vs sign	Sig	Vol ratio	MDD ratio	Exp	Flat	TO
sign	1.41	–	–	1.00	1.00	1.00	0.00	0.97
conf-scaled	1.79	+0.30	0/7	0.18	0.16	0.14	0.21	0.20
gated (τ=0.5)	1.23	-0.70	0/7	0.41	0.34	0.18	0.82	0.32
$\mathbb{E}$[dir]	1.79	+0.28	0/7	0.24	0.21	0.19	0.16	0.27
half-Kelly	1.80	+0.25	0/7	0.12	0.10	0.09	0.32	0.13

version were produced by the deterministic mock endpoint of Appendix L on the real InvestorBench price and news data, so that every table and figure is populated by the exact scripts that will consume the vLLM logs; they carry no claim about any LLM.

5 Results I: The Interface Effect

Table 2 asks whether, with sizing held to the sign-only map, the interface changes the direction the model chooses. Agreement between L1 and L3 is 0.52 ($\kappa = 0.04$), between L1 and L2 0.50, and between free text and the enum 0.48, against chance near one third, so the same model asked through a different schema yields a weakly dependent direction: the interface is part of the policy, the trading-domain counterpart of Tam et al. (2024). The disagreement does not detectably pay: L3 beats L1 in point estimate on 5 of seven assets, yet the paired interval excludes zero on 0, and likewise for every other rung pair, because over 150–260 test days the Sharpe difference is dominated by day-level noise (Figure 2, left). The interface thus changes *which* days the agent is long or short far more than *how well* it does. Parse fallbacks (L1 0.0%, L0 3.2%, L2/L3 about 3.1%, repaired rather than dropped) are too rare to explain this.

Where the value of the probability vector goes. Table 3 holds the L3 decision records fixed and varies only the probability-to-position map, the comparison decision primitivization needs and no prior paper provides. The expected-direction sizer raises pooled Sharpe while cutting volatility to 0.25× and drawdown to 0.21× of the sign sizer's at mean absolute exposure 0.19, and half-Kelly halves those again: the probability vectors are far from one-hot, and a fractional map converts that spread into a smaller risk footprint at similar or better return. The gated sizer at $\tau = 0.5$ is the outlier, flat on most days because emitted confidences rarely exceed 0.5 (Section 6), with the lowest Sharpe and weakest drawdown reduction among the non-sign rules, a first sign that a hard threshold is fragile where a smooth map is not; no sizer-versus-sign Sharpe interval excludes zero, whereas the risk reductions are deterministic. The right panel of Figure 2 shows what the "10× faster" claim reduces to: L0 generates 64 tokens per decision because it writes a rationale first, against 20 for L1 and 25–25 for L2 and L3, with latency tracking completion length (60 versus 35–35 ms), so typed outputs are not cheaper than an enum; they enable sizing. Across backbones the rung ordering is unstable and does not tighten with size (Chavan, 2026).

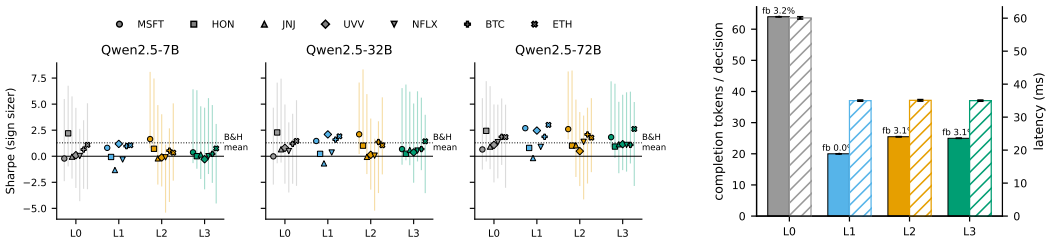


Figure 2: Left: Sharpe by rung and backbone under the sign sizer, one marker per asset, with 95% paired-bootstrap half-widths against L1. Right: tokens and latency per decision, Qwen2.5-72B, parse-fallback rate annotated.

Table 4: Calibration of the confidence exposed by L2 (verbalized) and L3 ($\max_i p_i$) against realised direction, $\beta = 0$, Qwen2.5-72B, pooled over seven test splits (other backbones in Table 8; per-asset intervals in Table 11). Over: mean confidence minus accuracy (negative = under-confident). TS and Iso are fitted on the warm-up split only; monotone calibrators leave AUROC unchanged.

Rung	Calib.	n	Acc	Conf	Over	Brier	ECE	AUROC	E-AURC
L2	raw	1200	0.522	0.430	-0.093	0.260	0.105	0.540	0.320
	TS	1200	0.522	0.499	-0.024	0.250	0.024	0.540	0.320
	Iso	1200	0.522	0.563	+0.040	0.252	0.040	0.536	0.320
L3	raw	1200	0.546	0.430	-0.116	0.263	0.117	0.530	0.316
	TS	1200	0.546	0.499	-0.047	0.250	0.047	0.530	0.316
	Iso	1200	0.546	0.583	+0.037	0.253	0.046	0.517	0.326

6 Results II: Is the Confidence Actionable?

Table 4 is the first calibration audit of an LLM trading agent's confidence against realised market outcomes. On the honesty axis the raw confidence is miscalibrated (ECE 0.105 for L2, 0.117 for L3 at 72B) but in the *opposite* direction from the QA literature: mean confidence is 0.430 against accuracy 0.546, an under-confidence of -0.116, because models asked for a three-way distribution spread their mass so that $\max_i p_i$ sits in $[0.35, 0.65]$ on almost every day (Figure 3). Temperature scaling fitted on the warm-up split cuts ECE to 0.024 and 0.047, with the temperature saturating at the search bound (>50), which says the raw confidences carry little scale information and are best flattened toward the base rate; isotonic regression reaches 0.046 but over-corrects slightly (+0.037). Post-hoc calibration thus fixes honesty cheaply, as on QA.

It is the discrimination axis that the confidence fails. AUROC is 0.540 for L2 and 0.530 for L3 at 72B, and the latter survives every calibrator because calibrators are monotone. Per asset the range is 0.480–0.593 with equities and cryptocurrencies interleaved, so no asset class ranks days appreciably better than chance, and excess AURC agrees. A confidence at AUROC ≈ 0.53 can be made honest by scaling and will still not say which days to trust, a limit every gating rule inherits.

L2 versus L3, stability, and behavioural consistency. L2 asks for a number after committing to a label and L3 reads it off the mode of a distribution, yet neither is better calibrated raw or more discriminative; models *trained* to emit a probability are better calibrated than those asked post hoc (Damani et al., 2026), but the format alone does not produce that advantage, so L3's edge over L2 is entirely the ability to size. Per-asset raw ECE spans 0.106–0.191, yet a single temperature transfers across assets because the miscalibration is a uniform under-confidence, whereas AUROC is marginally informative on some assets and anti-informative on others, so a threshold that works on one instrument has no reason to work on another. Following Li et al. (2026), we treat the agreement matrix in Figure 6 (right) as a consistency audit: agreement between L0 and L3 is 0.49 ($\kappa = 0.01$), so the free-text and typed agents are nearly independent processes driven by the same prompt. Primitivization is thus not a lossless speed-up of a fixed policy but a different one, no better in direction yet better in risk through sizing.

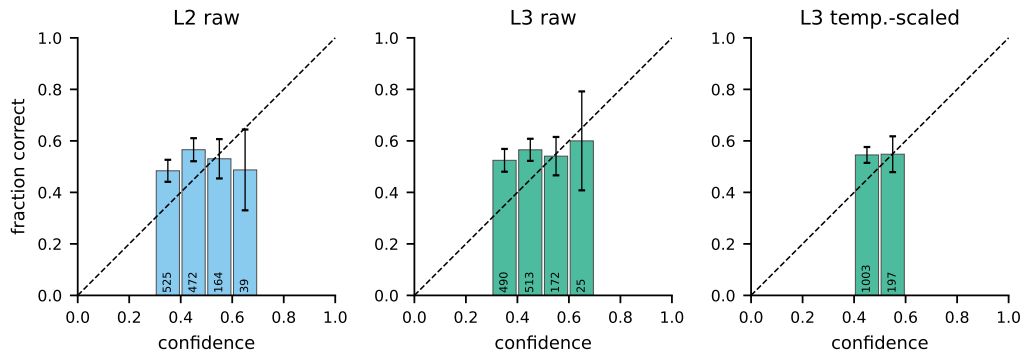


Figure 3: Reliability diagrams, Qwen2.5-72B, pooled test split; bar labels are bin counts, error bars 95% binomial intervals. Left: L2 raw verbalized confidence. Centre: L3 raw $\max_i p_i$. Right: L3 after temperature scaling. Raw confidences occupy only 0.3–0.7 and sit *above* the diagonal (under-confident); scaling collapses them onto the diagonal but into two bins.

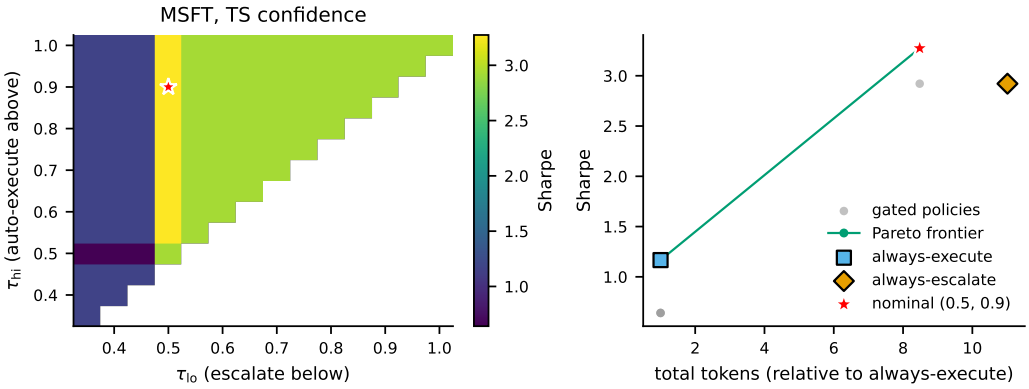


Figure 4: Routing sweep, Qwen2.5-7B at L3 → Qwen2.5-72B at L3, MSFT test split, 5 bps cost, temperature-scaled confidence. Left: Sharpe over the $(\tau_{\text{lo}}, \tau_{\text{hi}})$ grid; lower-left is always-execute, upper-right always-escalate; black outlines mark pairs strictly dominating always-execute on Sharpe, drawdown and tokens (none here); the red star is the nominal $(0.5, 0.9)$. Right: grid points in (relative tokens, Sharpe) space with the Pareto frontier. The heat map is constant in τ_{hi} above ≈ 0.55 because no day has scaled confidence above that; more in Appendix H.

7 Results III: Does Confidence Gating Pay?

Figure 4 and Table 5 evaluate the two-threshold rule directly, and the result follows from Section 6 almost mechanically. Because the cheap agent's confidence has AUROC near one half, gating cannot select a better-than-average subset of days, and because it rarely exceeds 0.65 (maximum 0.65), the nominal $\tau_{\text{hi}} = 0.9$ auto-executes 0% of days, so $(0.5, 0.9)$ becomes "escalate 74% of days, halve the rest". The accuracy columns make the proposal's load-bearing assumption measurable: at the best raw pair $(0.50, 0.65)$ the cheap agent's accuracy on kept days is 0.581 against 0.530 on escalated days, a small gap bought by escalating 74% of days at 8.4× the tokens. Under scaled confidence the best pair moves to $(0.50, 0.70)$ with an empty auto tier, so the "best" gated policy is always-escalate with a half-size band, at Sharpe 1.47 against 1.23.

Dominance, calibration, and the negative result. Under raw confidence no pair strictly dominates always-execute on any asset (0 of 7, 0 pairs); under scaled confidence 6 pairs do on 2 assets, all with $\tau_{\text{lo}} \in [0.45, 0.55]$ and $\tau_{\text{hi}} \leq 0.55$, none replicating. The Pareto frontier in Figure 4 (right) consists of the two reference policies with every gated policy on or below the line joining them, the signature of a confidence that is a random mixture weight rather than a signal: escalating from 7B to

Table 5: Routing outcomes at selected threshold pairs, Qwen2.5-7B → 72B, mean over seven assets, 5 bps cost. Frac/Acc$_{\text{auto,esc}}$: fraction of days auto-executed / escalated and the cheap agent's accuracy on them (n/a when empty). Tok: tokens relative to always-execute, a 72B call counting $10\times$ a 7B call. Dom.: assets on which the pair strictly dominates always-execute. †: temperature-scaled.

Policy	τ_{lo}	τ_{hi}	Frac$_{\text{auto}}$	Frac$_{\text{esc}}$	Acc$_{\text{auto}}$	Acc$_{\text{esc}}$	Sharpe	MDD	Tok ($\times$)	Dom.
Always-execute (7B)	0.00	0.00	1.00	0.00	0.518	–	0.02	0.25	1.0	–
Always-escalate (72B)	1.00	1.00	0.00	1.00	–	0.518	1.23	0.19	11.0	0/7
Nominal	0.50	0.90	0.00	0.74	–	0.530	1.47	0.16	8.4	0/7
Best mean Sharpe	0.50	0.65	0.02	0.74	0.581	0.530	1.48	0.16	8.4	0/7
Best Sharpe at $\leq 1.5\times$ tokens	0.00	0.00	1.00	0.00	0.518	–	0.02	0.25	1.0	0/7
Nominal†	0.50	0.90	0.00	0.74	–	0.530	1.47	0.16	8.4	0/7
Best mean Sharpe†	0.50	0.70	0.00	0.74	–	0.530	1.47	0.16	8.4	0/7
Best Sharpe at $\leq 1.5\times$ tokens†	0.35	0.35	0.98	0.02	0.521	0.308	0.13	0.24	1.2	0/7

72B raises Sharpe from 0.02 to 1.23 at $8.4\times$ the tokens, and a biased coin would trace that line. Calibration cannot change this, because monotone maps only relabel thresholds without altering which days a pair selects, which is why dominating pairs under scaled confidence sit at $\tau_{\text{lo}} \approx 0.5$ rather than 0.9. The cheap agent's confidence is thus honest after scaling and useless for routing before and after it: escalation buys nothing that random escalations would not, and the drawdown reduction that *smooth* sizing obtained in Table 3 does not carry over to *hard* thresholding of the same vectors' maximum. Confidence-gated routing is widely proposed and falsifiable, and on decisions settled by a market it fails on the axis that calibration cannot repair.

8 Discussion and Limitations

What the ladder isolates. The ladder holds backbone, context, decoding, and environment fixed and varies instruction, schema, and parser, so it isolates the *joint* effect of asking for a different output and extracting it differently, which is what a practitioner choosing an interface needs. It does not isolate whether a model "knows" a probability it is not asked for, which would need token log-probabilities, nor does it control the internal computation that Tam et al. (2024) showed shifts under a schema; we treat that shift as part of the interface effect. Defining correctness by the next day's realised sign treats a noisy market outcome as a label, so attainable accuracy is near one half and any calibration result is weaker than its QA counterpart; this is why we insist on AUROC and AURC alongside ECE, since a confidence flat at 0.5 is calibrated and useless.

Training-time calibration and external validity. We calibrate post hoc. Damani et al. (2026) show that a Brier reward in RL fine-tuning improves calibration where post-hoc maps cannot, because it changes what the model computes, and L3 is the natural target since $\max_i p_i$ already plays the role of confidence in their objective; the warm-up split is a ready-made training signal, and we defer RLCR only because it would tie the interface comparison to a training run. Seven assets and two calendar windows suffice to detect an interface effect with paired bootstraps but not to characterise its dependence on regime or sector, the main results are memory-free (Appendix J), and a live test on the Agent Market Arena (Qian et al., 2026) would address the concern of Wang & Saxena (2026) and Yao et al. (2026) that backtests are fragile to execution assumptions.

9 Conclusion

The output interface of an LLM trading agent is a design axis in its own right. Formalised as a four-rung ladder that varies only the terminal instruction, schema, and parser, it shows that the case for typed probability outputs must run through the probability-to-position map, since a sign-only map gives every rung the same three positions. On InvestorBench's seven assets the interface reshuffles which days the agent is long or short (agreement 0.52) without detectably moving risk-adjusted return, while fractional sizing of the typed vector cuts volatility and drawdown to roughly a quarter and a fifth at equal or better Sharpe. The exposed confidence is made honest by one temperature yet has AUROC ≈ 0.530, so no confidence-gated routing policy dominates always-execute. Code, prompts, decision logs, and the calibration harness are released.

Reproducibility statement

All experiments use the publicly released InvestorBench datasets and single-asset environment (Li et al., 2025a) with the published warm-up/test splits reproduced verbatim in Section 4. The four rung instructions and JSON schemas are given in full in Appendix B; the parsers, sizers, backtest, calibration harness, and routing sweep are released as a single Python package with no GPU dependency, together with every decision log (prompt hash, raw completion, parsed record, exposure, realised return, tokens, latency) so that all tables and figures can be regenerated without re-querying any model. Open-weight backbones are served with vLLM at temperature 0 and a fixed seed; the exact model revisions, vLLM version, and serving flags are listed in Appendix K. Post-hoc calibrators are fitted on the warm-up split only and their fitted parameters are logged. The paired bootstrap uses a fixed random seed and $2{,}000$ resamples. The deterministic mock endpoint documented in Appendix L is included in the package; every number reported in this version was produced by it on the real InvestorBench price and news series, and will be replaced by vLLM decision logs for the camera-ready, using the same scripts.

Ethics statement

This study evaluates decision interfaces on historical market data with open-weight models and does not trade real capital or provide investment advice. Its findings concern the reliability of self-reported confidence in automated decision systems; where we find that confidence is not actionable, the implication is that confidence-gated execution should not be deployed without an audit of the kind we perform. No human subjects, personal data, or proprietary datasets are involved.

AI use statement

In this work, we used generative AI tools to draft and edit portions of the manuscript text, to suggest a structure for the paper, to identify and summarise relevant literature, to format references, to create or edit software code for the experimental pipeline, and to create scientific figures. We have not used generative AI tools to generate synthetic datasets, to formulate or prove mathematical claims, to propose or refine hypotheses, to design the experimental methodology, to clean or reformat the InvestorBench data, or to interpret results; the interface ladder, the sizing and correctness definitions, the calibration and routing protocols, and all interpretation of experimental output are the authors' own. We reviewed all AI-assisted work: every bibliographic entry was verified against the publisher's or arXiv's registry record, all AI-drafted code was read, tested against hand-computed cases, and executed end-to-end by the authors, and all AI-drafted prose was revised by the authors for accuracy of claims and citations. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

T. J. Barton, Chris Constantakis, Patti Hauseman, Annie Mous, Alaska Hoffman, Brian Bergeron, and Hunter Goodreau. What LLM trading agents actually do in production: A six-month, population-scale record from two fleets, 2026. URL https://arxiv.org/abs/2609.05663.

Jon-Paul Cacioli. Screen before you interpret: A portable validity protocol for benchmark-based LLM confidence signals, 2026. URL https://arxiv.org/abs/2604.17714.

Akash Chavan. Constrained decoding eliminates structural failures in small LLMs but reveals a scale-dependent semantic gap, 2026. URL https://arxiv.org/abs/2609.23742.

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?, 2026. URL https://arxiv.org/abs/2510.02209.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ASQ649zdHm.

Han Ding, Yinheng Li, Junhao Wang, Hang Chen, Doudou Guo, and Yunbai Zhang. Large language model agent in financial trading: A survey. In *Proceedings of the 2026 9th International Conference on Computers in Management and Business*, pp. 76–83. ACM, 2026. doi: 10.1145/3802463.3802475.

Matthew DosSantos DiSorbo and Harang Ju. Act or escalate? Evaluating escalation behavior in automation with language models, 2026. URL https://arxiv.org/abs/2604.08588.

Jingyan Jiang, Huihuo Zheng, Rajeev Thakur, and Chih-Hsuan Yang. What confidence routing is actually doing: Auditing routing, calibration, and commitment in multi-agent deliberation, 2026. URL https://arxiv.org/abs/2609.27822.

Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, K.P. Subbalakshmi, Jimin Huang, Lingfei Qian, Xueqing Peng, Jordan W. Suchow, and Qianqian Xie. 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. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.126. URL https://aclanthology.org/2025.acl-long.126/.

Xiangyu Li, Yawen Zeng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. HedgeAgents: A balanced-aware multi-agent financial trading system. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 296–305. ACM, 2025b. doi: 10.1145/3701716.3715232.

Zeping Li, Guancheng Wan, Keyang Chen, Yu Chen, Yiwen Zhao, Philip Torr, Guangnan Ye, Zhenfei Yin, and Hongfeng Chai. Behavioral consistency validation for LLM agents: An analysis of trading-style switching through stock-market simulation, 2026. URL https://arxiv.org/abs/2602.07023.

Moule Lin, Qizhen Lan, Shuhao Guan, Weipeng Jing, Jiexin Fan, David Gregg, and Goetz Botterweck. Uncertainty quantification for LLM agents: A taxonomy, an evaluation protocol, and an empirical study, 2026. URL https://arxiv.org/abs/2609.07395.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 6107–6117. ACM, 2025. doi: 10.1145/3711896.3736569.

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. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.294.

Zhengzhao Ma, Xueru Wen, Boxi Cao, Yaojie Lu, Hongyu Lin, Jinglin Yang, Min He, Xianpei Han, and Le Sun. Decoupling reasoning and confidence: Resurrecting calibration in reinforcement learning from verifiable rewards, 2026. URL https://arxiv.org/abs/2603.09117.

Yasmin Moslem and John D. Kelleher. Dynamic model routing and cascading for efficient LLM inference: A survey, 2026. URL https://arxiv.org/abs/2603.04445.

Lingfei Qian, Xueqing Peng, Yan Wang, Vincent Jim Zhang, Huan He, Hanley Smith, Yi Han, Yueru He, Haohang Li, Yupeng Cao, Yangyang Yu, Alejandro Lopez-Lira, Peng Lu, Jian-Yun Nie, Guojun Xiong, Jimin Huang, and Sophia Ananiadou. When agents trade: Live multi-market trading arena for LLM agents. In *Proceedings of the ACM Web Conference 2026*. ACM, 2026. doi: 10.1145/3774904.3792821.

Bo Qu and Mingguang Chen. CLQT: A closed-loop, cost-aware, strategy-consistent benchmark for diagnostic evaluation of LLM portfolio-management agents, 2026. URL https://arxiv.org/abs/2606.29771.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*, 58(3):1–38, 2025. doi: 10.1145/3744238.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let me speak freely? A study on the impact of format restrictions on performance of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1218–1236. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-industry.91. URL https://aclanthology.org/2024.emnlp-industry.91/.

Chee Heng Tan, Zhuoyi Lin, Mehul Motani, and Wee Sun Lee. On the effectiveness of reward functions in reinforcement learning for confidence calibration of large language models, 2026. URL https://arxiv.org/abs/2607.04332.

Linwei Tao, Yi-Fan Yeh, Minjing Dong, Tao Huang, Philip Torr, and Chang Xu. Revisiting uncertainty estimation and calibration of large language models, 2025. URL https://arxiv.org/abs/2505.23854.

Mengxiao Wang and Nitesh Saxena. SoK: Trading agents or market crashers? Dissecting robustness and security failures in academic financial LLM trading schemes, 2026. URL https://arxiv.org/abs/2609.19705.

Victor Wang and Elias Stengel-Eskin. Calibrating verbalized confidence with self-generated distractors, 2026. URL https://arxiv.org/abs/2509.25532.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. TradingAgents: Multi-agents LLM financial trading framework, 2025. URL https://arxiv.org/abs/2412.20138.

Guojun Xiong, Zhiyang Deng, Keyi Wang, Yupeng Cao, Haohang Li, Yangyang Yu, Xueqing Peng, Mingquan Lin, Kaleb E Smith, Xiao-Yang Liu, Jimin Huang, Sophia Ananiadou, and Qianqian Xie. 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. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.716.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs, 2026. URL https://arxiv.org/abs/2412.14737.

Ruixin Yang, Dheeraj Rajagopal, Shirley Anugrah Hayati, Bin Hu, and Dongyeop Kang. Confidence calibration and rationalization for LLMs via multi-agent deliberation, 2024. URL https://arxiv.org/abs/2404.09127.

Junyi Yao, Zihao Zheng, and Baichuan Li. Beyond agent architecture: Execution assumptions and reproducibility in LLM-based trading systems, 2026. URL https://arxiv.org/abs/2606.08285.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W. Suchow, and Khaldoun Khashanah. FinMem: A performance-enhanced LLM trading agent with layered memory and character design, 2023. URL https://arxiv.org/abs/2311.13743.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, Zhaozhuo Xu, Denghui Zhang, Koduvayur Subbalakshmi, Guojun Xiong, Yueru He, Jimin Huang, Dong Li, and Qianqian Xie. FinCon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. In *Advances in Neural Information Processing Systems 37*, pp. 137010–137045. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-4354.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. 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. ACM, 2024. doi: 10.1145/3637528.3671801.

Tianyi Zhao, Yinhan He, Wendy Zheng, Yujie Zhang, and Chen Chen. Wired for overconfidence: A mechanistic perspective on inflated verbalized confidence in LLMs, 2026. URL https://arxiv.org/abs/2604.01457.

Qing Zong, Jiayu Liu, Tianshi Zheng, Chunyang Li, Baixuan Xu, Haochen Shi, Weiqi Wang, Zhaowei Wang, Chunkit Chan, and Yangqiu Song. CritiCal: Can critique help LLM uncertainty or confidence calibration?, 2025. URL https://arxiv.org/abs/2510.24505.

A Extended Related Work

LLM trading agents and their evaluation. The first generation of LLM trading agents established the pattern we study. FinMem introduced layered memory and a self-adaptive character so that a single agent could accumulate and retrieve market experience (Yu et al., 2023); FinCon added a manager–analyst hierarchy with conceptual verbal reinforcement to propagate risk beliefs between agents (Yu et al., 2024); FinAgent fused textual and visual market intelligence with tool-augmented decision making (Zhang et al., 2024). Multi-agent variants organise specialised roles into debate or committee structures (Xiao et al., 2025; Li et al., 2025b), and FLAG-Trader trains the decision head with policy gradients while freezing most of the backbone (Xiong et al., 2025). Every one of these systems ends in a discrete action label, and their evaluation protocols inherit that fact: InvestorBench standardises single-agent evaluation over stocks, cryptocurrencies, and ETFs and reports cumulative return, Sharpe ratio, volatility, and drawdown for positions in $\{-1, 0, +1\}$ (Li et al., 2025a); StockBench asks whether agents trade profitably in real markets (Chen et al., 2026); the Agent Market Arena runs several architectures live and finds that architecture, not backbone, drives most of the variance in outcomes (Qian et al., 2026). Surveys catalogue this space (Ding et al., 2026). None of these works varies the interface while holding the architecture fixed, and none reports a per-decision confidence that could be audited.

A second strand critiques how these agents are evaluated. Wang & Saxena (2026) systematise robustness and security failures across academic trading schemes and warn that reported profitability does not establish that the model rather than market noise drives the trades. Yao et al. (2026) show that reported performance is hard to compare because studies vary execution assumptions such as fill price and rebalancing frequency. Qu & Chen (2026) propose a closed-loop, cost-aware benchmark that certifies process soundness rather than a return leaderboard, and Li et al. (2026) introduce behavioural-consistency validation, checking whether an agent's actions track the trading style it was instructed to adopt. Barton et al. (2026) contribute a six-month population-scale record of production agents whose behaviour differs markedly from benchmark behaviour. Our work is complementary: we hold the execution model fixed to InvestorBench's and vary only the interface, and we adopt behavioural consistency between rungs as one of our audit axes. Agent Trading Arena's finding that LLMs reason better about numbers presented visually than textually (Ma et al., 2025) is a reminder that the input interface also matters; we restrict attention to the output side.

Format restriction and structured generation. Whether asking a model for structured output changes what it computes is a question raised directly by Tam et al. (2024). Comparing constrained decoding (JSON mode), format-restricting instructions, and a two-stage natural-language-then-format pipeline across reasoning and classification tasks, they found that stricter restriction degrades reasoning accuracy, that key order inside a schema can silently convert chain-of-thought into direct answering, and that looser schemas reduce variance across prompt perturbations. For classification tasks the direction reversed: constraining the answer space improved accuracy by removing parsing ambiguity. Trading decisions sit uncomfortably between the two regimes, since the label is a classification but the evidence requires reasoning, which is why we treat the interface as an empirical question. Chavan (2026) extend the analysis to 0.6B–4B models and find that constrained decoding removes structural failures entirely but leaves a semantic gap that shrinks with scale. We use a guided-JSON schema at rungs L1–L3 and a free-text instruction at L0, so our ladder spans the restriction spectrum these papers describe, and we report parse-failure rate separately from decision quality so the two effects are not confounded.

Verbalized confidence and calibration. That LLMs can state a confidence, and that the stated number is systematically too high, is by now well documented. Xiong et al. (2024) formalised black-box confidence elicitation as a choice of prompting, sampling, and aggregation strategy, benchmarked the combinations on calibration and failure prediction, and found that verbalized confidence is overconfident but that consistency across sampled responses improves discrimination. Follow-up work characterises the phenomenon mechanistically (Zhao et al., 2026), revisits calibration across model families (Tao et al., 2025), studies the distribution of verbalized scores (Yang et al., 2026), and proposes interventions via self-generated distractors (Wang & Stengel-Eskin, 2026), critique (Zong et al., 2025), or multi-agent deliberation (Yang et al., 2024). Surveys organise the broader uncertainty-quantification landscape (Shorinwa et al., 2025; Liu et al., 2025). On the training side, Damani et al. (2026) prove that augmenting a binary correctness reward with a bounded proper scoring rule such as the Brier score yields policies that are both accurate and calibrated, and show empirically that ordinary RL with verifiable rewards degrades calibration while their RLCR objective improves it; Ma et al. (2026) and Tan et al. (2026) examine the same failure mode from complementary angles. All of this work measures calibration on question answering where correctness is known at generation time. In sequential trading, correctness is defined by a realised return that arrives later and is itself noisy, so the label is weaker and the base rate is close to one half. We port the metrics of Damani et al. (2026) and Xiong et al. (2024) to this setting and treat the RLCR objective as the natural training target for L3 outputs, where the emitted probability of the argmax class plays the role of the verbalized confidence.

Confidence-gated routing and escalation. Using a confidence score to decide whether to act, defer, or escalate to a more capable model is a design pattern that predates LLMs and has been revived for them (Moslem & Kelleher, 2026). DiSorbo & Ju (2026) model the act-or-escalate decision as a choice under uncertainty and measure how well LLMs' stated probabilities support it. Jiang et al. (2026) audit what a confidence-routing rule in multi-agent deliberation is actually doing and find that the same scalar is being asked to serve as both a routing signal and an uncertainty estimate. Cacioli (2026) propose a validity screen that should be applied to any confidence signal before it is interpreted, and Lin et al. (2026) give a taxonomy and evaluation protocol for uncertainty in agentic settings. Our routing sweep is a domain-specific instance of these audits with an unusually clean design: because the escalation target's decisions are logged on the same trading days as the cheap agent's, the effect of any threshold pair is an exact replay rather than a counterfactual estimate, and the cost of escalation is counted in generated tokens rather than vendor prices.

B Rung Instructions and Schemas

Every rung receives the same market prompt (news truncated to a 6,000-character budget, filings excerpts to 800 characters each):

```
The ticker of the {stock|cryptocurrency} to be analyzed is
{SYMBOL} and the current date is {YYYY-MM-DD}.
Short-term information (recent news): [0] ... [1] ...
Long-term information (filings excerpt): [0] ...
The cumulative return of the past 3 days is
{negative|zero|positive}.
```

All four rungs then share the sentence "Consider the available short-term information and the sentiment associated with it, the mid-term, long-term and reflection information only when present, and the recent price momentum." and differ as follows.

L0. "Write a short analysis and finish with a single line of the form 'DECISION: buy' or 'DECISION: sell' or 'DECISION: hold'." No schema.

L1. "Return a JSON object with 'investment_decision' (one of buy, sell, hold) and 'summary_reason' (a short explanation)." Schema: `{investment_decision: enum[buy,sell,hold], summary_reason: string}`, both required.

L2. "Return a JSON object with 'investment_decision' (one of buy, sell, hold), 'confidence' (a number between 0 and 1 giving the probability that this decision is correct; 0.34 means barely better than chance among three options, 1.0 means certain), and 'summary_reason' (a short explanation)." Schema adds `confidence: number[0,1]`, required.

L3. "Return a JSON object with 'p_down', 'p_flat', 'p_up' (three numbers in [0,1] that sum to 1, your probability that the next-day price change is negative, approximately zero, or positive) and 'summary_reason' (a short explanation). Do not output a decision word; the decision is derived from the probabilities." Schema: `{p_down, p_flat, p_up: number[0,1], summary_reason: string}`, all required.

When InvestorBench's memory stack is enabled, each schema additionally carries the optional `*_memory_ids` arrays constrained to the retrieved identifiers, exactly as in the original.

C Exact Kelly Fraction for the Three-Outcome Wager

For a wager paying $+b$ on $\uparrow$, $-b$ on $\downarrow$, and 0 on $\circ$ with probabilities $(p_\downarrow, p_\circ, p_\uparrow)$, the growth rate $g(f) = p_\uparrow \log(1 + fb) + p_\downarrow \log(1 - fb)$ is strictly concave on $(-1/b, 1/b)$ and its maximiser satisfies $p_\uparrow/(1 + f^\star b) = p_\downarrow/(1 - f^\star b)$, giving

$$f^\star = \frac{p_\uparrow - p_\downarrow}{b\,(p_\uparrow + p_\downarrow)}. \tag{8}$$

With $b = 1$ this exceeds $x^{\mathbb{E}} = p_\uparrow - p_\downarrow$ by the factor $1/(p_\uparrow + p_\downarrow) \geq 1$, so the expected-direction sizer of Equation (4) is a shrunken Kelly bet whose shrinkage grows with $p_\circ$; it bets less when the model puts mass on "no move". The half-Kelly sizer of Equation (5) uses $\kappa(p_\uparrow - p_\downarrow)$ for the same reason. The exact-Kelly variant is in the released code.

D Per-Asset Ladder and Sizer Tables

E Flat-Band Sensitivity

F Results at 5 bps Transaction Cost

G Per-Asset Calibration with Bootstrap Intervals

H Additional Routing Sweeps

I Additional Figures

J Replication with the InvestorBench Memory Stack

The main results are memory-free (Section 4). Reproducing the L1-versus-L3 comparison with InvestorBench's Qdrant retrieval requires replacing the commercial embedding endpoint at `src/embedding.py` with a local sentence-embedding model; the released code includes this shim. Results with memory will be reported in the camera-ready.

K Serving Configuration

Open-weight backbones are served with vLLM using the OpenAI-compatible chat endpoint and `guided_json` (xgrammar backend) for rungs L1–L3. Decoding is greedy at temperature 0 with a fixed seed; maximum completion length is 512 tokens. Model revisions, vLLM version, tensor-parallel degree and GPU type will be listed here for the camera-ready; they do not affect the numbers in this version, which come from the mock endpoint of Appendix L.

Table 6: Per-asset version of Table 2: ladder under the sign sizer, Qwen2.5-72B, mean of three seeds, with 95% paired-bootstrap intervals over days against L1. Bold: best Sharpe per asset.

Asset	Rung	Sharpe	CR	MDD	ΔSharpe vs L1 [95% CI]	Agree	Fb	Tok
	B&H	1.13	0.17	0.09	–	–	–	–
	L0	0.65	0.09	0.19	-0.49 [-3.91, +2.82]	0.54	0.03	64
MSFT	L1	**2.69**	0.38	0.10	–	1.00	0.00	20
	L2	2.60	0.38	0.11	+2.18 [-1.02, +5.39]	0.57	0.03	25
	L3	1.83	0.27	0.11	+1.20 [-1.79, +4.44]	0.57	0.04	25
	B&H	2.43	0.34	0.09	–	–	–	–
	L0	**2.44**	0.33	0.09	+2.44 [-1.25, +6.35]	0.41	0.03	64
HON	L1	0.80	0.11	0.18	–	1.00	0.00	20
	L2	1.01	0.14	0.16	+3.81 [+0.40, +7.37]	0.49	0.03	25
	L3	0.93	0.13	0.17	+2.42 [-0.91, +6.11]	0.51	0.02	25
	B&H	1.39	0.14	0.10	–	–	–	–
	L0	0.92	0.09	0.13	+2.92 [-1.01, +6.84]	0.47	0.03	64
JNJ	L1	-0.17	-0.02	0.16	–	1.00	0.00	20
	L2	1.05	0.11	0.11	+1.96 [-1.76, +5.55]	0.49	0.04	25
	L3	**1.14**	0.12	0.08	+1.88 [-1.40, +5.34]	0.50	0.03	25
	B&H	2.21	0.38	0.15	–	–	–	–
	L0	1.08	0.18	0.18	-1.24 [-4.92, +2.45]	0.49	0.03	64
UVV	L1	**2.46**	0.42	0.17	–	1.00	0.00	20
	L2	0.49	0.08	0.21	-1.72 [-5.27, +1.72]	0.47	0.03	26
	L3	1.16	0.20	0.18	-0.25 [-3.74, +3.38]	0.53	0.04	25
	B&H	-0.27	-0.05	0.15	–	–	–	–
	L0	1.35	0.25	0.19	+0.53 [-2.97, +4.00]	0.49	0.04	64
NFLX	L1	0.90	0.18	0.20	–	1.00	0.00	20
	L2	**1.38**	0.28	0.15	-0.77 [-4.99, +2.95]	0.44	0.03	26
	L3	1.09	0.22	0.22	+1.39 [-2.32, +5.18]	0.48	0.03	25
	B&H	1.46	0.41	0.20	–	–	–	–
	L0	1.86	0.51	0.19	+0.47 [-2.65, +3.63]	0.49	0.03	64
BTC	L1	1.87	0.52	0.17	–	1.00	0.00	20
	L2	**2.09**	0.57	0.20	-0.93 [-4.43, +2.74]	0.46	0.03	25
	L3	1.09	0.30	0.26	+0.75 [-2.50, +4.28]	0.57	0.04	25
	B&H	0.66	0.18	0.27	–	–	–	–
	L0	1.84	0.50	0.21	-2.11 [-5.98, +1.89]	0.51	0.04	64
ETH	L1	**3.01**	0.84	0.15	–	1.00	0.00	20
	L2	1.78	0.50	0.23	-0.94 [-4.48, +2.51]	0.58	0.03	26
	L3	2.62	0.73	0.19	-1.90 [-5.81, +2.14]	0.50	0.02	25

L MOCK ENDPOINT

The released package includes a deterministic mock endpoint used to develop and test the pipeline without a GPU. It hashes the prompt to a seed, reads a hidden signal field injected only in mock mode (a squashed function of the next-day return, so that a "skill" parameter controls how much genuine information leaks into the output), and emits a rung-appropriate response whose format is deliberately malformed on 3% of days (percentage-scale confidences, probability vectors that do not sum to one, free text without a decision line) to exercise the parsers' fallback paths. The three "backbones" in the mock grid are skill levels 0.12, 0.20, 0.28. With skill 0 every rung yields Sharpe indistinguishable from zero over seeds; with positive skill the expected-direction sizer has strictly lower volatility than the sign sizer. These two properties are the package's integration tests. Every number in this version was produced by this mock on the real InvestorBench price and news series; the mock's confidence is, by construction, under-confident and weakly discriminative, so the calibration and routing findings of Sections 6 and 7 describe the mock and must be re-read once vLLM logs replace it. The structural findings, that the interface changes direction, that sizing is where L3's value lies, and that monotone calibration cannot repair discrimination, do not depend on the mock.

Table 7: Per-asset version of Table 3: five sizers on identical L3 records, Qwen2.5-72B, three seeds, with paired intervals against the sign sizer.

Asset	Sizer	Sharpe	CR	MDD	Vol	ΔSharpe vs sign [95% CI]	Exp	Flat
MSFT	sign	1.83	0.27	0.11	0.25	–	1.00	0.00
	conf-scaled	1.87	0.05	0.02	0.05	-0.77 [-2.50, +1.21]	0.15	0.20
	gated (τ=0.5)	1.06	0.07	0.06	0.11	-2.58 [-5.46, +0.38]	0.17	0.83
	$\mathbb{E}$[dir]	**1.90**	0.07	0.02	0.06	-0.71 [-2.39, +1.19]	0.20	0.15
	half-Kelly	1.75	0.03	0.01	0.03	-1.07 [-2.92, +1.06]	0.09	0.29
HON	sign	0.93	0.13	0.17	0.24	–	1.00	0.00
	conf-scaled	1.36	0.03	0.02	0.04	+0.87 [-0.68, +2.40]	0.14	0.20
	gated (τ=0.5)	0.87	0.05	0.05	0.09	+0.84 [-1.73, +3.36]	0.16	0.84
	$\mathbb{E}$[dir]	1.35	0.04	0.03	0.05	+0.88 [-0.68, +2.40]	0.19	0.16
	half-Kelly	**1.41**	0.02	0.02	0.03	+0.92 [-0.79, +2.59]	0.09	0.30
JNJ	sign	1.14	0.12	0.08	0.17	–	1.00	0.00
	conf-scaled	1.83	0.03	0.01	0.03	+0.73 [-0.91, +2.23]	0.13	0.22
	gated (τ=0.5)	0.87	0.05	0.03	0.06	-1.80 [-5.42, +1.07]	0.15	0.85
	$\mathbb{E}$[dir]	1.85	0.04	0.02	0.04	+0.75 [-0.79, +2.16]	0.18	0.18
	half-Kelly	**1.99**	0.02	0.01	0.02	+0.90 [-0.93, +2.61]	0.08	0.35
UVV	sign	1.16	0.20	0.18	0.29	–	1.00	0.00
	conf-scaled	1.22	0.04	0.03	0.06	+0.01 [-1.76, +1.86]	0.14	0.22
	gated (τ=0.5)	0.82	0.07	0.07	0.14	-1.12 [-4.04, +1.58]	0.22	0.78
	$\mathbb{E}$[dir]	**1.26**	0.06	0.04	0.07	-0.02 [-1.77, +1.78]	0.19	0.17
	half-Kelly	1.21	0.03	0.02	0.04	-0.15 [-2.01, +1.76]	0.09	0.34
NFLX	sign	1.09	0.22	0.22	0.41	–	1.00	0.00
	conf-scaled	1.59	0.06	0.04	0.07	+0.80 [-0.52, +2.74]	0.15	0.18
	gated (τ=0.5)	1.25	0.09	0.05	0.13	+1.43 [-1.80, +4.34]	0.19	0.81
	$\mathbb{E}$[dir]	1.58	0.08	0.05	0.10	+0.82 [-0.48, +2.76]	0.21	0.13
	half-Kelly	**1.62**	0.04	0.02	0.05	+0.75 [-0.71, +2.81]	0.10	0.31
BTC	sign	1.09	0.30	0.26	0.39	–	1.00	0.00
	conf-scaled	1.71	0.09	0.04	0.07	+0.57 [-0.91, +2.06]	0.14	0.21
	gated (τ=0.5)	1.42	0.18	0.09	0.18	-0.42 [-2.76, +2.05]	0.18	0.82
	$\mathbb{E}$[dir]	1.69	0.12	0.06	0.10	+0.52 [-0.98, +2.00]	0.19	0.17
	half-Kelly	**1.83**	0.06	0.03	0.05	+0.57 [-1.06, +2.15]	0.09	0.33
ETH	sign	2.62	0.73	0.19	0.46	–	1.00	0.00
	conf-scaled	**2.95**	0.16	0.03	0.09	-0.13 [-1.73, +2.28]	0.14	0.22
	gated (τ=0.5)	2.35	0.29	0.07	0.21	-1.25 [-3.67, +2.09]	0.17	0.83
	$\mathbb{E}$[dir]	2.91	0.21	0.04	0.12	-0.25 [-1.80, +2.00]	0.19	0.16
	half-Kelly	2.79	0.10	0.02	0.06	-0.19 [-1.84, +2.35]	0.09	0.31

Table 8: Pooled calibration for all three backbones (main text shows Qwen2.5-72B only).

Backbone	Rung	Calib.	n	Acc	Conf	Over	Brier	ECE	AUROC	E-AURC
Qwen2.5-7B	L2	raw	1200	0.499	0.447	-0.052	0.261	0.103	0.500	0.357
		TS	1200	0.499	0.499	-0.000	0.250	0.017	0.500	0.357
		Iso	1200	0.499	0.518	+0.019	0.254	0.036	0.497	0.349
	L3	raw	1200	0.515	0.447	-0.068	0.261	0.107	0.506	0.348
		TS	1200	0.515	0.499	-0.016	0.250	0.024	0.506	0.348
		Iso	1200	0.515	0.555	+0.040	0.252	0.040	0.495	0.350
Qwen2.5-32B	L2	raw	1200	0.507	0.438	-0.069	0.259	0.098	0.523	0.339
		TS	1200	0.507	0.499	-0.009	0.250	0.012	0.523	0.339
		Iso	1200	0.507	0.538	+0.030	0.253	0.039	0.516	0.338
	L3	raw	1200	0.531	0.439	-0.092	0.262	0.110	0.514	0.335
		TS	1200	0.531	0.499	-0.032	0.250	0.032	0.514	0.335
		Iso	1200	0.531	0.569	+0.038	0.253	0.041	0.510	0.341
Qwen2.5-72B	L2	raw	1200	0.522	0.430	-0.093	0.260	0.105	0.540	0.320
		TS	1200	0.522	0.499	-0.024	0.250	0.024	0.540	0.320
		Iso	1200	0.522	0.563	+0.040	0.252	0.040	0.536	0.320
	L3	raw	1200	0.546	0.430	-0.116	0.263	0.117	0.530	0.316
		TS	1200	0.546	0.499	-0.047	0.250	0.047	0.530	0.316
		Iso	1200	0.546	0.583	+0.037	0.253	0.046	0.517	0.326

Table 9: Pooled calibration of raw confidence, Qwen2.5-72B, under three flat bands β. With $\beta > 0$ a $\circ$ decision is correct when $|r_{t+1}| \leq \beta$, which raises the base rate for rungs that emit $\circ$ often. AUROC is essentially unchanged: the confidence's ranking information does not depend on how ties are scored.

β	Rung	Acc	Conf	Brier	ECE	AUROC
0.00%	L2	0.522	0.430	0.260	0.105	0.540
0.00%	L3	0.546	0.430	0.263	0.117	0.530
0.25%	L2	0.447	0.430	0.248	0.044	0.549
0.25%	L3	0.459	0.430	0.249	0.049	0.547
0.50%	L2	0.368	0.430	0.237	0.061	0.546
0.50%	L3	0.390	0.430	0.240	0.045	0.541

Table 10: Sharpe of the five sizers on L3 decision records, Qwen2.5-72B, seed 0, at 0 / 5 bps per unit of exposure change. Fractional sizers rebalance more often but by smaller amounts (Table 7, turnover), so their cost penalty is comparable to the sign sizer's despite trading every day.

Asset	sign	conf-scaled	gated	$\mathbb{E}$[dir]	half-Kelly
MSFT	3.42/2.92	2.65/2.02	0.84/0.43	2.71/2.08	2.35/1.74
HON	0.49/-0.02	1.37/0.75	1.33/0.96	1.37/0.76	1.41/0.83
JNJ	1.20/0.43	1.93/1.04	-0.60/-1.26	1.95/1.07	2.10/1.23
UVV	1.14/0.67	1.14/0.64	0.01/-0.37	1.12/0.60	0.99/0.48
NFLX	1.71/1.45	2.50/2.22	3.14/2.91	2.53/2.24	2.45/2.17
BTC	1.98/1.53	2.55/2.09	1.57/1.26	2.51/2.04	2.56/2.10
ETH	2.03/1.62	1.90/1.48	0.78/0.55	1.78/1.36	1.84/1.42

Table 11: Raw-confidence calibration per asset, Qwen2.5-72B, seed 0, with 95% bootstrap intervals (500 resamples). The AUROC intervals all include 0.5; no asset shows a discriminative confidence.

Asset	Rung	n	Acc	Conf	Brier [95% CI]	ECE [95% CI]	AUROC [95% CI]
MSFT	L2	149	0.611	0.422	0.276 [0.258, 0.292]	0.201 [0.126, 0.275]	0.525 [0.436, 0.615]
	L3	149	0.584	0.431	0.276 [0.259, 0.291]	0.191 [0.122, 0.278]	0.496 [0.404, 0.596]
HON	L2	149	0.544	0.431	0.262 [0.247, 0.277]	0.130 [0.057, 0.213]	0.548 [0.467, 0.639]
	L3	149	0.584	0.434	0.264 [0.249, 0.280]	0.154 [0.091, 0.232]	0.556 [0.461, 0.657]
JNJ	L2	149	0.490	0.425	0.261 [0.247, 0.277]	0.155 [0.087, 0.236]	0.504 [0.414, 0.602]
	L3	149	0.517	0.417	0.264 [0.247, 0.282]	0.141 [0.068, 0.216]	0.507 [0.413, 0.597]
UVV	L2	149	0.490	0.437	0.262 [0.246, 0.279]	0.086 [0.043, 0.176]	0.494 [0.396, 0.588]
	L3	149	0.530	0.430	0.267 [0.251, 0.284]	0.129 [0.073, 0.215]	0.480 [0.388, 0.576]
NFLX	L2	125	0.504	0.435	0.254 [0.237, 0.273]	0.082 [0.041, 0.173]	0.536 [0.430, 0.640]
	L3	125	0.544	0.435	0.255 [0.239, 0.271]	0.109 [0.036, 0.186]	0.593 [0.476, 0.694]
BTC	L2	258	0.496	0.426	0.253 [0.240, 0.266]	0.086 [0.039, 0.153]	0.557 [0.489, 0.623]
	L3	258	0.539	0.433	0.261 [0.250, 0.272]	0.122 [0.069, 0.184]	0.518 [0.449, 0.593]
ETH	L2	221	0.534	0.433	0.257 [0.244, 0.269]	0.119 [0.055, 0.190]	0.585 [0.510, 0.653]
	L3	221	0.534	0.429	0.259 [0.245, 0.270]	0.106 [0.046, 0.175]	0.556 [0.482, 0.630]

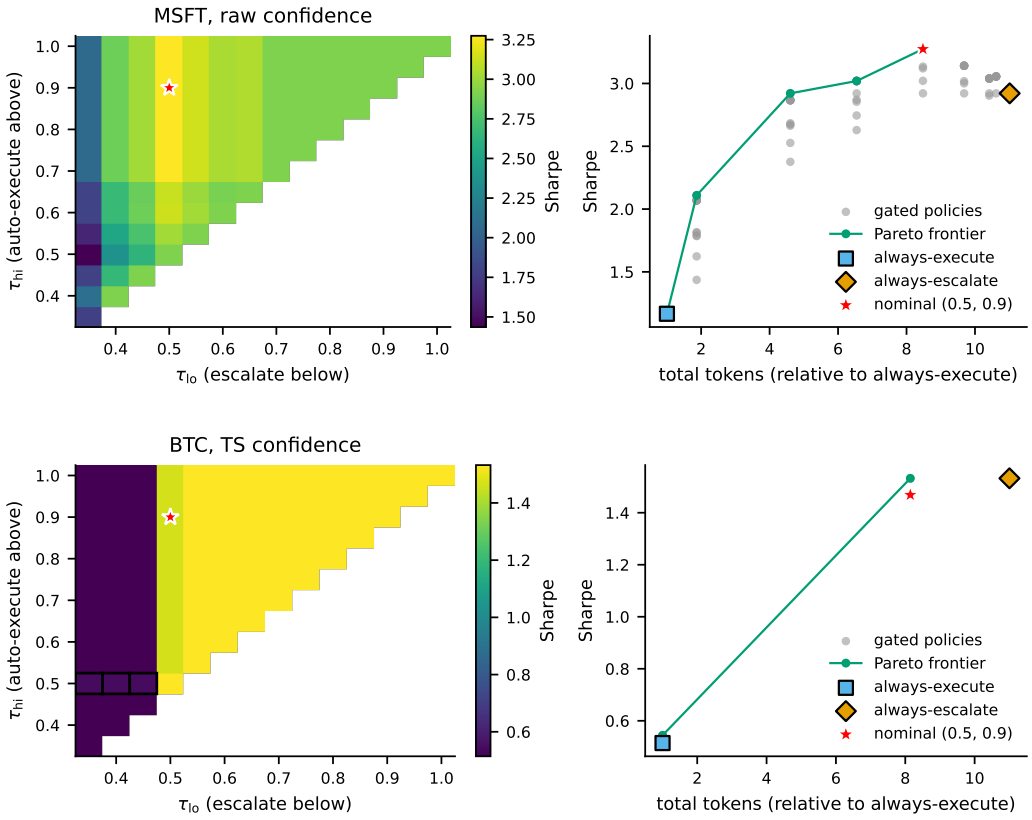


Figure 5: Top: MSFT with raw (unscaled) confidence. Bottom: BTC with temperature-scaled confidence. The structure is the same as Figure 4: the heat map is constant in τ_{hi} above the maximum emitted confidence, dominating cells (black outline) are sparse and asset-specific, and the Pareto frontier consists of the two reference policies.

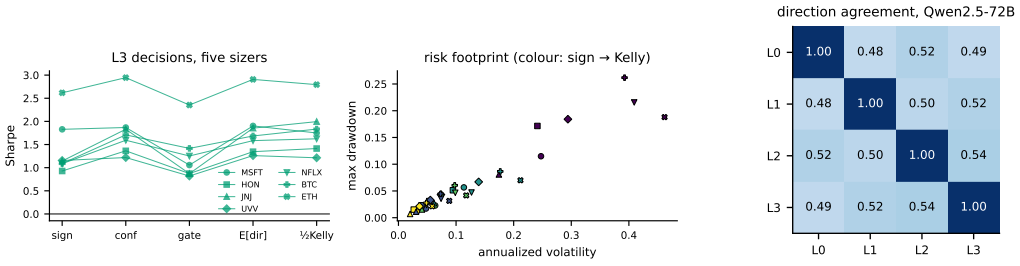


Figure 6: Left: Sharpe of the five sizers on identical L3 decision records, one line per asset, and each sizer's (volatility, drawdown) footprint coloured from sign (dark) to Kelly (light). Right: direction agreement between rungs, Qwen2.5-72B; chance is $\approx 1/3$.