

Valid but Wrong: Domain Invariants and a Cross-Model Plan-Validity Benchmark for Household-Finance LLMs

Vignesh Coumarane

InvestEd

vignesh.coumarane1994@gmail.com

Abstract

While large language models today perform well on scoped financial problems, a plan may be type-correct, schema-valid, and executable and still not be financially sound—double-spending the same dollar, or failing to claim free employer-match money. We call such plans *valid but wrong*, and show that current evaluation does not count them. Our contributions are: (1) a validity hierarchy that distinguishes syntactic, schematic, and executional validity from **domain** validity; (2) a concise set of **domain invariants** that operationalize this notion for household finance, split between conservation and efficiency invariants, each rooted in a real six-figure failure in a deployed system; (3) a **cross-model measurement**: we ask eleven current LLMs to generate plans for a balanced scenario bank ($N=585$ surface-valid plans), then use our invariants to evaluate each one. All pass type, schema, and execution checks, yet a nonzero proportion violate a domain invariant undetected by those surface checks. We separate a capturable-match dominance violation (unassailable) from an instruction-following violation (does-it-sum) and a contestable idle-cash violation. The pooled floor for violating unassailable invariants is **15.2%** (95% CI 12–18%), driven by two small models and by instruction-following; the dominance-only rate is 6.2% pooled and **3.6%** across the nine frontier models, whose per-model floors overlap. Even so, the best model (GPT-5) violates no unassailable invariant yet parks cash idle on 41% of plans, a small deployable model fails basic conservation **85%** of the time, and a median \$2,500 of guaranteed employer match is thrown away per dominance violation; the floor holds even on trap-free controls (7.6%). A stricter reading that also counts idle cash—a preference-driven choice—reaches **51.3%** as an upper bound. Capability reduces conservation errors but does not close the domain-validity gap.

1 Introduction

Ask a current model to work a well-posed financial problem and it usually reasons through it, which narrows the interesting question. A household-finance mistake that matters is not a misstep in the arithmetic; it is a plan that looks entirely valid on the surface and is still wrong. It type-checks, it fits the schema, it runs to completion—and it quietly spends the same dollar twice or leaves a guaranteed return on the table.

Two gaps produce this. A *verification gap*: a valid appearance on the surface does not imply validity in the domain, and standard evaluation stops at the surface. And an *input gap*: a plan is only as good as the situation the system gathered, and models seldom ask for the one fact that changes the answer (Su and Cardie, 2026; Takayanagi et al., 2025). Both matter in a domain where a user cannot tell a right answer from a confident one, and where they may be liable for the answer itself: a tribunal held a company responsible for its own assistant’s incorrect statement even when the correct information was available elsewhere (Moffatt, 2024).

We make no claim that this argues against LLMs computing. Capability is high on well-scoped tasks and rising. Our claims are narrower and measurable: there is a class of plans that are valid but wrong, current benchmarks do not count them, and we give evidence that current models produce them at a capability-graded rate. Contributions: a validity hierarchy; a household-finance invariant suite drawn from real failures; and a cross-model measurement of the valid-but-wrong rate, quantified in dollar harm and sensitive to persona.

2 Related work and our delta

Recent work argues component benchmarks do not equal deployment readiness for financial LLM systems (Payzun et al., 2026) and frames deter-

minism as an auditable property across financial workflows (Zhou et al., 2026). We take a next step both leave open: a concrete, domain-specific validity definition for household-finance plans and a measurement of how often it is violated. Closest in spirit, a real-world benchmark finds LLM investment reasoning fluent yet weakly grounded in market events (Jiang et al., 2026), and finance QA degrades sharply when users omit key conditions (Dai et al., 2026); neither defines plan-level domain invariants nor measures a valid-but-wrong rate. We build on verifier-guided generation established in the abstract—program-aided computation routes arithmetic to an executed program (Gao et al., 2023), and the LLM-Modulo view casts the model as a proposer inside an external checker (Kambhampati et al., 2024). Existing executable financial-reasoning benchmarks cover step-level arithmetic (Xie et al., 2025) and statement verification (Panda, 2026), but plan-level domain invariants and a valid-but-wrong rate are still missing. On the input side, models recognize ambiguity yet rarely ask clarifying questions (Su and Cardie, 2026), accurate preference elicitation is decisive for financial-advice quality (Takayanagi et al., 2025), and large-scale evidence shows AI advice quality depends heavily on how users describe themselves (Choukhmane et al., 2026)—which our persona study operationalizes at the level of plan validity. Our contribution is a first executable household-finance benchmark with plan-level domain invariants and a measured valid-but-wrong rate.

3 A validity hierarchy for financial plans

We distinguish four nested levels. A plan is **type-valid** if its fields have the right types; **schema-valid** if it matches the input schema and ranges; **executable** if the engine runs it to completion without error; and **domain-valid** if it also respects the conservation and feasibility laws of household finance. We know how to check the first three, and they are what most pipelines already enforce. A plan can be schema-valid and executable and still be impossible, so the expensive errors live in the valid-but-wrong region: $\{\text{executable}\} \setminus \{\text{domain-valid}\}$. §4 makes it concrete; §5 measures it.

4 The invariant suite

We give a concise set of invariants that operationalize domain validity, split across two layers; each is keyed to a real failure in a deployed system’s

history, not a hypothetical.

Conservation (arithmetic). *Same-money*: in a comparison, both arms deploy identical total cash (a real erratum compared two arms differing by a five-figure sum). *Sources equal deployment*, per loan and in aggregate. $\text{Prepay} \leq \text{outstanding principal}$. $\text{Horizon} \geq \text{required tenure}$.

Efficiency (dominance). No cash is committed to a dominated use when a strictly better one is available in the same plan. Two instances are unambiguous regardless of risk preference: (i) an employer match left unclaimed while cash is deployed elsewhere—a foregone near-100% return; and (ii) cash left idle at 0% while any positive-return use exists. The efficiency form catches the “cash leak,” where a default stranded roughly one fifth of every user’s money at 0% while conservation still held. We claim the mechanism—a dominance condition checked as a pre-execution gate—not the economics of dominance, which is long established.

The instructive point for evaluation: several of these were once “tested” by assertions that re-derived the value they checked, so they passed while six-figure errors shipped. A real invariant is checked on the integration boundary against an independent oracle, not as a local tautology.

5 Cross-model measurement

Setup. We build a balanced bank of 23 household scenarios (debt-vs-invest, multi-loan, employer-match, idle-cash; easy to hard), of which six carry *no* dominance trap (controls) and five have a paraphrased variant. Each states a lump sum, loan rates, any match, and an expected return. We prompt eleven current LLMs to return a dollar allocation in a fixed JSON schema (per-loan prepayment, retirement contribution, taxable investment, idle cash) summing to the lump sum, with no hint toward the answer, twice per prompt. Retaining type-, schema-, and execution-valid plans yields $N=585$, scored by an *independent* reference implementation of the suite (not the deployed engine). We decompose failures into *instruction* (conservation/does-it-sum; arithmetic), *dominance* (a capturable match left unclaimed; unassailable), and *idle* (cash at 0%; contestable). The *floor* counts instruction and dominance only.

Result. All 585 plans pass type, schema, and execution checks, so the surface filter conceals

Model	Floor	95% CI	Strict
GPT-5	0.0%	0.0–6.8	40.7%
Claude Sonnet 4	1.8%	0.3–9.8	44.6%
Claude Opus 4	3.6%	1.0–12.5	39.3%
Mistral Large	3.7%	1.0–12.6	20.4%
Gemini 2.5 Flash	5.4%	1.8–15.0	39.3%
DeepSeek-Chat	5.4%	1.8–15.0	50.0%
Qwen2.5 72B	7.1%	2.7–17.3	58.9%
GPT-4o	8.9%	3.8–19.6	58.9%
GPT-4o-mini	9.4%	4.1–20.2	59.4%
Llama 3.3 70B	37.5%	25.7–51.0	67.9%
Llama 3.1 8B	84.9%	72.9–92.1	88.7%
Pooled	15.2%	12–18	51.3%

Table 1: Domain-invalid rate over $N=585$ surface-valid plans (11 models, 23 scenarios + 5 paraphrase variants, 2 runs). *Floor* = unassailable violations only (conservation, prepay, unclaimed match); *Strict* adds idle cash. *95% CI* is a Wilson interval on the floor ($n \approx 53/\text{model}$, approximate). The nine non-Llama models have overlapping intervals, so we claim no ordering within the frontier; only the two Llama models separate. Sorted by floor.

everything below. Pooled, **15.2%** (95% CI 12–18%) violate the floor and **51.3%** the strict reading. The floor muddles two different errors: instruction-following (does-it-sum) explains about two thirds of it, and the prompt already asks for a sum, so we report the *dominance* rate—a capturable match left unclaimed, which no prompt or preference excuses—as the clean domain-validity number: 6.2% pooled and **3.6%** across the nine non-Llama models. The contrast that remains is coarse, not fine (Table 1): the two small Llama models carry the pooled figure (floor 37.5% and 84.9%, almost entirely by not making the money sum), while the nine stronger models sit between 0.0% and 9.4% with overlapping 95% Wilson intervals, so we claim no ordering within the frontier. What is stable is that even GPT-5, which never fails an unassailable invariant (0.0% floor), still holds idle cash on 40.7% of plans. The failures carry money: among dominance violations, a median **\$2,500** (mean \$3,618) of guaranteed match is discarded, \$130k in aggregate across the run.

The checker is specific, and the gap is not rigged. The checker is a deterministic rule test, so a flagged violation holds by construction; the open question is false positives. On the no-trap control scenarios the dominance invariant false-fires on only 0.8% (a 0.8% false-positive rate), yet the floor there is still 7.6%. It flags a missed match only when one is genuinely capturable, so what remains is a concrete

harm, not a definitional artefact.

Persona sensitivity. Holding the facts fixed and varying only the user’s persona (expert, anxious novice, simplified/non-native English, rambling, neutral) across six scenarios and eight strong models, core validity or match-capture flipped on 10–15% of model–scenario cells (two independent runs). The strongest models are the most robust—GPT-5 and Claude Opus flipped on none of six fact-sets in either run, while other frontier models flipped at most once—and dollar allocations were otherwise stable. We read this as an upper bound, not a demonstrated causal effect: resampling the same prompt already flips validity at a comparable rate, so the persona signal sits within sampling noise, and some flips further reflect a persona that explicitly asks for liquidity (which the efficiency invariant should not penalize). The reliable take-away is robustness—the strongest models do not flip—rather than persona-driven fragility.

Reproducibility. We release the scenario bank, persona templates, the fixed prompt, and the reference checker. An independent re-run on separate hardware reproduced the pooled floor at 16.1% (95% CI 13–19%) and the control false-fire rate at 0.8%; the floor is stable across runs while the strict rate varies more, consistent with idle cash being the one contestable axis.

6 Verification-guided elicitation (future work)

The invariants also indicate which missing input can make a plan invalid: a fact is *decision-critical* if its unknown value can flip an invariant or the feasibility verdict. This suggests ordering intake by value of information under the verifier—asking the match rate before lower-value fields, since it determines the efficiency invariant’s status. We leave a formulation and user study to future work, and note it connects to information-value question selection (Li et al., 2025; Suri et al., 2025). Scoring the deployed system that motivated the invariant suite against this benchmark—which composes plans with a deterministic engine behind the same validator, and so should not enter the valid-but-wrong region by construction—is also left to future work.

7 Conclusion

Capability alone does not close the domain-validity gap in household-finance AI. We separate domain

validity from surface validity, give an invariant suite grounded in real six-figure failures, and quantify a gap that surface checks hide entirely: on the clean dominance reading, 3.6% of the frontier’s surface-valid plans leave a guaranteed employer match unclaimed, the floor persists at 7.6% even on trap-free controls, and a small deployable model fails basic conservation 85% of the time. Within the frontier the per-model differences fall inside overlapping intervals, so the durable claim is not a ranking but that surface validity is not domain validity—and even the strongest model still holds cash idle. The invariant suite and benchmark are the durable artifacts.

Limitations

The measurement uses 23 synthetic scenarios and eleven models at one temperature; coverage is finite and absolute rates would shift with a different mix, though we include no-trap controls, report 95% CIs, and give a per-model breakdown rather than a single number. The floor mixes an instruction-following failure (does-it-sum) with domain violations; we therefore foreground the dominance rate as the clean domain-validity number. The model set spans generations and sizes, so the coarse weak-vs-strong contrast is partly a proxy for vintage and scale, and the per-model floors within the frontier fall inside overlapping Wilson intervals; we make no fine-grained ordering of the models. We report no human or advisor baseline, so the numbers characterize a failure mode rather than an acceptable rate. One reasoning model (Gemini 2.5 Pro) exceeded our output budget before emitting the plan and is excluded. Scoring uses an independent reference implementation of the suite rather than the deployed engine, and the suite is drawn from one deployed system in one regulatory context; it is compact by design, not exhaustive, and may not transfer to markets with different primitives. The strict efficiency reading treats any idle cash as a violation; we report the floor as the headline and strict as an upper bound. The persona and paraphrase studies show variance (about 10%) comparable to same-prompt resampling, so we treat it as within sampling noise rather than a demonstrated intake effect; they do not constitute a full intake-non-determinism study. The efficiency-invariant novelty claim rests on a targeted, non-exhaustive literature search.

Ethical Considerations

The work is education-first and non-advisory: invariants express consequences on a user’s own numbers, not recommendations. Measuring where models produce valid-but-wrong plans is a safety contribution, but published failure modes could be misread as endorsement of unverified LLM financial advice; we frame verification and determinism as requirements, not guarantees of correctness. Scenario data is synthetic; no personal financial data is used. Model outputs are reported in aggregate to characterize a failure mode, not to rank vendors for procurement.

References

- Payzun, B., Demirtas, I., Scala, S., Ferretti, E., and Arslan, S. 2026. Benchmarks Are Not Validation: A System-Level View of Financial LLM Applications. arXiv:2607.28840.
- Choukhmane, T., de Silva, T., et al. 2026. AI Financial Advice: Supply, Demand, and Life Cycle Implications. arXiv:2608.01607.
- Jiang, Y., et al. 2026. Evaluating Investment Logic in Large Language Models: A Real-World Benchmark Towards Personalized Financial Agents. arXiv:2608.06108.
- Dai, Y., Lin, Y., Xie, Z., and Wang, Y. 2026. RealFin: How Well Do LLMs Reason About Finance When Users Leave Things Unsaid? Findings of ACL 2026. arXiv:2602.07096.
- Suri, M., Mathur, P., Lipka, N., Dernoncourt, F., Rossi, R. A., and Manocha, D. 2025. Structured Uncertainty-guided Clarification for LLM Agents. arXiv:2511.08798.
- Xie, Z., et al. 2025. FinChain: A Symbolic Benchmark for Verifiable Chain-of-Thought Financial Reasoning. arXiv:2506.02515.
- Panda, S. 2026. FinVerBench: Benchmark Validity and Calibration in Large Language Model Financial Statement Verification. arXiv:2605.29586.
- Gao, L., et al. 2023. PAL: Program-aided Language Models. ICML 2023.
- Kambhampati, S., et al. 2024. LLMs Can’t Plan, But Can Help Planning in LLM-Modulo Frameworks. ICML 2024.
- Moffatt v. Air Canada, 2024 BCCRT 149.
- Li, B. Z., Kim, B., and Wang, Z. 2025. QuestBench: Can LLMs Ask the Right Question to Acquire Information in Reasoning Tasks? NeurIPS 2025 Datasets and Benchmarks. arXiv:2503.22674.

- Su, J., and Cardie, C. 2026. Knowing but Not Showing: LLMs Recognize Ambiguity but Rarely Ask Clarifying Questions. arXiv:2605.25284.
- Takayanagi, T., et al. 2025. Are Generative AI Agents Effective Personalized Financial Advisors? SIGIR 2025. arXiv:2504.05862.
- Zhou, R., et al. 2026. From Accuracy to Auditability: A Survey of Determinism in Financial AI Systems. arXiv:2605.23955.