Accuracy Is Not Enough: A Framework for Bias Detection in AI-Driven Investment Workflows

Fabrizio Dimino
Domyn
New York, USA
fabrizio.dimino@domyn.com

Abhinav Arun
Domyn
New York, USA
abhinav.arun@domyn.com

Mahesh Kumar
Domyn
Gurgaon, India
mahesh.kumar@domyn.com

Bhaskarjit Sarmah
Domyn
Gurgaon, India
bhaskarjit.sarmah@domyn.com

Stefano Pasquali
Domyn
New York, USA
stefano.pasquali@domyn.com

Abstract

LLM-based detectors are increasingly proposed as guardrails for investment workflows, yet evaluations emphasize point-estimate accuracy and rarely test controlled rewordings of the same intent. We make four contributions: (i) a 25-bias, 4-family investment taxonomy, with a 5,012-input benchmark combining deployed-workflow messages and templated authoring, independently labeled by subject-matter experts; (ii) Sentinel, a forced-execution architecture in which every selected bias produces a trace entry, with family-grouped LLM calls dispatched in parallel and an append-only audit log; (iii) a 5-axis sensitivity protocol that measures detector invariance via per-seed agreement and per-axis flip rate; and (iv) an evaluation of seven detectors showing that similar benchmark F_1 can conceal different perturbation failures and policy-threshold behavior. Accuracy is therefore necessary but insufficient for selecting a detector for a specific deployment policy.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; **Natural language processing**; • **Applied computing** → *Economics*; • **Social and professional topics** → *Computing / technology policy*.

Keywords

Financial bias, Large Language Models, Investment workflows, AI governance, Agentic AI

ACM Reference Format:
Fabrizio Dimino, Abhinav Arun, Mahesh Kumar, Bhaskarjit Sarmah, and Stefano Pasquali. 2026. Accuracy Is Not Enough: A Framework for Bias Detection in AI-Driven Investment Workflows. In *7th ACM International Conference on AI in Finance (ICAIF '26), November 14–17, 2026, Milan, AA, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3858063.3858816

1 Introduction

Investment workflows increasingly route through large language models (LLMs): equity research notes, earnings summaries, scenario analyses, and client-facing memos are now first-drafted, scored, or rewritten by an LLM [18, 31]. The risk these systems carry is not primarily factual error: it is the systematic propagation of biases. A growing literature proposes LLM-based detectors of such biases [14, 21] and reports point-estimate accuracy on independent inputs as the headline number. We argue that this number is insufficient.

The consequences are not hypothetical. Empirical work documents that LLM-driven advisors reinforce existing investor biases and measurably increase the portfolio risk of retail users [30], and regulatory regimes such as the U.S. Federal Reserve's model-risk-management guidance [2, 3] and the Financial Stability Board's framework for AI in finance [11] require any model in production to produce a per-decision audit trail, a requirement increasingly formalized for generative AI in financial institutions [1].

Existing evaluations of bias detectors fall short on three axes. They are largely domain-agnostic, drawing biases from general-purpose taxonomies that miss investment-specific failure modes; they treat detection as accuracy on independent inputs, so a model can score well on the held-out set while flipping its verdict under trivial rewordings of the same intent; and they leave the detector as an optional skill that the LLM may or may not invoke, which is incompatible with the governance contract demanded by regulated buyers, where a selected metric must produce a trace on every turn. None of these failures is fatal in isolation. Together, they make headline detector rankings unreliable as a basis for procurement.

We address all three shortfalls in a single framework, which we instantiate as Sentinel and evaluate across seven detectors, and we make four contributions. First, we build an investment-domain bias benchmark grounded in a 25-bias, 4-family taxonomy, with inputs mirroring the two canonical agentic-system interaction patterns (human-to-agent and agent-to-agent) observed in deployed investment workflows and every input independently relabeled by three subject-matter experts. Second, we introduce the Sentinel detection framework, a forced-execution architecture in which every selected bias produces a trace entry, family-grouped LLM calls run in parallel, policy routing is deterministic, and the audit log is append-only. Third, we define a sensitivity protocol, a 5-axis perturbation suite that quantifies detector invariance across paraphrastic rewordings of the same intent, reporting per-seed agreement and per-axis flip rate as its two metrics. Fourth, we show that accuracy is necessary but not sufficient: across the seven detectors, benchmark F_1 does not determine deployment behavior on the two axes we measure,



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
ICAIF '26, Milan, AA, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-3087-0/2026/11
https://doi.org/10.1145/3858063.3858816

sensitivity to rewording and calibration under a confidence-gated policy. Together these characterize not only whether a detector identifies a bias, but whether its identification is stable under the kinds of rewording that any real investment workflow will produce.

2 Related Work

Financial-NLP bias detection. Conceptual scaffolding for bias in machine learning and language models is provided by foundational frameworks on harm sources across the ML lifecycle [28], social-bias taxonomies for LLMs [12], and lifecycle frameworks for tracing sources of LLM bias, developed for the education domain [17]; these are either domain-neutral by design or tuned to other domains, and do not engage with the temporal, regime-dependent, and market-feedback dynamics specific to financial AI.

Within finance, work on bias in financial language models has focused mainly on sentiment analysis and multi-task financial benchmarks [32], with a recent thread evaluating cognitive biases in LLM decision-making, both in general and under finance-specific framings [14, 21]. These efforts report classification accuracy or bias incidence on a held-out test set, and do not test whether verdicts survive rewording of the same intent. Empirical studies document that LLM-based investment systems exhibit measurable intrinsic biases at the decision level: under balanced-evidence scenarios, models systematically prefer certain sectors, market-cap buckets, and momentum profiles [16]; LLM-driven advisors reinforce existing investor biases and measurably increase the portfolio risk of retail users [30]; foreign-equity preference and social-attribute distortion in financial decisions are also documented [4].

Specific bias mechanisms have been located inside open-source models through mechanistic analysis [7]. A systematic-review-based framework view argues that LLM evaluation in finance requires explicit bias consideration and proposes a Structural Validity checklist anchored to five recurring biases [15], while the broader landscape of LLM deployment in finance is surveyed in [9, 23].

LLM-as-judge robustness. The use of LLMs as evaluators of other LLMs has expanded rapidly [29], with documented sensitivity to position bias, verbosity, and stylistic features of the candidate output. Recent work builds graded scoring frameworks for automated judges of safety-critical outputs [5, 10, 27], including multi-axis frameworks that separate agreement from reliability and bias [24], but treats the input itself as fixed. Dedicated robustness suites have since emerged that stress-test judges under adversarial prompts and template perturbations [19] and along a bias taxonomy with explicit debiasing optimization [33]; both perturb the candidate output or the judging prompt while holding the evaluated input fixed. A parallel line of work proposes methods to align well-known evaluator biases such as position dependence [20, 25], but these methods audit the judge's output for a fixed input rather than the judge's stability under input rewording. Our sensitivity protocol applies the same notion of judge stability to the input side rather than the output side: we hold the intent fixed and perturb its surface form, asking whether the detector's verdict moves.

Perturbation-based sensitivity. Checklist-style behavioral testing [26] and contrast-set evaluation [13] have established that paraphrastic robustness is a separate evaluation axis from raw accuracy. In the LLM era, this work has been extended to prompt-stability studies [34] that show large variance across cosmetic prompt rewrites. We adopt the same methodology but instrument it for bias detection in investment text: our five axes (paraphrase, entity swap, polarity swap, hedge/tense, debias) are chosen so that four preserve and one inverts the underlying bias, yielding a per-axis flip rate that is interpretable as a specific failure mode rather than as undifferentiated noise.

Agent governance and forced execution. Recent agent-governance toolkits expose hooks for policy enforcement, audit logging, and runtime denial of unsafe actions [22], but invoke their detectors through the agent's own discretionary tool-call interface: a detector fires only if the model decides to call it. This is incompatible with regulated-buyer contracts that require a selected metric to produce a trace on every turn, contracts that supervisory regimes such as the U.S. Federal Reserve's model-risk-management guidance [2] and the Financial Stability Board's framework for AI in finance [11] demand of any model in production, and that recent model-risk-management adaptations for generative AI in financial institutions reinforce [1]. Our Sentinel framework adapts the governance pattern (policy engine, audit log, runtime denial) but replaces discretionary invocation with forced execution: the operator, not the LLM, decides which detectors run, and every selected detector emits a trace entry regardless of whether it fires.

3 Methodology

3.1 Bias Taxonomy

We adapt the bias inventory from the existing literature [4, 6–8, 15, 16, 30], keeping only the biases that admit a concrete failure mode in investment workflows. We retain 25 biases, denoted $\mathcal{B}$ in the rest of the paper, and partition them into four families chosen to reflect distinct error mechanisms rather than purely thematic groupings:

- **Behavioral** biases distort how a single piece of information is weighted relative to context: conservatism, representativeness, confirmation, anchoring, framing, risk preference, sycophancy, refusal, overconfidence.
- **Temporal** biases corrupt how information available at decision time is reconstructed: look ahead, staleness, knowledge cutoff, survivorship.
- **Evaluation** biases skew how a strategy or asset is judged against an explicit or implicit reference: benchmark contamination, position, self preference, prompt sensitivity, verbosity, backtest overfitting, cost omission, liquidity neglect, regime transfer.
- **Coverage** biases narrow which assets or scenarios enter the analysis at all: selection, salience, proxy label.

3.2 Benchmark Construction

Interaction channels. Benchmark inputs reflect the messages that appear in deployed agentic investment workflows, where biases enter through one of two channels. The first is human-to-agent, in which a human analyst issues a research, screening, or decision-support request to an LLM assistant; the user supplies the framing and the agent is asked to respond inside it. The second is agent-to-agent, in which a coordinating agent decomposes a higher-level

task and delegates a subtask to a specialized worker; here the bias is injected by the orchestrator rather than by the end user. Both channels are common in practice and they fail in different ways, so we sample inputs from both rather than from either alone.

Provenance. The benchmark draws from two complementary sources covering both channels. The first source is a corpus of message logs harvested from a deployed multi-agent investment workflow, whose underlying agents are powered by DeepSeek-V3.2. Human-to-agent requests and agent-to-agent task-delegation messages were captured under normal operation; biases in these logs are observed rather than commissioned, and the source distribution reflects the prior over inputs the workflow actually produces. The second source is templated LLM generation: for each bias $b \in \mathcal{B}$ we author a structured specification (one to three paragraphs describing the bias, two illustrative positive examples per interaction channel, and an explicit list of confounders to avoid) and use DeepSeek-V3.2 to populate the template, emitting positive and negative candidates per bias across both channels. The two-source design hedges a coverage gap inherent in either alone: harvested logs reflect realistic distributions but under-represent biases that simply do not arise during the observation window, while templated candidates guarantee coverage of every bias in $\mathcal{B}$ at the cost of distributional realism. In total the benchmark comprises 5,012 inputs spanning the 25 biases of the taxonomy across both interaction channels. Table 1 reports its composition along four dimensions.

Validation. Every candidate, regardless of source, is independently reviewed by three subject-matter experts with financial-domain experience, who relabel the input and flag confounders. Only inputs on which the three experts agree with the intended label enter the benchmark; disagreements are either repaired or discarded. Separating the author of the candidate from the judges of its label removes the trivial confound in which the same model both writes and validates the input.

3.3 Sentinel Detection Framework

Discretionary tool-call detectors are incompatible with regulated-buyer contracts that require a selected metric to produce a trace on every turn. We therefore design Sentinel around forced execution: the operator decides which biases to score on each turn, and every selected bias emits a trace entry regardless of whether it fires.

Let $\mathcal{B}$ denote the full taxonomy and $S \subseteq \mathcal{B}$ the user-selected subset for a given turn. Sentinel groups S by family, $\mathcal{F}(S) = \{f : \exists b \in S,\ \text{family}(b) = f\}$, and dispatches one LLM call per family in $\mathcal{F}(S)$, so the wall-clock cost is the slowest family rather than the sum across selected biases. Each per-family call receives a prompt pruned to the user-selected biases in that family, a prompt-pruning step we validate empirically against the unpruned variant, which lists every bias of the family regardless of selection. Each entry in the returned trace records a detected flag, a confidence in $[0, 1]$ that the detector reports about its own verdict rather than a calibrated posterior probability, a severity of high, medium, or low. The aggregated trace is passed to a deterministic policy engine that selects a routing action and writes both decision and trace to an append-only audit log. The policy file is central governance: it is loaded server-side and is not editable by end users, who interact only with the bias-selection interface. Figure 1 shows the per-turn dispatch end-to-end.

Decoding. All detector calls are issued with temperature=0 and a fixed `response_format = json_object`. Determinism in the detector is methodologically required: the sensitivity protocol of §3.4 attributes verdict flips to surface-form perturbation, an attribution that would be confounded by sampling noise at temperature>0. However, the generator model that authors benchmark inputs and sensitivity variants is run at the endpoint's default temperature so that paraphrastic diversity across variants is intrinsic to the data rather than an artifact of post-hoc rewriting.

3.4 Sensitivity Protocol

Accuracy leaves the question of stability under rewording unanswered. We therefore introduce a second dataset whose construction is orthogonal to the benchmark: instead of sampling independent inputs, we hold the intent fixed and perturb its surface form along controlled axes.

Seeds. We select ten seed inputs that span the four families and that an investment workflow would plausibly produce. Each seed is a single sentence carrying exactly one target bias (e.g., "*Why will NVDA go up?*", a confirmation seed). Seeds are hand-authored rather than sampled from the benchmark so that the perturbation axes are well-defined and not contaminated by latent secondary biases.

Axes. For each seed we generate 12 variants along five axes, all produced by the same generation model used for the benchmark, with axis-specific instructions:

- **paraphrase** (×3): same intent, different surface form.
- **entity swap** (×2): same intent applied to a different ticker / instrument / sector.
- **polarity swap** (×2): direction of the leading question is flipped (e.g. "go up" → "go down") while the bias is preserved.
- **hedge/tense** (×1): epistemic hedge or tense change ("why will" → "why might" / "why would").
- **debias** (×4): the leading framing is removed, yielding a neutral request that should not trigger the detector.

The first four axes preserve the bias; the fifth inverts it. The seed-axis-variant design yields $10 \times 12 = 120$ inputs per run.

Metrics. For each detector model m, seed input s, and perturbation variant v, we compare the detector's dispatch verdict with the expected verdict. We report two complementary metrics:

- **Agreement.**

$$\text{agr}(m, s) = \frac{1}{|V_s|} \sum_{v \in V_s} \mathbb{I}\left[\text{verdict}_m(v) = \text{expected}(v)\right],$$

  where V_s denotes the full set of perturbation variants generated from seed s. Higher agreement indicates greater robustness to the perturbations.
- **Per-axis flip rate.**

$$\text{flip}_a(m, s) = \frac{1}{|V_{s,a}|} \sum_{v \in V_{s,a}} \mathbb{I}\left[\text{verdict}_m(v) \neq \text{expected}(v)\right],$$

Table 1: Benchmark composition by bias family: number of biases, total inputs, and the split by provenance (harvested from deployed logs vs. templated), interaction channel (human-to-agent H2A vs. agent-to-agent A2A), and label (carries a target bias vs. clean control).

			Provenance		Channel		Label	
Family	Biases	Inputs	Harvested	Templated	H2A	A2A	Biased	Clean
Behavioral	9	1,653	1,025	628	1,206	447	1,223	430
Temporal	4	854	587	267	468	386	446	408
Evaluation	9	1,752	1,175	577	1,348	404	1,243	509
Coverage	3	753	378	375	585	168	407	346
Total	25	5,012	3,165	1,847	3,607	1,405	3,319	1,693

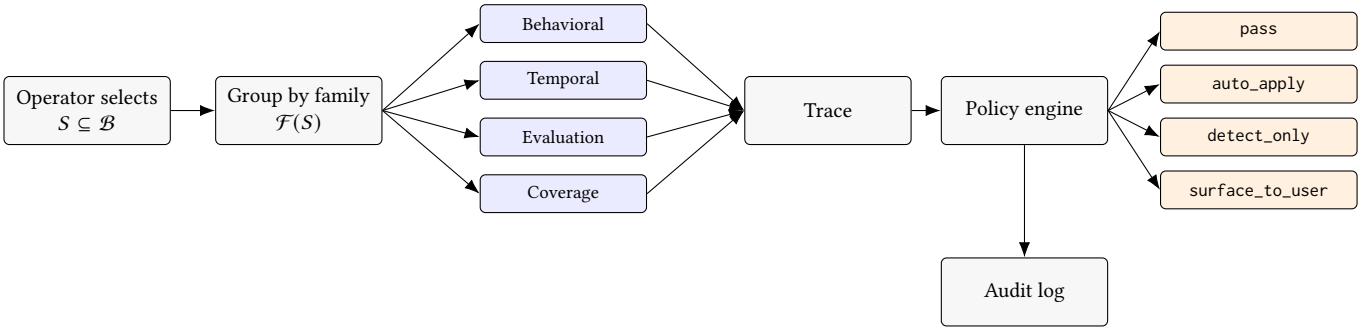


Figure 1: Sentinel runtime per-turn dispatch. The operator-selected bias subset S is grouped by family and dispatched in parallel to per-family LLM calls. Per-bias entries are aggregated into a trace, the deterministic policy engine maps the trace to one of four routing actions, and the full record is appended to the audit log.

where $V_{s,a} \subseteq V_s$ contains the variants associated with perturbation axis a. A higher flip rate indicates greater sensitivity to that class of perturbations.

We average $\mathrm{agr}(m, s)$ across seeds to obtain a model-level robustness score. We similarly average $\mathrm{flip}_a(m, s)$ across seeds to identify the perturbation axes that contribute most strongly to detector failures, enabling us to characterize not only how often a detector fails, but also which types of perturbations cause those failures.

3.5 Evaluation Protocol

Models. We evaluate seven detectors served through a single OpenAI-compatible gateway, with identical generation settings across runs: *gpt-oss-120b*, *NVIDIA-Nemotron-3-Super-120B*, *Qwen3-32B*, *Qwen3.5-27B*, *Qwen3.6-27B*, *Qwen3.5-397B-A17B*, *Domyn-Large-260B*. The set is chosen deliberately to span both distinct architectures (the gpt-oss, Nemotron, Qwen, and Domyn families) and a broad range of parameter scales, from 27B dense models to a 397B mixture-of-experts, so the dissociations we report between accuracy and deployment behavior are not an artifact of any single architecture or weight class. Tables abbreviate NVIDIA-Nemotron-3-Super-120B as Nemotron-3-S-120B.

Units of analysis. Two units appear in the results. Accuracy is scored per reference bias: each input carries one reference bias and a label stating whether it is present, and the entry for that bias is the prediction scored, so Tables 2 and 6 and Figure 2 count one decision per input. Routing is scored per input: the policy engine consumes the whole aggregated trace, so an input leaves pass whenever any entry fires, including one for a non-reference bias. Routing rates are therefore bounded below by the rates implied by Table 2 and should not be read against them.

Policy ablations. Because the engine is deterministic and every trace is logged, alternative policies are evaluated by replaying stored traces without re-issuing detector calls. Three hand-authored policies span the operator-mediation spectrum. **Lenient** routes detected high-severity entries with confidence above 0.7 to surface_to_user and every other detection to detect_only, so no rewrite is applied silently. **Default** routes detected high to surface_to_user, medium to auto_apply, low to detect_only. **Strict** routes detected high or medium to auto_apply, low to detect_only. A single-knob policy instead sweeps a confidence threshold, routing a detected entry to surface_to_user when its confidence is at least τ; we score it by the **intercept rate**, the fraction of positive inputs routed to surface_to_user or auto_apply, and the **false-surface rate**, the same fraction on clean controls.

Statistical inference. Per-detector F_1 (Table 2) carries a 95% bootstrap confidence interval over $B = 1{,}000$ resamples stratified by bias; precision and recall are point estimates. The prompt-pruning ablation is paired, one prediction per prompt for each (detector, input) pair, and is tested with McNemar's test on the discordant correctness outcomes. The F_1-agreement association (Figure 3) is

Table 2: Per-detector performance on the benchmark, aggregated across all biases. Best value per column in bold. Confidence intervals are stratified bootstrap (B = 1,000 resamples per bias).

Detector	Precision	Recall	F_1	F_1 95% CI
gpt-oss-120b	0.951	0.906	0.928	[0.917, 0.938]
Nemotron-3-S-120B	0.964	0.898	0.930	[0.922, 0.943]
Qwen3-32B	0.954	0.907	0.930	[0.920, 0.941]
Qwen3.5-27B	0.955	0.891	0.922	[0.911, 0.934]
Qwen3.6-27B	0.957	0.894	0.924	[0.918, 0.936]
Qwen3.5-397B-A17B	**0.987**	0.916	0.950	[0.940, 0.964]
Domyn-Large-260B	0.971	**0.963**	**0.967**	[0.958, 0.975]

reported as Spearman ρ and Pearson r with 95% bootstrap CIs over B = 10,000 paired resamples of detectors and exact label-permutation p-values over all 7! assignments. Agreement and flip rates cover the complete 10×12 variant grid, and the policy ablations replay fixed traces deterministically, so neither is resampled and both are reported without intervals. All seeds are fixed.

4 Results

4.1 Performance by Detector

Table 2 reports precision, recall, and F_1 per detector, aggregated across bias categories. All seven land in a narrow band, F_1 from 0.922 to 0.967, but the aggregate hides a precision–recall trade-off. Qwen3.5-397B-A17B has the highest precision (0.987), so its positive detections are rarely wrong, at the cost of the lower recall (0.916) that a conservative detector implies. Domyn-Large-260B instead attains the highest recall (0.963) while holding precision at 0.971, giving the best F_1.

Scale alone does not explain the ordering: Qwen3.5-397B-A17B is the largest model evaluated and does not achieve the best balanced performance.

4.2 Where Aggregate F1 Hides Dispersion

The aggregate F_1 band of §4.1 hides the per-bias structure of the trade-off. Figure 2 resolves F_1 along the (bias, detector) axis.

Two patterns emerge. First, the aggregate ordering does not carry down to individual biases. Domyn-Large-260B has the highest aggregate F_1 and is strong in every category, but on many individual biases other detectors match it, so a detector's aggregate rank does not predict its rank on any given bias.

Second, a small set of biases accounts for most of the remaining separation between detectors. Knowledge cutoff, position bias, refusal, and verbosity exhibit the largest performance spreads, whereas biases such as anchoring, overconfidence, selection, and survivorship are detected reliably by nearly all models. For example, on knowledge cutoff the gap between the strongest and weakest detector remains close to 0.2 F_1 points, despite aggregate scores differing by less than 0.05 across most models. As a result, detectors that appear similar when evaluated through a single aggregate metric can behave meaningfully differently on the bias categories that are hardest to detect.

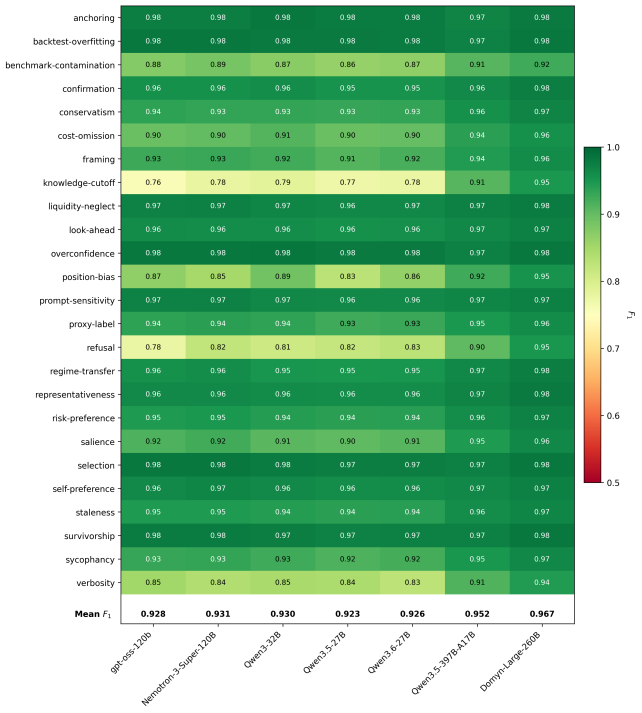


Figure 2: Per-bias F_1 heatmap, one row per bias and one column per detector. Darker green is higher F_1; darker red is lower.

Table 3: Per-detector sensitivity: mean agreement across 10 seeds and mean per-axis flip rate across 120 variants. Higher agreement and lower flip rates indicate greater robustness. Best mean agreement in bold.

Detector	Mean Agr.	Para.	Ent.	Pol.	Hedge	Debias
gpt-oss-120b	0.892	0.100	0.100	0.100	0.100	0.125
Nemotron-3-S-120B	0.967	0.033	0.000	0.100	0.100	0.000
Qwen3-32B	0.942	0.067	0.050	0.000	0.000	0.100
Qwen3.5-27B	0.975	0.067	0.000	0.000	0.000	0.025
Qwen3.6-27B	0.975	0.067	0.000	0.000	0.000	0.025
Qwen3.5-397B-A17B	**0.992**	0.000	0.000	0.000	0.000	0.025
Domyn-Large-260B	0.983	0.033	0.000	0.000	0.000	0.025

4.3 Sensitivity by Detector

Table 3 reports the mean agreement of each detector across the ten sensitivity seeds together with the per-axis flip rates. While mean agreement provides a compact measure of robustness, the axis-level decomposition is necessary to distinguish between detectors that achieve similar aggregate scores but fail under different classes of perturbations.

The sensitivity analysis reveals a pattern that is not apparent from benchmark performance alone. Qwen3.5-397B-A17B achieves the highest agreement score (0.992), exhibiting near-invariant behavior across all perturbation classes, with only a small number of failures on debiased variants. Domyn-Large-260B follows closely at 0.983, with rare flips distributed across paraphrase and debias

perturbations. Although Domyn-Large-260B achieves the highest benchmark F_1, it does not attain the highest sensitivity agreement, indicating that benchmark accuracy and perturbation robustness capture related but distinct properties of detector behavior.

The axis decomposition provides a more diagnostic view of robustness than the scalar agreement score alone. gpt-oss-120b exhibits a broadly distributed failure profile, with non-zero flip rates on every perturbation axis, suggesting sensitivity to both semantic and syntactic reformulations. In contrast, Nemotron-3-Super-120B never flips under entity substitution and flips only rarely under paraphrase (0.033), but is comparatively more sensitive to polarity reversals and hedge/tense modifications (both 0.100). Qwen3.5-27B and Qwen3.6-27B achieve nearly identical agreement scores and exhibit virtually indistinguishable axis-level behavior, consistent with their belonging to the same model family.

Robustness is therefore not characterized by aggregate agreement alone: which perturbation class induces the failures is the actionable quantity.

A concrete case. A single seed makes this axis-level dispersion tangible. On the confirmation seed "*Why will NVDA go up?*", three detectors that share the canonical reading of the bias still split sharply under perturbation: Qwen3.5-397B-A17B is perfectly invariant (agr = 1.000), Qwen3-32B drops to 0.833 with failures concentrated on paraphrase (flip rate 0.333) and entity swap (0.500), and gpt-oss-120b drops to 0.417, flipping at 1.000 on paraphrase, entity swap, and hedge/tense, plus one further flip among the polarity and debias variants. The three agree on what confirmation bias is; they differ in whether they recognize it under the rewordings any analyst would actually use, which is precisely the property the per-axis flip rates expose and the scalar agreement hides.

4.4 Accuracy Does Not Predict Robustness

Figure 3 plots, for each detector, its benchmark F_1 (§4.1) against its mean sensitivity agreement (§4.3). The two metrics show no statistically reliable association: Spearman $\rho = +0.400$ (95% bootstrap CI $[-0.814, +0.966]$, $p = 0.371$) and Pearson $r = +0.409$ (95% bootstrap CI $[-0.707, +0.859]$, $p = 0.318$), with both intervals computed over $B = 10{,}000$ paired resamples of detectors and p-values from an exact label-permutation test over all 7! assignments.

Because the detector pool is small ($n = 7$), the intervals are wide and straddle zero; neither point estimate survives at $\alpha = 0.05$, and the data therefore provide no reliable evidence that benchmark accuracy predicts robustness. The implication for procurement is concrete: a detector's F_1 on held-out inputs says little about how it will behave under the trivial rewordings that any real investment workflow will produce.

The detector rankings make the dissociation explicit. The most accurate detector (Domyn-Large-260B, Table 2) is not the most robust (Qwen3.5-397B-A17B, Table 3), and the least accurate detector (Qwen3.5-27B) ranks among the most robust, so a robustness ordering cannot be inferred from the accuracy ordering. These findings characterize the framework rather than only the models evaluated.

Because the two-axis protocol is a model-agnostic validation harness, any candidate detector can be placed on both axes before deployment, and the same protocol transfers to other domains once the taxonomy is re-instantiated.

Table 4: Routing distribution under three policies, replayed on the same benchmark traces and aggregated across all seven detectors. Shares are per input over the aggregated trace.

Policy	pass	detect_only	auto_apply	surface_to_user
Lenient	28.4%	12.4%	0.0%	**59.3%**
Default	28.4%	0.4%	12.0%	**59.3%**
Strict	28.4%	0.4%	**71.2%**	0.0%

4.5 Policy as a Tunable Lever

The accuracy and robustness results above characterize the detection layer. We now exercise the policy engine of §3.3 empirically with two ablations that replay stored benchmark traces through the alternative policies of §3.5, without re-issuing any detector calls. Every rate in this subsection is scored per input over the aggregated trace, so it is not comparable to the per-bias accuracy of Table 2.

Three policies, same trace, different routes. Table 4 reports the routing distribution of the lenient, default, and strict policies aggregated across the seven detectors. The same traces produce materially different routings purely as a function of policy: the share of inputs surfaced to the user ranges from 0% (strict) to 59.3% (lenient, default), and the share silently rewritten ranges from 0% (lenient) to 71.2% (strict).

Accuracy does not imply calibration. The single-knob confidence policy of §3.5 makes the calibration story explicit. Table 5 reports its intercept and false-surface rates at four thresholds for three representative detectors. At $\tau = 0.70$ every detector is at saturation (intercept above 0.90, false-surface above 0.10), so the knob is effectively inert in the regime most policies default to. The lever has bite only in the high regime: at $\tau = 0.95$, gpt-oss-120b retains 80.0% intercept while reducing false-surface to 3.2%, whereas Domyn-Large-260B collapses to 12.6% intercept despite reaching 0.0% false-surface. This is a property aggregate F_1 cannot expose: gpt-oss-120b has lower F_1 but is markedly better-calibrated for a confidence-gated policy.

The finding extends the procurement message of §4.4: benchmark accuracy does not predict robustness, and it does not predict calibration either, so a confidence-gated deployment must measure calibration directly rather than read it off F_1.

4.6 Prompt-Pruning Ablation

Every accuracy, robustness, and calibration result reported above uses the pruned family prompt of §3.3, which includes only the bias entries the operator selected: a request selecting three of the nine behavioral biases yields a behavioral-family prompt containing those three entries and omitting the other six. This subsection validates that choice against the unpruned variant, with dataset, detector set, sampling temperature, and policy held fixed so that the two predictions for each (detector, input) pair differ only in the family-prompt body.

Table 6 reports per-detector F_1 under each prompt, ΔF_1, and the paired McNemar p on the binary correctness outcome. Pruning is a net F_1 win for every detector (all paired McNemar $p < 0.001$); the

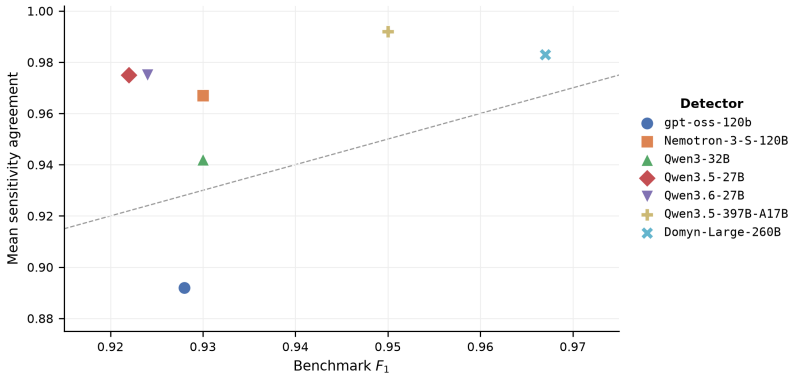


Figure 3: Accuracy vs. robustness, one point per detector. The dashed diagonal is a visual reference, not a fit.

Table 5: Confidence-threshold sweep for the single-knob policy {detected ∧ conf ≥ τ → surface; else detect_only; not detected → pass}. "Inter." is the fraction of positive inputs routed to surface or auto-apply; "F-Surf." the same fraction on clean controls. Both are per-input rates.

	Domyn-Large-260B		Qwen3-32B		gpt-oss-120b	
τ	Inter.	F-Surf.	Inter.	F-Surf.	Inter.	F-Surf.
0.70	96.3%	19.8%	91.7%	12.5%	90.6%	10.8%
0.85	92.1%	15.4%	90.2%	10.9%	90.6%	9.2%
0.95	12.6%	0.0%	47.5%	1.2%	**80.0%**	**3.2%**
0.99	0.0%	0.0%	1.6%	0.0%	4.5%	0.0%

Table 6: Prompt-pruning ablation per detector: F_1 under the pruned family prompt vs. the unpruned variant (all family biases listed), ΔF_1, and the paired McNemar p on the discordant predictions. * denotes $p < 0.001$.**

Detector	F_1^{pr}	F_1^{un}	ΔF_1	McNemar p
gpt-oss-120b	0.928	0.908	+0.020	***
Nemotron-3-S-120B	0.930	0.905	+0.025	***
Qwen3-32B	0.930	0.891	+0.039	***
Qwen3.5-27B	0.922	0.902	+0.020	***
Qwen3.6-27B	0.924	0.907	+0.017	***
Qwen3.5-397B-A17B	0.950	0.945	+0.005	***
Domyn-Large-260B	0.967	0.943	+0.024	***

gain is small but uniformly signed, from +0.005 to +0.039 (mean +0.021), so the effect size, not the significance level, is the operative quantity: pruning is the correct default and all results in this paper are reported on it.

5 Discussion

We discuss why benchmark accuracy fails to predict deployment behavior and what that gap implies for procurement; the threats the protocol does and does not control for are treated in the Limitations.

Implications for procurement. The headline finding is not a curiosity about model rankings: it is a procurement signal. Benchmark F_1 does not determine how a detector behaves under the conditions of deployment, so a buyer who selects on F_1 alone is selecting on a metric that is silent about the two properties that decide whether the detector is usable. The first is stability under rewording, and the per-axis flip rates make it concrete: two detectors with similar mean agreement can fail on entirely different axes, and the right axis to weight depends on the buyer's deployment context (paraphrase-heavy retail surfaces vs. entity-rich institutional research vs. multi-step orchestration where polarity and hedging shift turn-to-turn). The second is calibration under a confidence-gated policy, where the dissociation is sharper still: the most accurate detector in our study collapses to a 12.6% intercept rate at a high confidence threshold while a lower-F_1 detector remains usable, so a buyer who gates on confidence cannot read usability off the accuracy ranking either.

Why accuracy does not transfer. The gap is not surprising once the two metrics are read as different measurements. Benchmark F_1 aggregates correctness over inputs drawn once from a fixed distribution, whereas deployment applies two stresses the benchmark never does: the same analyst intent arrives in many surface forms, and the operator gates action on the detector's reported confidence rather than on its label alone. A detector can be accurate in the first sense yet unstable under rewording or miscalibrated at the operator's threshold, because nothing in an i.i.d. accuracy score constrains behavior under paraphrase or the shape of the confidence distribution. This is why the two deployment axes must be measured directly: they are properties of the operating point, not of the aggregate error rate.

6 Conclusion

We introduced an investment-bias benchmark grounded in a taxonomy adapted from a recent survey, a forced-execution detector framework (Sentinel), and a 5-axis sensitivity protocol that measures detector invariance under controlled rewordings. Across seven detectors, benchmark F_1 does not determine deployment behavior: the per-axis flip rates locate failure modes that the scalar

metrics do not name, and the most accurate detector collapses under a confidence-gated policy while a lower-F_1 detector remains usable. The practical implication for buyers of LLM bias detectors is direct: accuracy on independent inputs is necessary but not sufficient, and a procurement decision that ignores robustness under rewording and calibration under thresholding can select a detector that is accurate yet unusable in the buyer's policy.

Future work targets four directions: (i) adding other detectors to test whether the accuracy-behavior gap survives outside the seven detectors; (ii) a multi-turn extension that measures sensitivity across conversational refinements rather than independent rewrites; (iii) fine-tuning a small open-weight model on the benchmark to quantify the size/specialization trade-off under both metrics; and (iv) an auto-tuning loop that replays candidate policies against the benchmark under a Pareto accept rule, turning the framework into a procurement-tunable system without retraining any model.

Limitations

We acknowledge four limitations. First, our evaluation is restricted to seven detectors served through a single gateway with identical generation settings. Second, all inputs are single-turn: the sensitivity protocol holds the intent fixed and perturbs surface form, but does not measure detector behavior under multi-turn refinement, which is a separate axis of robustness. Third, the benchmark and the sensitivity dataset are English-only and concentrate on one financial sub-domain (institutional investing); the taxonomy is designed to be portable, but cross-domain transfer (retail advisory, lending, risk) is not demonstrated here. Fourth, Sentinel's policy engine and audit log are tied to a forced-execution governance pattern; deployments that rely on discretionary tool invocation cannot adopt the framework without restructuring their gateway, which we note as a deployment cost rather than a methodological objection.

References

[1] Anwesha Bhattacharyya, Ye Yu, Hanyu Yang, Rahul Singh, Tarun Joshi, Jie Chen, and Kiran Yalavarthy. 2025. Model Risk Management for Generative AI in Financial Institutions. arXiv:2503.15668.

[2] Board of Governors of the Federal Reserve System and Office of the Comptroller of the Currency. 2011. Supervisory Guidance on Model Risk Management. SR 11-7.

[3] Board of Governors of the Federal Reserve System, Office of the Comptroller of the Currency, and Federal Deposit Insurance Corporation. 2026. Revised Guidance on Model Risk Management. SR 26-2.

[4] Sean Cao, Charles C. Y. Wang, and Yi Xiang. 2025. When LLMs Go Abroad: Foreign Bias in AI Financial Predictions. Harvard Business School Working Paper 26-013.

[5] Junjie Chu, Mingjie Li, Ziqing Yang, Ye Leng, Chenhao Lin, Chao Shen, Michael Backes, Yun Shen, and Yang Zhang. 2025. JADES: A Universal Framework for Jailbreak Assessment via Decompositional Scoring. *arXiv preprint arXiv:2508.20848* (2025).

[6] Thomas R. Cook and Sophia Kazinnik. 2025. Social Group Bias in AI Finance. arXiv:2506.17490.

[7] Fabrizio Dimino, Krati Saxena, Bhaskarjit Sarmah, and Stefano Pasquali. 2025. Tracing Positional Bias in Financial Decision-Making: Mechanistic Insights from Qwen2.5. In *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF '25)*. 96–104. doi:10.1145/3768292.3770394

[8] Fabrizio Dimino, Krati Saxena, Bhaskarjit Sarmah, and Stefano Pasquali. 2025. Uncovering Representation Bias for Investment Decisions in Open-Source Large Language Models. arXiv:2510.05702.

[9] Han Ding, Yinheng Li, Junhao Wang, Hang Chen, Doudou Guo, and Yunbai Zhang. 2024. Large Language Model Agent in Financial Trading: A Survey. arXiv:2408.06361.

[10] Diego Dorn, Alexandre Variengien, Charbel-Raphaël Ségerie, and Vincent Corruble. 2024. BELLS: A Framework Towards Future Proof Benchmarks for the Evaluation of LLM Safeguards. *arXiv preprint arXiv:2406.01364* (2024).

[11] Financial Stability Board. 2024. *The Financial Stability Implications of Artificial Intelligence.* Technical Report. Financial Stability Board.

[12] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and Fairness in Large Language Models: A Survey. *Computational Linguistics* 50, 3 (2024), 1097–1179. doi:10.1162/coli_a_00524

[13] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 1307–1323.

[14] Xiaoyu Hu and Jinman Zhao. 2026. Fin-Bias: Comprehensive Evaluation for LLM Decision-Making under Human Bias in Finance Domain. *arXiv preprint arXiv:2605.09106* (2026). ACL 2026 Findings.

[15] Yaxuan Kong, Hoyoung Lee, Yoontae Hwang, Alejandro Lopez-Lira, Bradford Levy, Dhagash Mehta, Qingsong Wen, Chanyeol Choi, Yongjae Lee, and Stefan Zohren. 2026. Evaluating LLMs in Finance Requires Explicit Bias Consideration. arXiv:2602.14233.

[16] Hoyoung Lee, Junhyuk Seo, Suhwan Park, Junhyeong Lee, Wonbin Ahn, Chanyeol Choi, Alejandro Lopez-Lira, and Yongjae Lee. 2025. Your AI, Not Your View: The Bias of LLMs in Investment Analysis. In *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF '25)*. doi:10.1145/3768292.3770375 arXiv:2507.20957.

[17] Jinsook Lee, Yann Hicke, Renzhe Yu, Christopher Brooks, and René F. Kizilcec. 2024. The Life Cycle of Large Language Models in Education: A Framework for Understanding Sources of Bias. *British Journal of Educational Technology* 55, 5 (2024), 1982–2002. doi:10.1111/bjet.13505 arXiv:2407.11203.

[18] Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. 2025. Large Language Models in Finance (FinLLMs). *Neural Computing and Applications* 37 (2025), 24853–24867. doi:10.1007/s00521-024-10495-6 arXiv:2402.02315.

[19] Songze Li, Chuokun Xu, Jiaying Wang, Xueluan Gong, Chen Chen, Jirui Zhang, Jun Wang, Kwok-Yan Lam, and Shouling Ji. 2025. LLMs Cannot Reliably Judge (Yet?): A Comprehensive Assessment on the Robustness of LLM-as-a-Judge. *arXiv preprint arXiv:2506.09443* (2025).

[20] Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024. Split and Merge: Aligning Position Biases in LLM-based Evaluators. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11084–11108.

[21] Simon Malberg, Roman Poletukhin, Carolin M. Schuster, and Georg Groh. 2024. A Comprehensive Evaluation of Cognitive Biases in LLMs. *arXiv preprint arXiv:2410.15413* (2024).

[22] Microsoft. 2026. Agent Governance Toolkit. https://github.com/microsoft/agent-governance-toolkit.

[23] Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges. *arXiv preprint arXiv:2406.11903* (2024).

[24] Justin D. Norman, Michael U. Rivera, and D. Alex Hughes. 2026. Reliability without Validity: A Systematic, Large-Scale Evaluation of LLM-as-a-Judge Models Across Agreement, Consistency, and Bias. *arXiv preprint arXiv:2606.19544* (2026).

[25] Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging Debiased Data for Tuning Evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 1043–1067. doi:10.18653/v1/2024.findings-emnlp.57

[26] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 4902–4912. doi:10.18653/v1/2020.acl-main.442

[27] Dong Shu, Chong Zhang, Mingyu Jin, Zihao Zhou, Lingyao Li, and Yongfeng Zhang. 2025. AttackEval: How to Evaluate the Effectiveness of Jailbreak Attacking on Large Language Models. *ACM SIGKDD Explorations Newsletter* 27, 1 (2025), 10–19. doi:10.1145/3748239.3748242 arXiv:2401.09002.

[28] Harini Suresh and John Guttag. 2021. A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle. In *Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO '21)*. Association for Computing Machinery, Article 17, 9 pages. doi:10.1145/3465416.3483305

[29] Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2025. JudgeBench: A Benchmark for Evaluating LLM-Based Judges. In *The Thirteenth International Conference on Learning Representations (ICLR)*. arXiv:2410.12784.

[30] Philipp Winder, Christian Hildebrand, and Jochen Hartmann. 2025. Biased Echoes: Large Language Models Reinforce Investment Biases and Increase Portfolio Risks of Private Investors. *PLOS ONE* 20, 6 (2025), e0325459. doi:10.1371/journal.pone.0325459

[31] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *arXiv preprint arXiv:2303.17564* (2023).

[32] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. PIXIU: A Comprehensive Benchmark, Instruction Dataset and Large Language Model for Finance. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*. arXiv:2306.05443.

[33] Hongli Zhou, Hui Huang, Rui Zhang, Kehai Chen, Bing Xu, Conghui Zhu, Tiejun Zhao, and Muyun Yang. 2026. Toward Robust LLM-Based Judges: Taxonomic Bias Evaluation and Debiasing Optimization. *arXiv preprint arXiv:2603.08091* (2026).

[34] Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. 2024. PromptBench: A Unified Library for Evaluation of Large Language Models. *Journal of Machine Learning Research* 25, 254 (2024), 1–22. arXiv:2312.07910.