Instruction Following for Finance: Verifying language models' ability to follow complex financial instructions

Anonymous Author(s)
Affilation, Address
anon.email@example.org

Abstract

Language Models (LMs) demonstrate an impressive ability to follow instructions, but the risk of hallucination when executing complex, interdependent commands has limited their effectiveness in domains like finance where precision is critical. We introduce `IFF`, a high-difficulty benchmark designed to assess the instruction-following capabilities of LMs for finance. `IFF` provides 88 human-authored prompts that mirror financial analysis tasks and uses a verification system with chainable, verifiable constraints to provide fine-grained reward signals. We evaluate 53 models in a zero-shot setting, including leading proprietary, open-weight, and open-source systems. Our key findings reveal that open-weight models can meet or surpass the instruction-following capabilities of proprietary systems. However, even the top-performing models fail to achieve perfect compliance and struggle with the `IFF` benchmark's complex requirements. We release our dataset and code as an open-source resource to promote research into Reinforcement Learning with Verified Rewards (RLVR) for the financial domain .[1].

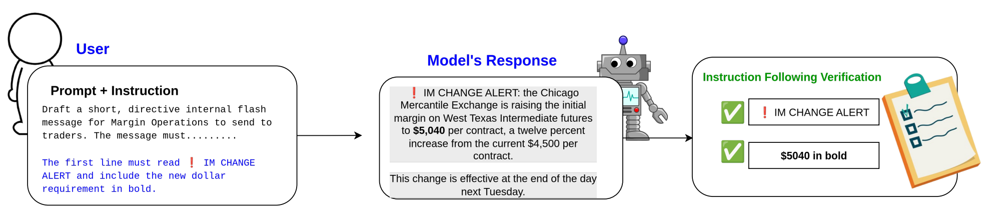


Figure 1: Instruction Following Pipeline.

1 Introduction

The value of Language Models (LM) hinges on their ability to follow instructions precisely while generating responses [1], [2]. While instruction tuning has improved model obedience [3], [4], recent benchmarks reveal a critical gap: LMs still fail to adhere to long or complex instructions that require decomposing interdependent constraints [5], [6], [7].

This gap is particularly acute in high-stakes domains like finance, where workflows combine domain reasoning with strict, machine-checkable constraints; e.g., *"Create a table with exact headers, sort by spread descending, and bold values above a threshold."* In this context, a factually correct answer that violates formatting instructions can cause operational failures. While existing Financial NLP (FinNLP) benchmarks measure task-specific accuracy for tasks like sentiment analysis or QA [8],

[1]The code and data are available for review at: https://anonymous.4open.science/r/iff_anon-443F/

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

[9], [10], [11], they do not evaluate a model's ability to adhere to the complex, multi-part constraints common in financial queries.

To address this gap, we introduce Instruction Following for Finance (`IFF`), a benchmark that stress-tests LMs using 88 human-authored tasks that mirror realistic financial workflows. The benchmark's instructions are designed to be verifiable, covering a wide range of compositional and stylistic constraints such as table structures, conditional formatting, and domain-specific data normalization. Using this framework, we evaluate 53 leading proprietary and open-source LMs in a zero-shot setting, with compliance scored automatically via regex-based verification. We release the entire benchmark,including all prompts, verifiers, and evaluation code as open-source tool to support future research into verifiable-reward training for high-stakes domains.

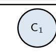
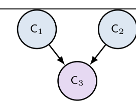
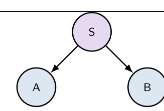
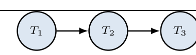
Composition Type	Description	Example	Illustration
Single	The output satisfies a single constraint or requirement	Write a 200-word end-of-day crypto market recap. Include Bitcoin's seven-day realized volatility decline from 45% to 32%.	
And	The output satisfies multiple constraints simultaneously	Write 250 words on Meta's Q3 setup. Include upside levers AND downside watch items AND begin with bold sentence AND add specific headings.	
Selection	The output selects from multiple options based on specified criteria	Update employee trading restricted list. Include only items marked ADD (exclude WATCHLIST and REMOVE).	
Chain	The output completes sequential tasks, each building on the previous	Finance Committee packet: (1) explain budget gap → (2) create table → (3) show bold gap → (4) add bullets → (5) end with source line.	

Figure 2: Composition types in `IFF`.

2 Related work

2.1 Evaluating instruction-following

While instruction tuning methods like InstructGPT [4], Self-Instruct [12], and WizardLM [13] have improved model obedience, evaluating this capability requires more than task-level accuracy. Assessing instruction fidelity demands specialized benchmarks. Frameworks like IFEval [5] and its successors ([14], [15], [16]) were created for this purpose, introducing verifiable, machine-checkable constraints on length, formatting, and keyword inclusion. However, these evaluations are largely domain-agnostic and do not test the specialized constraints required in financial workflows.

2.2 Financial NLP benchmarks

Existing FinNLP benchmarks primarily focus on core NLP tasks including sentiment analysis (FOMC [17]), numerical question answering (FinQA, [18]), and text summarization (SubjECTive-QA, [19]). These benchmarks assess the labeling accuracy of LMs on FinNLP tasks, supporting the development of domain-adapted models. However, none of these benchmarks address the paper's main concern, which is instruction-following. They lack mechanisms to verify compliance with the procedural and formatting constraints common in financial workflows, such as producing tables with exact headers, reporting numerical results to a mandated precision, or including regulatory disclaimers. Consequently, they cannot determine if a model's output is both factually correct **and** compliant with user constraints. `IFF` is designed to fill this critical gap.

3 Methodology

3.1 Instruction-following task

Our methodology, Instruction Following for Finance (`IFF`), is the first instruction-following benchmark designed specifically for finance. We created `IFF` to evaluate this fundamental capability in LMs and to provide a robust environment for Reinforcement Learning with Verified Rewards (RLVR). The benchmark is built on a set of 88 difficult, SME-curated tasks that replicate real-world financial scenarios, prioritizing complexity and breadth over a large volume of simple tasks. Each task contains verifiable output constraints and is designed around four composition types: **single** (a single instruction), **and** (multiple concurrent instructions), **chained** (sequential instructions), and **nested** (conditional instructions) [14]. See Section C for detailed examples.

3.2 Evaluation procedure

We evaluate a wide range of text-only LMs, deferring multimodal evaluation for future work. The models studied include leading proprietary, open-weight, and open-source systems; a complete list is available in Table 1. All experiments were conducted via cloud APIs, using commercial endpoints for proprietary models and services like TogetherAI and Cirrascale for open models.

Following the precedent set by IFEval [5] and IFBench [6], we evaluate outputs using both "strict" and "loose" criteria. The loose evaluation programmatically cleans model outputs to normalize for minor formatting artifacts. Our benchmark instructions test five broad categories of compliance: (1) exact cardinality, (2) table schema, (3) ordering/deduplication, (4) style cues, and (5) word limits.

4 Results

Our findings reveal a clear hierarchy in instruction-following capabilities, with open-weight models leading the `IFF` benchmark. As shown in 1, systems like Llama-4 Maverick 17B surpass the strongest closed-source models. While proprietary systems remain competitive, they no longer define the frontier of performance. Fully open-source models, despite recent progress, continue to lag behind both their open-weight and proprietary counterparts.

Model	Strict	Loose
Closed-source LMs		
Claude 3.7 Sonnet	**65.9**	**70.5**
GPT-4o	63.6	69.3
Claude Opus 4.1	62.5	69.3
Claude Opus 4.0 (2025-05)	60.2	65.9
OpenAI O3	60.2	62.5
GPT-4.1	59.1	61.4
O4-mini	59.1	60.2
O1	58.0	58.0
O3-mini	56.8	56.8
GPT-4o-mini	54.5	58.0
Claude 3.5 Haiku (2024-10)	53.4	61.4
GPT-5 mini	53.4	53.4
Claude 3 Haiku (2024-03)	51.1	54.5
GPT-4.1 nano	47.7	52.3
GPT-5	47.7	50.0
GPT-4.1 mini	46.6	51.1
Claude Sonnet 4 (2025-05)	45.5	52.3
GPT-5 nano	15.9	15.9

Model	Strict	Loose
Open-source LMs		
OLMo-2 32B Instruct	**45.5**	**48.9**
Marin 8B Instruct	37.5	38.6
OLMo-2 13B Instruct	31.8	39.8
OLMo-2 7B Instruct	29.5	36.4
OLMo-2 1B Instruct	14.8	18.2
Open-weight LMs		
Llama-4 Maverick 17B	**76.1**	**79.5**
Llama-3.3 70B (free)	71.6	72.7
Llama-4 Scout 17B	70.5	75.5
Llama-3.3 70B	70.5	71.6
Llama-3.1 405B	67.0	72.7
Cogito v2 70B (prev)	69.3	73.9
Cogito v2 405B (prev)	64.8	67.0
Qwen-QWQ 32B	63.6	64.8
Qwen-3 235B (tput)	61.4	67.0
Moonshot Kimi K2	60.2	63.6
Qwen-2.5 72B	59.1	63.6
DeepSeek-R1 (0528)	59.1	60.2

Model	Strict	Loose
Open-weight LMs		
DeepSeek-v3.1	58.0	59.1
DeepSeek-v3	56.8	60.2
DeepSeek-R1-Distill 70B	55.7	60.2
GPT-OSS 120B	53.4	53.4
GPT-OSS 20B	53.4	53.4
Llama-3 70B Chat-HF	53.4	63.6
DeepSeek-R1-Distill Qwen-14B	50.0	53.4
Mistral-Small 24B (2501)	47.7	51.1
Cogito v2 109B-MoE (prev)	45.5	62.5
Llama-3.1 8B	44.3	48.9
Gemma-3N e4b-IT	42.0	45.5
Cogito v2 DeepSeek-671B (prev)	40.9	54.5
Llama-3 8B Lite	39.8	47.7
Qwen-2.5 7B	35.2	37.5
Mistral-7B v0.3	28.4	34.1
Llama-3.2 3B	28.4	29.5
Mistral-7B v0.2	27.3	34.1
Mistral-7B v0.1	19.3	21.6

Table 1: Model performance on the IFF Benchmark

A surprising result is the relatively weak performance of several models explicitly designed for complex reasoning. Despite their focus on multi-step or conversational tasks, their reliability in our strict evaluation is not markedly stronger than general-purpose models, and in some cases, it is worse. This suggests that current reasoning-oriented tuning does not necessarily translate to more precise instruction following.

Across all model classes, we find that failures often stem from an inability to adhere to precise formatting or structural requirements. The significant gap between "strict" and "loose" scores, particularly for open-source models, indicates that many failures are "near misses" where the semantic content is correct but a specific constraint is violated.

5 Limitations

Our evaluation has two primary limitations. First, we rely on a single response per prompt due to API cost, which limits robustness against stochastic model variation. Second, our regex-based validators, while transparent and reproducible, can be brittle. They may occasionally fail to recognize a semantically correct response due to minor formatting deviations. More flexible validation methods, such as LM-as-a-judge or hybrid semantic checks, are a natural next step for future work.

6 Conclusion

In this study, we introduced the `IFF` Benchmark to assess instruction-following reliability for financial applications. While proprietary systems remain competitive, they are no longer dominant, with top-performing open-weight models now leading in instruction-following capabilities; fully open-source models, however, continue to lag behind. We find that reasoning-oriented models and smaller checkpoints surprisingly underperform, suggesting that scaling or explicit reasoning design alone does not ensure robustness. The observed gap between strict and loose scoring indicates that many failures are "near-misses",outputs that are semantically plausible but procedurally flawed. This highlights that for models to be trusted in finance, the standard for success must be precise, verifiable compliance, not just plausible generation.

References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners.”

[2] H. Touvron *et al.*, “LLaMA: Open and Efficient Foundation Language Models.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2302.13971

[3] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2201.11903

[4] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2203.02155

[5] J. Zhou *et al.*, “Instruction-Following Evaluation for Large Language Models.” Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2311.07911

[6] V. Pyatkin *et al.*, “Generalizing Verifiable Instruction Following.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2507.02833

[7] H. Peng, Y. Qi, X. Wang, B. Xu, L. Hou, and J. Li, “VerIF: Verification Engineering for Reinforcement Learning in Instruction Following.” Accessed: Aug. 13, 2025. [Online]. Available: http://arxiv.org/abs/2506.09942

[8] R. S. Shah *et al.*, “WHEN FLUE MEETS FLANG: Benchmarks and Large Pre-trained Language Model for Financial Domain.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2211.00083

[9] X. Wu *et al.*, “Golden Touchstone: A Comprehensive Bilingual Benchmark for Evaluating Financial Large Language Models.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2411.06272

[10] G. Lu, X. Guo, R. Zhang, W. Zhu, and J. Liu, “BizFinBench: A Business-Driven Real-World Financial Benchmark for Evaluating LLMs.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2505.19457

[11] G. Matlin, M. Okamoto, H. Pardawala, Y. Yang, and S. Chava, “Finance Language Model Evaluation (FLaME).” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2506.15846

[12] Y. Wang *et al.*, “Self-Instruct: Aligning Language Models with Self-Generated Instructions,” May 2023. Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2212.10560

[13] C. Xu *et al.*, “WizardLM: Empowering large pre-trained language models to follow complex instructions,” May 2025. Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2304.12244

[14] B. Wen *et al.*, “Benchmarking Complex Instruction-Following with Multiple Constraints Composition.” Accessed: Aug. 29, 2025. [Online]. Available: http://arxiv.org/abs/2407.03978

[15] S. Chen *et al.*, “Recent Advances in Large Langauge Model Benchmarks against Data Contamination: From Static to Dynamic Evaluation.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2502.17521

[16] Y. Hwang, Y. Kim, J. Koo, T. Kang, H. Bae, and K. Jung, “LLMs can be easily Confused by Instructional Distractions,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds., Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 19483–19496. doi: 10.18653/v1/2025.acl-long.957.

[17] A. Shah, S. Paturi, and S. Chava, “Trillion dollar words: A new financial dataset, task & market analysis,” May 13, 2023. [Online]. Available: http://arxiv.org/abs/2305.07972

[18] Z. Chen *et al.*, “FinQA: A dataset of numerical reasoning over financial data,” presented at the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.300.

[19] H. Pardawala *et al.*, “SubjECTive-QA: Measuring subjectivity in Earnings Call Transcripts' QA through six-dimensional feature analysis,” Oct. 27, 2024. [Online]. Available: http://arxiv.org/abs/2410.20651

[20] Y. Chang *et al.*, “A Survey on Evaluation of Large Language Models,” *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, Jun. 2024, doi: 10.1145/3641289.

[21] B. Wen *et al.*, “Benchmarking Complex Instruction-Following with Multiple Constraints Composition.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2407.03978

[22] Q. He *et al.*, “Can Large Language Models Understand Real-World Complex Instructions?,” 2023, doi: 10.48550/ARXIV.2309.09150.

[23] K. Yan, H. Guo, X. Shi, S. Cao, D. Di, and Z. Li, “CodeIF: Benchmarking the Instruction-Following Capabilities of Large Language Models for Code Generation.” Accessed: Aug. 29, 2025. [Online]. Available: http://arxiv.org/abs/2502.19166

[24] Z. Zhang, J. Chen, and D. Yang, “DARG: Dynamic Evaluation of Large Language Models via Adaptive Reasoning Graph.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2406.17271

[25] J. Heo *et al.*, “Do LLMs "know" internally when they follow instructions?,” 2024, doi: 10.48550/ARXIV.2410.14516.

[26] J. Heo, M. Xiong, C. Heinze-Deml, and J. Narain, “Do LLMs estimate uncertainty well in instruction-following?,” 2024, doi: 10.48550/ARXIV.2410.14582.

[27] V. Reddy, R. Koncel-Kedziorski, V. D. Lai, and C. Tanner, “DocFinQA: A long-context financial reasoning dataset,” Jan. 12, 2024. [Online]. Available: http://arxiv.org/abs/2401.06915

[28] W. Hu, J. Duan, C. Wei, L. Zhang, Y. Zhang, and K. Xu, “DynaCode: A Dynamic Complexity-Aware Code Benchmark for Evaluating Large Language Models in Code Generation.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2503.10452

[29] K. Zhu, J. Wang, Q. Zhao, R. Xu, and X. Xie, “Dynamic Evaluation of Large Language Models by Meta Probing Agents.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2402.14865

[30] K. Zhu, J. Chen, J. Wang, N. Z. Gong, D. Yang, and X. Xie, “DyVal: Dynamic Evaluation of Large Language Models for Reasoning Tasks.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2309.17167

[31] Y. Perlitz *et al.*, “Efficient Benchmarking of Language Models,” arXiv, 2023. doi: 10.48550/ARXIV.2308.11696.

[32] T. Zou, X. Zhang, H. Yu, M. Wang, F. Huang, and Y. Li, “EIFBENCH: Extremely Complex Instruction Following Benchmark for Large Language Models.” Accessed: Aug. 29, 2025. [Online]. Available: http://arxiv.org/abs/2506.08375

[33] Z. Yuan, J. Liu, Q. Zi, M. Liu, X. Peng, and Y. Lou, “Evaluating Instruction-Tuned Large Language Models on Code Comprehension and Generation,” 2023, doi: 10.48550/ARXIV.2308.01240.

[34] K. Kamble, M. Russak, D. Mozolevskyi, M. Ali, M. Russak, and W. AlShikh, “Expect the Unexpected: FailSafe Long Context QA for Finance.” Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2502.06329

[35] Z. Yuan *et al.*, “FinLLMs: A Framework for Financial Reasoning Dataset Generation with Large Language Models.” Accessed: Aug. 29, 2025. [Online]. Available: http://arxiv.org/abs/2401.10744

[36] S. Ye *et al.*, “FLASK: Fine-grained Language Model Evaluation based on Alignment Skill Sets,” 2023, doi: 10.48550/ARXIV.2307.10928.

[37] V. Pyatkin *et al.*, “Generalizing Verifiable Instruction Following.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2507.02833

[38] L. Diao, X. Xu, W. Sun, C. Yang, and Z. Zhang, “GuideBench: Benchmarking Domain-Oriented Guideline Following for LLM Agents,” 2025, doi: 10.48550/ARXIV.2505.11368.

[39] D. Jaroslawicz, B. Whiting, P. Shah, and K. Maamari, “How Many Instructions Can LLMs Follow at Once?.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2507.11538

[40] Y. K. Chia, P. Hong, L. Bing, and S. Poria, “INSTRUCTEVAL: Towards Holistic Evaluation of Instruction-Tuned Large Language Models.” Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2306.04757

[41] S. Li *et al.*, “Instruction-following Evaluation through Verbalizer Manipulation,” arXiv, 2023. doi: 10.48550/ARXIV.2307.10558.

[42] Y. Jiang *et al.*, “Instruction-Tuning Data Synthesis from Scratch via Web Reconstruction.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2504.15573

[43] T. Kim, J. Kim, G. Lee, and S.-y. Yun, “Instructive Decoding: Instruction-Tuned Large Language Models are Self-Refiner from Noisy Instructions,” presented at the International Conference on Learning Representations, Nov. 2023. Accessed: Aug. 13, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Instructive-Decoding%3A-Instruction-Tuned-Large-are-Kim-Kim/c0d698950a4560fc2a63acb30a91aa2deb042ed3

[44] R. Murthy, P. Venkateswaran, P. Kumar, and D. Contractor, “KCIF: Knowledge-Conditioned Instruction Following,” May 2025. Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2410.12972

[45] Y. Hwang, Y. Kim, J. Koo, T. Kang, H. Bae, and K. Jung, “LLMs can be easily Confused by Instructional Distractions.”

[46] Y. He *et al.*, “Multi-IF: Benchmarking LLMs on Multi-Turn and Multilingual Instructions Following.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2410.15553

[47] J. Huang *et al.*, “Open-FinLLMs: Open Multimodal Large Language Models for Financial Applications.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2408.11878

[48] J. Zeng *et al.*, “Order Matters: Investigate the Position Bias in Multi-constraint Instruction Following,” in *Findings of the Association for Computational Linguistics: ACL 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds., Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 12479–12492. doi: 10.18653/v1/2025.findings-acl.646.

[49] J. Zeng *et al.*, “Order Matters: Investigate the Position Bias in Multi-constraint Instruction Following.” Accessed: Aug. 29, 2025. [Online]. Available: http://arxiv.org/abs/2502.17204

[50] Q. Xie *et al.*, “PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2306.05443

[51] R. Vir, S. Shankar, H. Chase, W. Fu-Hinthorn, and A. Parameswaran, “PROMPTEVALS: A Dataset of Assertions and Guardrails for Custom Production Large Language Model Pipelines.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2504.14738

[52] W. Liu *et al.*, “RECAST: Strengthening LLMs' Complex Instruction Following with Constraint-Verifiable Data.” Accessed: Aug. 29, 2025. [Online]. Available: http://arxiv.org/abs/2505.19030

[53] Y. Liu *et al.*, “ReIFE: Re-evaluating Instruction-Following Evaluation.” Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2410.07069

[54] Y. Wang *et al.*, “Self-Instruct: Aligning Language Models with Self-Generated Instructions.” Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2212.10560

[55] J. Cheng *et al.*, “SPaR: Self-Play with Tree-Search Refinement to Improve Instruction-Following in Large Language Models,” Mar. 2025. Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2412.11605

[56] P. Venkateswaran and D. Contractor, “Spotlight Your Instructions: Instruction-following with Dynamic Attention Steering.” Accessed: Aug. 26, 2025. [Online]. Available: http://arxiv.org/abs/2505.12025

[57] Q. Ren *et al.*, “Step-by-Step Mastery: Enhancing Soft Constraint Following Ability of Large Language Models.” Accessed: Aug. 29, 2025. [Online]. Available: http://arxiv.org/abs/2501.04945

[58] H. Li *et al.*, “Synthetic Data (Almost) from Scratch: Generalized Instruction Tuning for Language Models.” Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2402.13064

[59] M. Zheng *et al.*, “TableDreamer: Progressive and Weakness-guided Data Synthesis from Scratch for Table Instruction Tuning,” in *Findings of the Association for Computational Linguistics: ACL 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds., Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 7290–7315. doi: 10.18653/v1/2025.findings-acl.381.

[60] S. Kim *et al.*, “The BiGGen Bench: A Principled Benchmark for Fine-grained Evaluation of Language Models with Language Models,” arXiv, 2024. doi: 10.48550/ARXIV.2406.05761.

[61] X. Chen *et al.*, “The SIFo Benchmark: Investigating the Sequential Instruction Following Ability of Large Language Models.” Accessed: Sep. 01, 2025. [Online]. Available: http://arxiv.org/abs/2406.19999

[62] J. Cook, T. Rocktäschel, J. Foerster, D. Aumiller, and A. Wang, “TICKing All the Boxes: Generated Checklists Improve LLM Evaluation and Generation.” Accessed: Aug. 31, 2025. [Online]. Available: http://arxiv.org/abs/2410.03608

[63] Y. Cao *et al.*, “Toward Generalizable Evaluation in the LLM Era: A Survey Beyond Benchmarks,” 2025, doi: 10.48550/ARXIV.2504.18838.

A Appendix: Models

Table 1: Models evaluated in this study. The table lists all proprietary, open-source, and open-weight models along with their configuration parameters.

Provider	Family	Model	MaxTok	Temp	Top-p
Closed-source LMs					
OpenAI	O-Series	o4-mini	4096	0.20	0.90
OpenAI	O-Series	o3	4096	0.20	0.90
OpenAI	O-Series	o3-mini	4096	0.20	0.90
OpenAI	O-Series	o1	4096	0.20	0.90
OpenAI	GPT-5	gpt-5	4096	1.00	n/a
OpenAI	GPT-5	gpt-5-mini	4096	1.00	n/a
OpenAI	GPT-5	gpt-5-nano	4096	1.00	n/a
OpenAI	GPT-4.1	gpt-4.1	4096	0.20	0.90
OpenAI	GPT-4.1	gpt-4.1-mini	4096	0.20	0.90
OpenAI	GPT-4.1	gpt-4.1-nano	4096	0.20	0.90
OpenAI	GPT-4o	gpt-4o	4096	0.20	0.90
OpenAI	GPT-4o	gpt-4o-mini	4096	0.20	0.90
Anthropic	Claude Opus	claude-opus-4-1-20250805	4096	n/a	n/a
Anthropic	Claude Opus	claude-opus-4-20250514	4096	0.20	0.90
Anthropic	Claude Sonnet	claude-sonnet-4-20250514	4096	0.20	0.90
Anthropic	Claude Sonnet	claude-3-7-sonnet-20250219	4096	0.20	0.90
Anthropic	Claude Haiku	claude-3-5-haiku-20241022	4096	0.20	0.90
Anthropic	Claude Haiku	claude-3-haiku-20240307	4096	0.20	0.90
Open-source LMs					
Cirrascale	OLMo 2	OLMo-2-0325-32B-Instruct	2048	0.20	0.90
TogetherAI	Marin	marin-8b-instruct	2048	0.20	0.90
Cirrascale	OLMo 2	OLMo-2-1124-13B-Instruct	2048	0.20	0.90
Cirrascale	OLMo 2	OLMo-2-1124-7B-Instruct	2048	0.20	0.90
Cirrascale	OLMo 2	OLMo-2-0425-1B-Instruct	2048	0.20	0.90
Open-weight LMs					
Together.ai	Meta Llama 4	Llama-Maverick-17B-128E-Instruct-FP8	4096	0.20	0.90
Together.ai	Meta Llama 4	Llama-Scout-17B-16E-Instruct	4096	0.20	0.90
Together.ai	Meta Llama 3	Llama-3.3-70B-Instruct	4096	0.20	0.90
Together.ai	Meta Llama 3	Llama-3.1-405B-Instruct	4096	0.20	0.90
Together.ai	Meta Llama 3	Llama-3.3-70B-Instruct	4096	0.20	0.90
Together.ai	Deep Cogito	cogito-v2-preview-llama-70B	4096	0.20	0.90
Together.ai	Deep Cogito	cogito-v2-preview-llama-405B	4096	0.20	0.90
Together.ai	Qwen	QwQ-32B	4096	0.20	0.90
Together.ai	Qwen	Qwen3-235B-A22B-Instruct-2507	4096	0.20	0.90
Together.ai	Moonshot	Kimi-K2-Instruct	4096	0.20	0.90
Together.ai	Qwen	Qwen2.5-72B-Instruct	4096	0.20	0.90
Together.ai	DeepSeek	DeepSeek-R1-0528	4096	0.20	0.90
Together.ai	DeepSeek	DeepSeek-V3.1	4096	0.20	0.90
Together.ai	DeepSeek	DeepSeek-V3	4096	0.20	0.90
Together.ai	DeepSeek	DeepSeek-R1-Distill-Llama-70B	4096	0.20	0.90
Together.ai	GPT-OSS	gpt-oss-120b	4096	0.20	0.90
Together.ai	GPT-OSS	gpt-oss-20b	4096	0.20	0.90
Together.ai	Meta Llama 3	Llama-3-70b-chat-hf	4096	0.20	0.90
Together.ai	DeepSeek	DeepSeek-R1-Distill-Qwen-14B	4096	0.20	0.90
Together.ai	Mistral AI	Mistral-Small-24B-Instruct-2501	4096	0.20	0.90
Together.ai	Deep Cogito	cogito-v2-preview-llama-109B-MoE	4096	0.20	0.90
Together.ai	Meta Llama 3.1	Meta-Llama-3.1-8B-Instruct	4096	0.20	0.90
Together.ai	Google Gemma	gemma-3n-E4B-it	4096	0.20	0.90
Together.ai	Deep Cogito	cogito-v2-preview-deepseek-671b	4096	0.20	0.90
Together.ai	Meta Llama 3	Meta-Llama-3-8B-Instruct-Lite	4096	0.20	0.90
Together.ai	Qwen 2.5	Qwen2.5-7B-Instruct	4096	0.20	0.90
Together.ai	Mistral AI	Mistral-7B-Instruct-v0.3	4096	0.20	0.90
Together.ai	Meta Llama 3.2	Llama-3.2-3B-Instruct	4096	0.20	0.90
Together.ai	Mistral AI	Mistral-7B-Instruct-v0.2	4096	0.20	0.90
Together.ai	Mistral AI	Mistral-7B-Instruct-v0.1	4096	0.20	0.90

B Appendix: Evaluation Framework

B.1 Architecture

We designed the IFF framework to use a modular, layered architecture to support maintainability and reproducibility. The application layer exposes a user interface through command-line scripts. The primary entry points are evaluation_bin.py for running evaluations and generate_responses.py for producing model outputs. The business logic layer implements the core functionality. The evaluation_lib.py module orchestrates end-to-end evaluation, and the instruction modules define and resolve the logic for each verifiable constraint. The utility layer provides shared functions required by the business logic. The instructions_util.py module implements text processing and validation routines such as word counting, sentence splitting, and table detection that are reused by multiple checkers. The data layer holds the artifacts consumed and produced by the system, including input JSONL files with prompts and output JSONL files containing model responses and evaluation results.

B.2 Core Components & Data Flow

Benchmark construction begins with build_input_jsonl, which assembles test cases using instructions_registry to combine instruction types and parameters into financial prompts. The prompts are serialized to an input JSONL file. The generate_responses module reads this file and submits the prompts to external models through a multi-provider gateway. The evaluation phase is initiated by evaluation_bin, which uses evaluation_lib to load prompts and their corresponding responses. For each prompt, evaluation_lib requests the appropriate checker from instructions_registry and applies the selected evaluation mode to the model output. The instructions_registry maintains the mapping from instruction identifiers to checker classes and returns the correct implementation at runtime. The finance_instructions module contains the concrete checkers, each inheriting from a common base and implementing the domain-specific rules required to validate its constraint. The instructions_util module supplies the supporting text-processing functions that are reused across checkers. The overall data flow is linear and reproducible: prompt generation produces the input JSONL, response generation produces the responses JSONL, and evaluation produces result files that record pass/fail outcomes and statistics.

B.3 Design Patterns

The registry pattern appears in instructions_registry, which centralizes the association between instruction identifiers and checker implementations so that the evaluation engine does not depend on concrete classes. The strategy pattern governs evaluation behavior, allowing Strict and Loose modes to be selected at runtime without changing the engine's control flow. The factory pattern is used when instantiating checkers from identifiers in a prompt; evaluation_lib requests a class from the registry, constructs the checker with any provided parameters, and executes validation. This combination allows new instructions to be added by registering a checker class, without modifying orchestration code or other components.

C Appendix: Evaluation Dataset

This appendix provides detailed information on the prompts and instructions used in our experiments, along with samples of model outputs.

C.1 System Prompt

The LLMs are primed using the following system prompt to aid in aligning with our instruction following and improve performance.

```
System prompt
```

You write for institutional finance. Use ASCII punctuation only: straight quotes ("), apostrophes ('), hyphen (-), percent (%), commas, periods, colons, semicolons, parentheses.

OUTPUT CONTRACT
- Follow the user's instructions exactly, including any Markdown (bold, italic), headings (#/##/###), fenced code blocks using triple backticks, tables in plain Markdown, and checkboxes like "[ ] ".
- Return only the requested content. No preambles, no meta-explanations, no extra commentary.

DEFAULTS
- Do not invent data. Preserve given tokens exactly (numbers, dates, casing, punctuation, separators).
- Do not introduce any formatting (headings, lists, tables, code blocks, HTML, links, images, footnotes, blockquotes) unless explicitly asked.
- Do not reorder items unless the instruction explicitly asks for a different order.
- Whitespace hygiene: no leading/trailing spaces on any line. No extra blank lines beyond what is explicitly requested. Do not add a trailing newline unless requested.

COUNTS • SCOPE • VOCAB
- When a count is given (lines, rows, bullets, checkboxes, occurrences), satisfy it exactly.
- Apply scope exactly where specified (e.g., "inside Section B only" vs "across the entire output") and nowhere else.
- When an allowed vocabulary is given, use only those tokens, case-sensitively, with no substitutions or synonyms.

DELIMITERS
- When delimiters/sentinels are specified (e.g., [FORM] ... [/FORM]), print them exactly, in the exact positions requested, and nowhere else.

SEPARATORS
- When the instruction specifies separators (e.g., "; " or ", "), reproduce them exactly (including spacing). Do not normalize or trim.

VERBATIM COPYING
- When asked to copy text from a base/source, copy it verbatim including punctuation and spacing. Do not paraphrase.

CONFLICTS
- If global and section-specific rules conflict, the section-specific rule governs within its section.
- If two rules conflict in the same scope, prefer the stricter interpretation (fewer items, fewer occurrences, fewer formats).

MARKDOWN PRIMITIVES (only when explicitly requested)
- TABLE: Render as a GitHub-style Markdown table:
  1) one header row with the exact column names,
  2) one separator row of hyphens and pipes (no alignment colons),
  3) the exact number of data rows requested.
  Do not add/remove columns, pad with extra spaces, or reorder rows/columns.
- BULLETED LIST: Use "- " for each bullet, one item per line, in the order requested.
- ORDERED LIST: Use "1.", "2.", ... one item per line, in the order requested. Do not auto-renumber unless asked.
- ITALICS: Use *...* only (no _, no mixing).
- BOLD: Use **...** only (no __).
- FONT COLOR: Use tags to color. For eg. 2025.
- CHECKBOXES: Use "[ ] " for unchecked or "[x] " for checked exactly as requested (case-sensitive "x").
- CODE BLOCKS: Use fenced code blocks with triple backticks. If a language is specified, put it immediately after the opening backticks (e.g., '''json). Emit exactly the number of code blocks requested.
  • JSON code block: contain exactly one valid JSON object unless otherwise specified; no comments or trailing commas.

```
- HEADINGS: Use "#", "##", "###", etc. only if explicitly requested, with the
wording provided.

BLANK-LINE RULE
- Unless a specific blank-line pattern is requested (e.g., "one blank line between
Block 1 and Block 2"), produce no blank lines. When a pattern is requested, follow
it exactly and avoid any additional blank lines.
```

C.2 Sample Prompts

Representative examples of prompts are provided to illustrate the variety and structure of inputs used.

C.2.1 Single Output Prompts

```

Prompt
Rates quants want a 150-word explainer of how the SOFR–TONA basis feeds into
cross-currency-swap valuation. TL;DR: the basis spread is added to one leg so both
discount curves align before fair-value logic is applied. Close with the
adjustment formula.
Instruction: Begin with exact italicised TL;DR sentence above and end with a
fenced LaTeX formula block showing the basis-spread adjustment.
```

C.2.2 Chained Output Prompts

```

Prompt
The bank is scheduling a release of the core payments switch identified as release
twenty twenty five point zero nine. The change window runs from Saturday twelve
September two thousand twenty five at twenty two hundred hours until Sunday
thirteen September two thousand twenty five at zero four hundred hours local time.
The update affects instant bank transfers, automated clearing house files, and
card authorisation routing. The contingency plan is to return to the prior
production version within thirty minutes if specific failure indicators appear.
Prepare a cutover runbook that coordinates technology, operations, and
communications. Instruction: Begin with a one sentence goal statement, then set
out the runbook in order: (1) provide a timeline table with columns time,
activity, owner, and success signal from pre change to post change; (2) write
three checklists titled pre cutover, cutover, and post cutover using Markdown
checkboxes for items such as freezes, backups, message drains, smoke tests, and
reconciliation; (3) add a short section titled rollback triggers that lists the
exact indicators that require a return to the prior version; (4) conclude with a
communications tree naming who is notified at start, at checkpoints, and at
completion. Output in Markdown only.
```

C.3 Sample Responses

This section presents representative outputs of Llama 4 Maverick, including both correct and incorrect responses, to illustrate model performance for both strict and loose mode.

The model produces outputs that fully adhere to the prompt, meeting all specified instructions and correctly following the intended structure and formatting.

```

#### Prompt

Write about 250 words on Meta’s Q3 setup using only the facts below. Include upside
levers (Reels RPM, tighter opex) and downside watch items (foreign exchange drag,
ads macro). Do not invent data.
 Facts: revenue $36.8 billion; EBIT margin 39%.
Begin with one sentence in bold stating your overall call. Add the heading "Upside
Watch" followed by exactly three bullets, then the heading "Downside Watch" followed
by exactly three bullets; each bullet must begin with one of these tokens : Increase,
Tighten, Improve, Monitor, Reduce, Shift, Manage. If your opening bold sentence
contains the word "probably".Insert a single bold line that reads "Confidence: Low."
```

immediately before the final line. Finish with one italic line "Downside risk: FX drag; ads macro.".

Response (PASSED)

Meta is probably well-positioned for Q3 given its historical performance and current trends.
 Upside Watch
- Improve Reels RPM as the format continues to gain traction.
- Tighten opex by optimizing operational efficiencies.
- Increase ad load without compromising user experience.
 Downside Watch
- Monitor foreign exchange fluctuations that could negatively impact revenue.
- Manage ads macro environment that may affect advertiser spending.
- Reduce dependence on a single revenue stream to mitigate potential risks.
Confidence: Low.
Downside risk: FX drag; ads macro.

The example below illustrates a case in which the model does not fully adhere to the prompt, producing outputs that deviate from the specified instructions and formatting requirements. In this instance, the model fails to follow multiple instructions provided in the prompt.

Prompt

Summarize the three deficiencies identified by the model-risk audit in mortgage prepayment model version 5.2. The deficiencies are: an outdated seasoning curve, a missing macroeconomic linkage for refinancing incentive, and an inadequate back-testing window. Present them clearly and concisely, without adding or inventing any further issues.
List the three deficiencies as bullets and end with the remediation deadline 2025-10-31 in bold red text.

Response (FAILED)

- The mortgage prepayment model version 5.2 has an outdated seasoning curve.
- The model lacks a macroeconomic linkage for refinancing incentive.
- The back-testing window used is inadequate.
Remediation deadline: **2025-10-31**

The example below illustrates another case in which the model does adhere to the prompt, again producing outputs that match the specified instructions and formatting.

Prompt

Create a detailed checklist for the loan-closing team, covering the settlement of a seven hundred fifty million-dollar leveraged-loan allocation scheduled for tomorrow on ClearPar. The checklist must be grouped into three sections: Booking, Know Your Customer (KYC), and Cash Settlement. Each step within the checklist should be preceded by an empty tick box written as [ ]. After the checklist, add the sign-off line:
Ops Lead: ______
Follow the group headings exactly and append the sign-off line.

Response (Passed)

Booking

[ ] Verify loan allocation amount of $750,000,000 in ClearPar system
[ ] Confirm settlement date matches tomorrow's business date
[ ] Validate borrower entity details and loan facility information
[ ] Check trade confirmation details against original allocation notice
[ ] Ensure proper booking codes and GL accounts are assigned
[ ] Verify interest rate, spread, and payment frequency parameters
[ ] Confirm loan maturity date and amortization schedule
[ ] Review and approve any fee structures (commitment, facility, agent fees)
[ ] Validate collateral details and security interest documentation

```
[ ] Ensure proper risk weighting and regulatory capital allocation
[ ] Confirm booking system integration with downstream systems
[ ] Generate and review preliminary booking entries

# Know Your Customer (KYC)

[ ] Verify borrower KYC documentation is current and complete
[ ] Confirm beneficial ownership information is up to date
[ ] Check sanctions screening results for all related parties
[ ] Validate anti-money laundering (AML) compliance status
[ ] Review politically exposed person (PEP) screening results
[ ] Confirm customer due diligence (CDD) documentation
[ ] Verify enhanced due diligence (EDD) requirements if applicable
[ ] Check regulatory watch list screening results
[ ] Validate customer risk rating and approval status
[ ] Confirm compliance with know your customer's customer (KYCC) requirements
[ ] Review any adverse media or negative news screening results
[ ] Ensure all KYC exceptions have proper approvals

# Cash Settlement

[ ] Confirm settlement instructions in ClearPar match internal records
[ ] Verify correspondent banking relationships for settlement
[ ] Check available liquidity and funding arrangements
[ ] Validate wire transfer instructions and beneficiary details
[ ] Confirm settlement currency and any FX requirements
[ ] Review nostro account balances for settlement currency
[ ] Verify cut-off times for same-day settlement processing
[ ] Check for any settlement holds or restrictions
[ ] Confirm backup settlement procedures are in place
[ ] Validate settlement reporting requirements
[ ] Ensure proper cash management and liquidity monitoring
[ ] Prepare settlement confirmation and reporting documentation

Ops Lead: ______
```

Selection prompts are effective because they require the model to generate outputs that fit into a rigid structure, leaving little room for ambiguity. This ensures we can clearly evaluate whether the model followed the exact formatting rules and constraints.

```
#### Prompt

FinOps review for Q2-25 cloud spend: AWS Compute $650 k, Storage $280 k, Network
$90 k, Other $120 k (Total $1.14 m); Azure Compute $540 k, Storage $180 k, Network
$60 k, Other $70 k (Total $0.85 m); GCP Compute $720 k, Storage $210 k, Network $70
k, Other $50 k (Total $1.05 m). Build a table by provider and line item, add a Grand
Total row, and flag every quarterly cost above $1 million with ⚠.
Show the Grand Total row and mark each cost > $1 m with ⚠.

#### Response (PASSED)

| Provider | Compute | Storage | Network | Other | Total |
|----------|---------|---------|---------|-------|-------|
| AWS      | $650k   | $280k   | $90k    | $120k | $1.14m ⚠ |
| Azure    | $540k   | $180k   | $60k    | $70k  | $0.85m |
| GCP      | $720k   | $210k   | $70k    | $50k  | $1.05m ⚠ |
| **Grand Total** | **$1.91m ⚠** | **$670k** | **$220k** | **$240k** | **$3.04m
⚠** |
```

C.4 Instruction Registry

This section provides a comprehensive registry of all instructions used.

1. fin:equities_bold_intro_italic_risk
2. fin:credit_table_spread_vs_carry

3. fin:fx_calc_codeblock_limit
4. fin:compliance_rule10b5_numbered
5. fin:ops_settlement_checklist
6. fin:ir_six_bullets_verb_buyback
7. fin:treasury_liquidity_risk_section
8. fin:deriv_black76_latex_sigma
9. fin:risk_var_numbered_boldusd
10. fin:pe_subheaders_dashes
11. fin:quant_pseudocode_comments
12. fin:crypto_recap_percent_italic
13. fin:abs_table_then_comments
14. fin:reit_underline_wordlimit
15. fin:structured_protect_terms
16. fin:ecb_timestamp_boldrates
17. fin:ratings_three_numbered
18. fin:pension_table_footnote
19. fin:margin_im_alert
20. fin:etf_timed_checklist
21. fin:fintech_swot_licence_italic
22. fin:cet1_formula_block_end
23. fin:ma_synergies_arrows
24. fin:esg_scopes_bold
25. fin:tax_carried_interest
26. fin:ifrs17_callout_two
27. fin:retail_liquidity_underline
28. fin:embond_yield_boldprice
29. fin:rrp_drivers_subs_bold
30. fin:climate_grid_critical
31. fin:syndicate_table_boldcover
32. fin:aml_numbered_high
33. fin:xccy_basis_tldr_latex
34. fin:clearpar_groups_signoff
35. fin:modelrisk_deadline
36. fin:carbon_divergences_two
37. fin:clo_table_italic_shortest
38. fin:vc_proscons_underline_terms
39. fin:ndf_numbered_bold_vals
40. fin:lch_margin_timestamp_code
41. fin:marketing_irr_threehashtags
42. fin:project_keyparties_boldusd
43. fin:finops_table_grandtotal_warn
44. fin:payments_iso20022_qa_underline
45. fin:munis_tey_footnote_bold
46. fin:index_rebal_subject_bullets
47. fin:correlations_table_red
48. fin:scf_redflags_callout_limit
49. fin:sdr_csv_italic_note
50. fin:eqderivs_gamma_table_bold
51. fin:muni_budget_gap_snapshot
52. fin:esg_csrd_checklist
53. fin:settlement_runbook_notes
54. fin:ma_board_slide_apextech
55. fin:rmbs_poolcard_stratification
56. fin:fxhedge_calendar_novamed
57. fin:airline_fuel_exposuremap
58. fin:aml_case_northbridge
59. fin:treasury_monthend_runsheet
60. fin:rm_securitization_factsheet

61. fin:pe_quarterly_letter_skeleton
62. fin:claims_reserve_rollforward
63. fin:merchant_chargeback_pack
64. fin:jetfuel_hedgeplan_outline
65. fin:bank_gap_snapshot
66. fin:lockup_monitoring_plan
67. fin:branchclosure_oakview_plan
68. fin:merchant_onboarding_arcadia
69. fin:realestate_valuation_runbook
70. fin:greenbond_allocation_riverton
71. fin:procurement_cards_auditpack
72. fin:research_blueharbor_compliance
73. fin:payments_cutover_runbook
74. fin:venturedebt_covenant_monitor
75. fin:payments_switch_cutover_runbook
76. fin:restricted_list_adds_table
77. fin:fund_profile_fees_contact
78. fin:treasury_accounts_rationalization
79. fin:sec_filing_pack_zephyr
80. fin:esg_extract_orion_2024
81. fin:credit_limit_changes_summary
82. fin:saas_amendment_change_json
83. fin:pci_scope_extract_2025
84. fin:ap_payment_run_2025_10_15
85. fin:tax_1099_candidates_2025
86. fin:saas_quanta_commercial_controls
87. fin:revops_deals_hygiene_q4_2025
88. fin:ir_roadshow_blueharbor_oct2025

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [YES]

   Justification: The abstract and introduction claim to introduce a new benchmark, `IFF`, for financial instruction following, present a large-scale evaluation of modern LMs, and release the associated code and data. These contributions are described in detail in the Methodology section and the results are presented in the Results section. The appendix provides further details on the models, framework, and dataset.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [YES]

   Justification: The paper includes a dedicated "Limitations" section. We discuss that our evaluation relies on a single response sample per model, which may not fully account for stochastic model outputs, and note that this was due to the high API costs of generating multiple responses.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper introduces a new benchmark and reports empirical findings. It does not include any theoretical results, theorems, or proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [YES]

   Justification: We provide detailed descriptions of our evaluation methodology, including the "strict" and "loose" accuracy metrics. The appendix contains the full list of models evaluated and their hyperparameters, a description of the evaluation framework, and representative examples from our benchmark dataset to ensure our results can be reproduced.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way

to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [YES]

Justification: Yes, we are releasing the complete benchmark, including all prompts, instructions, and evaluation code, under an open-source license. For the review process, we provide an anonymized link to the repository. The public URL will be included in the final camera-ready version of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [YES]

Justification: Our study evaluates existing models in a zero-shot setting, so no training was performed. We provide all relevant testing details in the appendix, which includes a table of the models used and the specific inference hyperparameters (e.g., temperature, max tokens) for each one.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NO]

Justification: We report the exact accuracy scores for each model on our benchmark based on a single run. We do not report error bars or conduct statistical significance tests, as our current evaluation is based on a single response generated for each prompt. This is noted as a limitation in our paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NO]

Justification: We do not provide a detailed breakdown of the compute resources used to run our evaluation framework. The majority of the models evaluated were accessed via third-party APIs, meaning the computational load for model inference was handled externally. For the self-hosted open-source models, the required resources are specified in their respective documentation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`

Answer: [YES]

Justification: We have reviewed the NeurIPS Code of Ethics and confirm our research is in full compliance. The benchmark dataset was authored by the research team and does not contain any real-world, sensitive, or personally identifiable information. Our work does not involve human subjects.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NO]

Justification: Our paper focuses on the positive impact of improving model reliability in finance. While we acknowledge the risks of instruction-following failures, we do not include a detailed discussion of potential negative societal impacts or malicious uses of more capable models. Our work is foundational, intended to improve evaluation rather than deploy a new application.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This question is not applicable. Our paper introduces a benchmark dataset of synthetic prompts and evaluation code, not a new generative model or a scraped dataset. The prompts are authored by the research team and do not contain sensitive or private information, posing a low risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NO]

    Justification: We credit the creators of the models we evaluate by citing their original research papers. However, we do not explicitly state the license or terms of use for each model in the paper itself. The models used are all well-known and their licensing information is publicly available.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `https://paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [YES]

    Justification: Yes, the new asset (`IFF` benchmark) is documented throughout the paper. We detail its design principles and content in the Methodology section. The appendix provides further details on the framework architecture, and we include a README file with usage instructions in the public code repository.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not applicable. Our paper does not involve crowdsourcing or research with human subjects. The benchmark prompts were created by the authors of the paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Not applicable. As our research did not involve human subjects, Institutional Review Board (IRB) approval was not required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.