Who Evaluates the Evaluator: Reaching Autonomous Consensus on Agentic Outputs

Authors: *Jake Skinner, Davis Li, Adithya Ramanathan*
Team: *Hebbia Research & Product*

Abstract (TLDR):

Large language model (LLM) systems now sit at the heart of many enterprise workflows, yet the industry still lacks a principled, reproducible way to decide whether one prompt, model, or agentic configuration is genuinely better than another. We introduce Hebbia's consensus-based evaluation framework—a statistically grounded, model-agnostic approach that scales from single-prompt tweaks to multi-agent orchestration. Building on prior work such as G-Eval, GPTScore, and BooookScore, we fuse long-context awareness with permutation-based hypothesis testing to minimize false positives while surfacing actionable signal for ML engineers. Alongside a full methodological walkthrough, we present a case study benchmarking popular OpenAI, Google, Grok, and Anthropic models across abstractive (reasoning and summarization) and extractive (exact string, term, and verbatim) tasks within the financial services industry. We found that when we validated our automated scores against human-labeled examples from former hedge fund and private equity investors the results matched their expert opinion. More generally, our results highlight clear capability trade-offs that inform production model selection and agent design.

Introduction - Why "Good Enough" Isn't Enough

The distinction between evaluation and testing is often lost—just ask any teacher or school administrator. Testing tends to focus on measuring specific skills or knowledge, usually with a right-or-wrong answer key in hand. Evaluation, on the other hand, is the awkward cousin that asks a more philosophical question—is the answer *actually* good? Is it relevant, insightful, or readable. That distinction matters the moment you stop demoing a single prompt in a Jupyter notebook and start orchestrating a chain of half a dozen agents against 3000-token earnings transcripts. One stray temperature tweak can lift ROUGE-L while simultaneously hallucinating a brand-new division of your company or a legal precedent that doesn't actually exist. Welcome to the deep end.

At Hebbia, we run hundreds of thousands of LLM calls each day—often in parallel, always under latency constraints. Our prompt-engineering process looks less like quiet academic research and more like a Formula 1 pit stop: wrench, refuel, deploy, pray. Except the car is built from stochastic fog, the track layout changes every lap, and the telemetry lights are occasionally hallucinated. As AI engineers, when we bump verbosity by 20%, conciseness can crater; when

we clamp hallucination rate, suddenly factual coverage could evaporate. Very quickly, the notion of “better” devolves into pure vibes—and nobody wants to explain to security why *vibes* crashed production.

Given how long the AI race has been underway (barreling full speed toward AGI, consciousness… or just better autocomplete), you’d think LLM evaluation would be a solved problem. Turns out, it’s not. The open-source evaluation landscape is still a patchwork of clever scripts, one-off metrics, and statistical shortcuts that would make a first-year biostatistician cry. We need a framework that treats qualitative judgments with the same mathematical respect we give to A/B tests: hard p-values, bootstrap confidence intervals, and clear “ship or skip” decisions that even product managers can trust. In short, *vibes need standard errors*. This paper lays out how we got there, what we learned from benchmarking the usual LLM suspects, and how you can replicate—and extend—the approach without reinventing the wheel.

Standing on the Shoulders of Giants

Before we built our own evaluation framework, we reviewed several foundational efforts. If you're serious about LLM evaluation, these are required reading:

G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment

- Link: Paper Here
- Authors: Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, Chenguang Zhu
- One-liner: Pioneered the use of logits to create a continuous distribution of evaluation scoring.

GPTScore: Evaluate as You Desire

- Link: Paper Here
- Authors: Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, Pengfei Liu
- One-liner: Created a robust framework for generating and applying evaluation criteria across tasks and models.

BooookScore: A systematic Exploration of Book-length Summarization in the Era of LLMs

- Link: Paper Here
- Authors: Yapei Chang, Kyle Lo, Tanya Goyal, Mohit Iyyer
- One-liner: Took on the Herculean task of long-form summarization and invented a clever head-to-head scoring routine. Also gets points for best paper title in this list.

Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena

- Link: Paper Here
- Authors: Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al.
- One-liner: Asked the original question—can an LLM judge another LLM? Spoiler alert: yes, and it turns out GPT-4 agrees with humans more than humans agree with humans. One of the most comprehensive and scaled studies of LLM-based judgment to date.

Hebbia's Qualitative LLM Evaluation Framework

Our approach to LLM evaluation borrows heavily from G-Eval and GPTScore, with some long-context seasoning from *BooookScore*. We've tried to pull together the best elements of these frameworks and build something that feels more in line with how traditional statistical hypothesis testing works—because who doesn't love bootstrapped p-values?

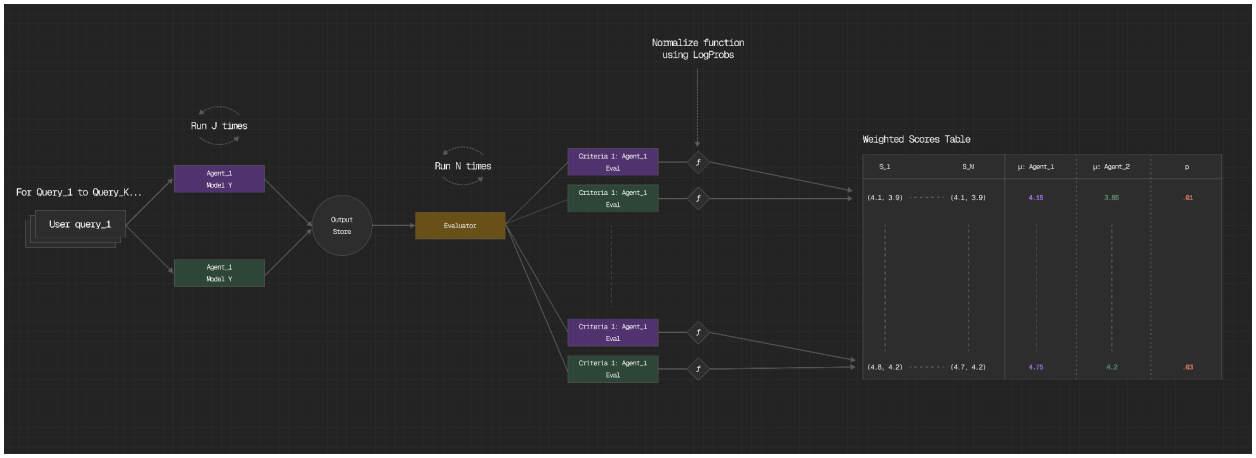
Enough talk—let's get in the ring.

Suppose we want to optimize our prompts based on a set of arbitrary-but-reasonable qualitative criteria. In the red corner, we have Agent_{HO} (our control, the null hypothesis). In the blue corner, Agent_{HA} (our new and possibly improved agent).

Here's the simplified version of the process:

1) Give a common set of source documents to Agent_{HA} and Agent_{HO}
2) Submit a series of questions or tasks to each agent and store their responses. For more robust results, submit each question J times with varying temperature settings.
3) Construct an evaluation agent (Agent_{EVAL}) to independently score the responses. Each criterion-question pairing is a separate LLM call, and each response is rated on a Likert scale from 1 to 5. This separation is key: it ensures that each score is independent and discrete, untouched by the bias of scoring multiple things at once or comparative bias by doing head-to-head evaluations.
4) For each Agent_{EVAL} response, capture and store the log probabilities for its score token and exponentiate these into linear probabilities for easier interpretability.
5) Use the array of linear probabilities to normalize their respective tokens and then sum them to produce a weighted-average score for a specific criterion-question pairing.
6) Once this process is complete for all criteria, queries, and for Agent_{HA} and Agent_{HO}, store the results; if there are other surface areas (aka other prompt families), then run them.
7) Repeat this entire process N times.
    a) LLMs are inherently noisy and stochastic, and running this just once is akin to flipping a coin and calling it science. By repeating, you get a sample distribution over which you can actually do hypothesis testing and avoid Type I errors—aka false positives, aka "we thought it was better, but it was just lucky."
8) After N runs are complete, align and pool the data across disparate runs and store as an experimental set.

A visual representation of this algorithm is as follows:



Fun with Numbers: Evaluation Criteria and Statistical Testing

For abstractive evaluation criteria, we devised a shortlist assembled from across our literature review. Each criterion includes three positive and three negative examples to anchor what “good” and “bad” looks like. This isn’t just window dressing—we found that giving LLM based scorers concrete examples dramatically improves consistency and sharpens distinctions between vague concepts like “clarity” and “factual grounding.” We also annotate specific criteria with custom notes for the evaluator model that call out common pitfalls we’ve run into while iterating on this framework—things like scoring verbosity too generously, or judging markdown-based responses too harshly.

Once the results are in, the real question becomes: when is a difference in scores *actually* meaningful?

To test this, we explored three paths for significance testing between sample means:

1. **Isolated by question (cross-criteria)** – Does one prompt outperform another across *all criteria* on a specific question?

2. **Isolated by criterion (cross-question)** – Does one prompt perform better on a specific *criterion* across *many questions*?

3. **Isolated by both question and criterion** – Does one prompt do better on *this criterion* on *this specific question*?

For each path, we run a two-sided permutation test with $\alpha = 0.05$ over 10,000 iterations. This lets us capture not only improvements but also regressions (things can always get worse). If we

had a much larger dataset, we'd also consider non-parametric alternatives like the Mann-Whitney U test—just make sure it's two-sided to catch both directions of change.

Bringing It All Together: Applications

Given that Hebbia is model and platform agnostic, we offer a wide variety of models from Anthropic (Claude) and OpenAI (the GPT family). We also cater to Gemini models, but they were not tested in this experiment. Therefore, one of the questions we most often get is which models to use or not use for a given task.

In our setup, we take a one-vs-all and one-vs-one approach. For testing a specific agent relative to all peers, we state that Agent_{HO} is the collection of all criteria scores from agents powered by our non-target model with Agent_{HA} representing the evaluator scores for our target model. We then design a battery of extractive and abstractive questions that require synthesis, critical thinking, and different forms of needle-in-the-haystack-like extraction.

With this experimental design in mind, the question we want to answer is, "Does providing an agent with model X, in comparison to model Y, improve or degrade the quality of answers as defined by our criteria?" Stated more scientifically across the aforementioned criteria, we can represent the hypothesis testing procedure as:

Let

H_0^i be the *null hypothesis* for the i-th test.

H_A^i be the *alternative hypothesis* for the i-th test.

α be the *significance level* (e.g., 0.05).

p_i be the *p-value* obtained from statistical test i.

m be the t*otal number of hypotheses tested*.

Then the full test procedure becomes a set of simultaneous hypothesis tests, one for each level as defined by our aforementioned testing paths, with Type I errors controlled by our permutation-based p-values. Who knew vibes could be so scientific?

With the entire procedure being represented as:

$$[\forall i \in \{1, \ldots, m\}, \quad \text{if } p_i < \alpha, \quad \text{reject } H_0^i \text{ in favor of } H_A^i.]$$

And the null hypothesis as:

H_O^i: There is no significant difference in the i-th test statistic

Since Hebbia works with many large financial institutions, we decided to focus our evaluation across a number of tasks that our clients tackle most often. We split the tasks across abstractive and extraction tasks—for extractive tasks we evaluated not only whether the agent was able to get the correct answer but also whether it properly followed formatting instructions for the returned answer. Abstractive tasks required a bit more nuance; for these questions, we evaluated the response across the following criteria:

- Plausibility
- Relevance
- Specificity and Detail
- Insightfulness/Analytical Sharpness
- Logical Coherence
- Conciseness

A sample of the types of questions we asked are as follows:

Extractive Samples

- What was the company's total revenue for the quarter?
- What was the company’s digital sales as a percentage of total revenue?
- What is the total number of stores open in the Middle East?
- What is the company’s guidance for revenue in the next quarter?
- What are the LTOs this quarter?
- What are the new store targets (number and date)?

Abstractive Samples

- Summarize the primary factors that impacted sales performance this quarter.
- Summarize management's commentary on industry trends.
- Summarize management’s commentary on macroeconomic consumer spending trends.
- Why is management confident in their growth targets?
- What were analysts concerned about?
- What did management struggle to answer?
- Return verbatim all of the analyst questions asked by John Smith (name redacted).

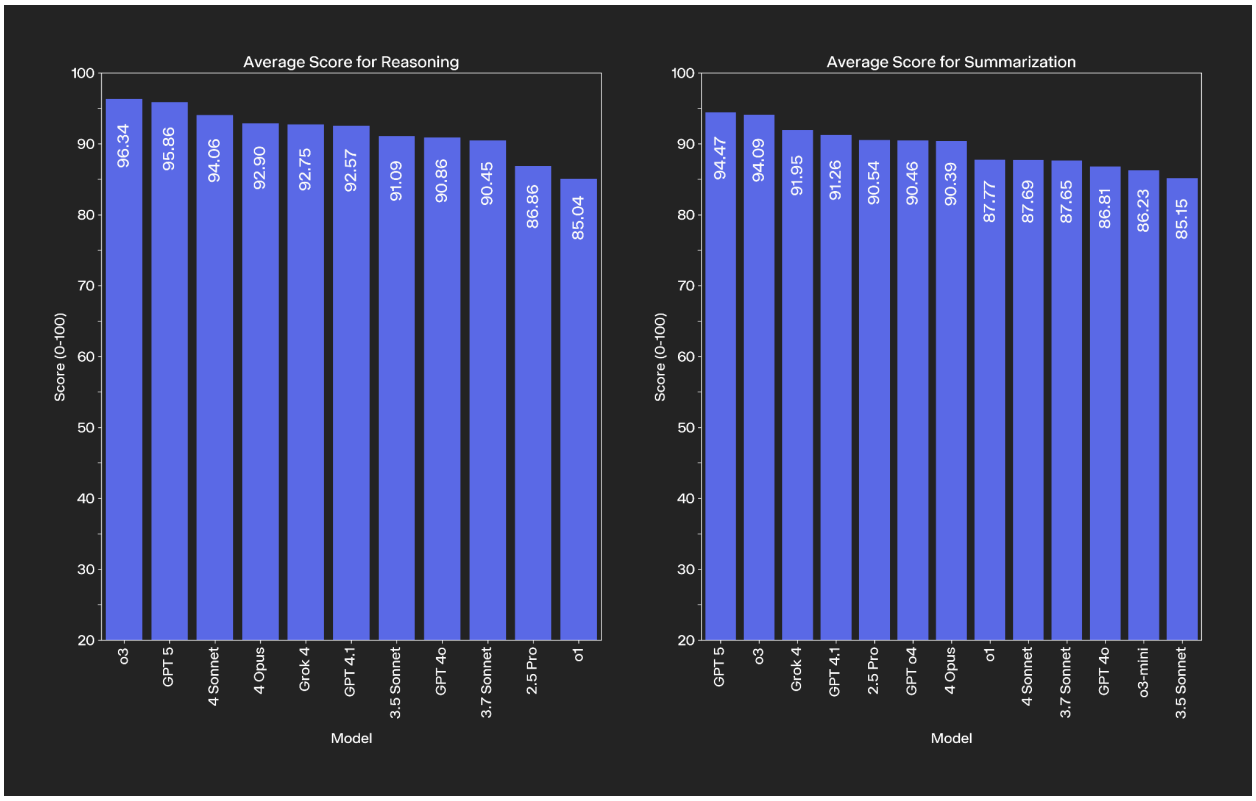
Our hyperparameters were as follows:

- $J = 3$
- $N = 50$
- Model Type: varied
- Model Temperature: 0.1-1.0 (model dependent variations)

Data! Data! Data!

Abstractive Results

Abstractive Criteria Scores by Question Category

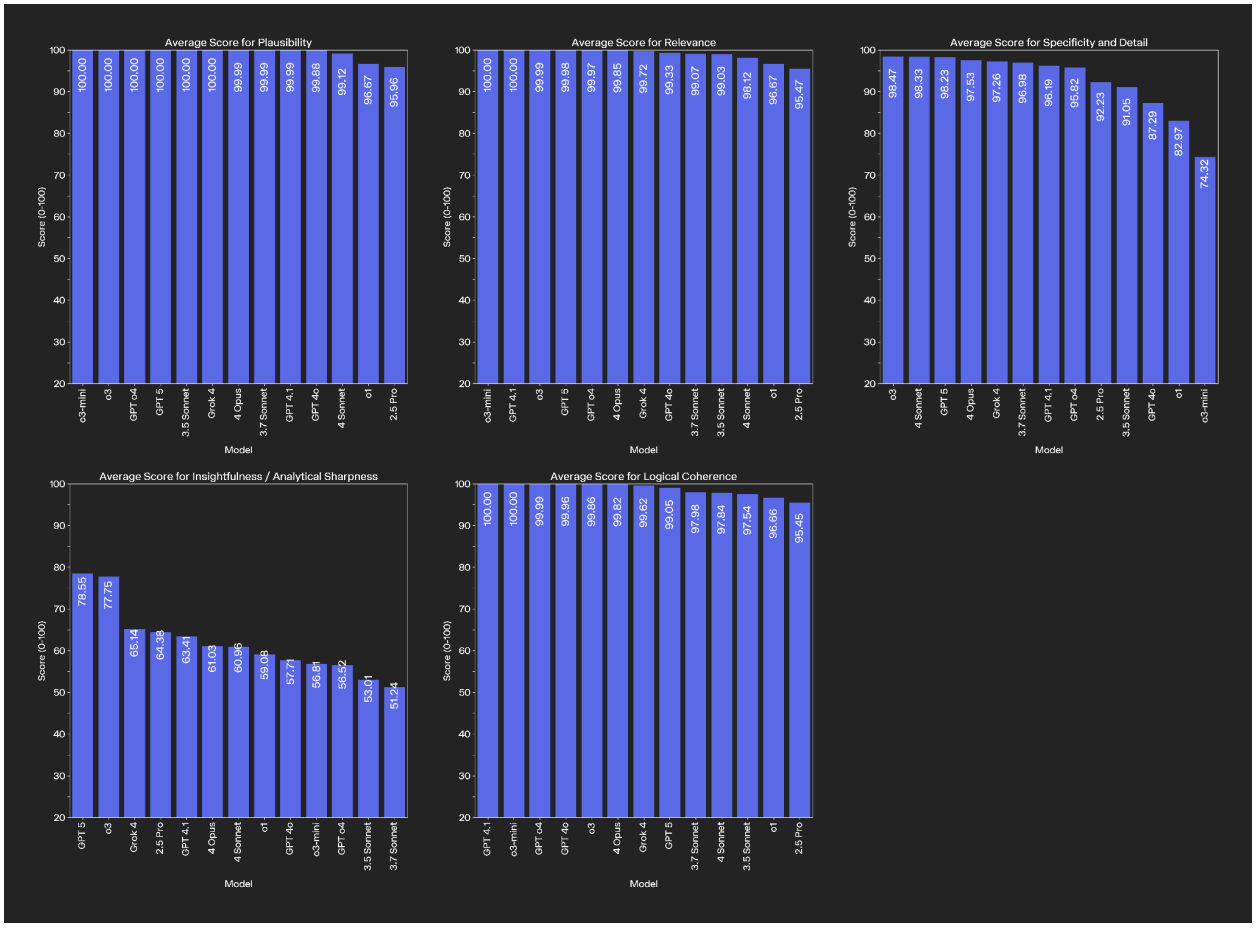


The aggregated raw score results reveal some fascinating patterns! Across our two broad meta-categories of abstractive tasks—Summarization and Reasoning—we see:

- That o3 and GPT-5 consistently emerge as the top performers with meaningful distance between them and their closest peers
- We observe that the Anthropic models (4-sonnet and 4-opus) perform better with reasoning than summarization–suggesting architectural or training differences that favor analytical tasks over distillation tasks

Note that for Reasoning, we removed the mini-models since they consistently performed below the rest of the pack—a clear indication that model size matters for complex reasoning tasks.

Abstractive Criteria Scores by Question Subcategory

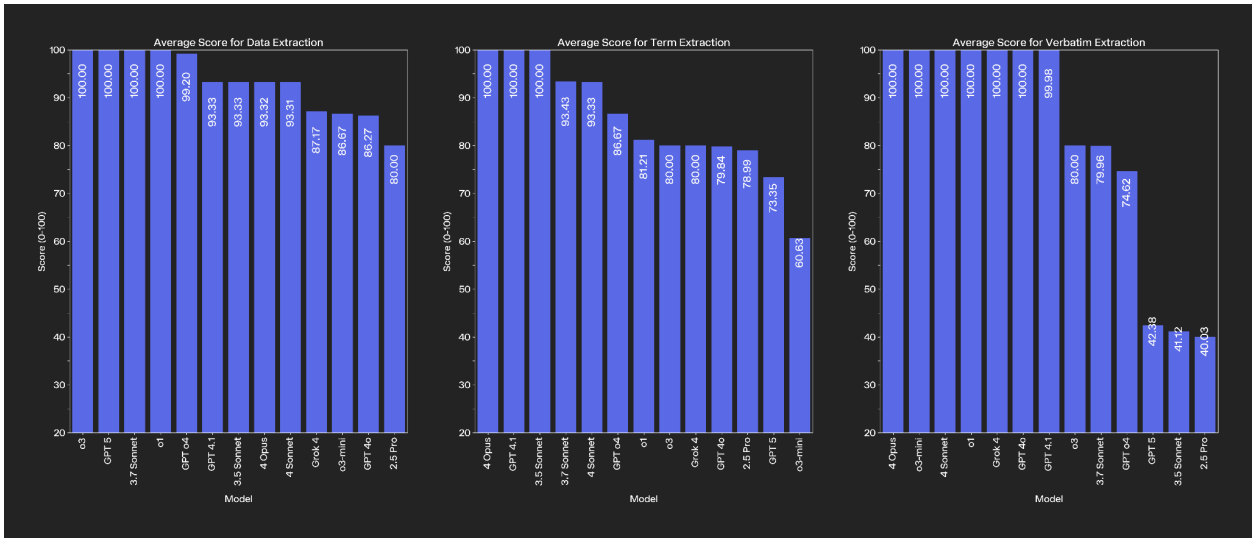


Looking deeper into the individual criteria scores:

- **GPT-5's dominance** is particularly pronounced in Insightfulness/Analytical Sharpness, suggesting its training has optimized for deep analytical capabilities. More on this in the extractive section below.
- **GPT-4.1** shows remarkably consistent performance across criteria, never falling below 4.0, making it a reliable "generalist" choice.

Extractive Results

Extractive Criteria Scores by Question Category



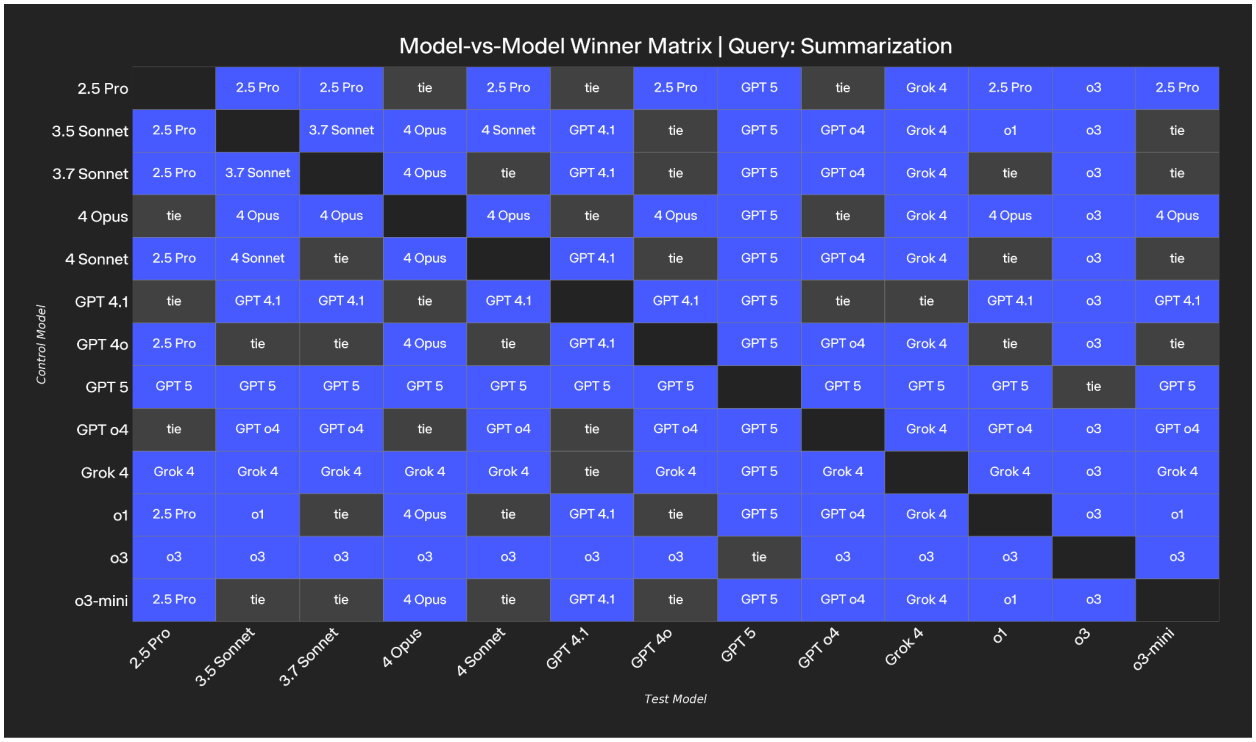
The extractive results paint a different picture entirely. While o3 and GPT-5 dominated in Reasoning and Summarization categories, we see that they struggle at both term extraction and verbatim extraction compared to their peers. This is a critical finding—GPT-5 and o3's strength in understanding and synthesis comes at the cost of precision in literal reproduction.

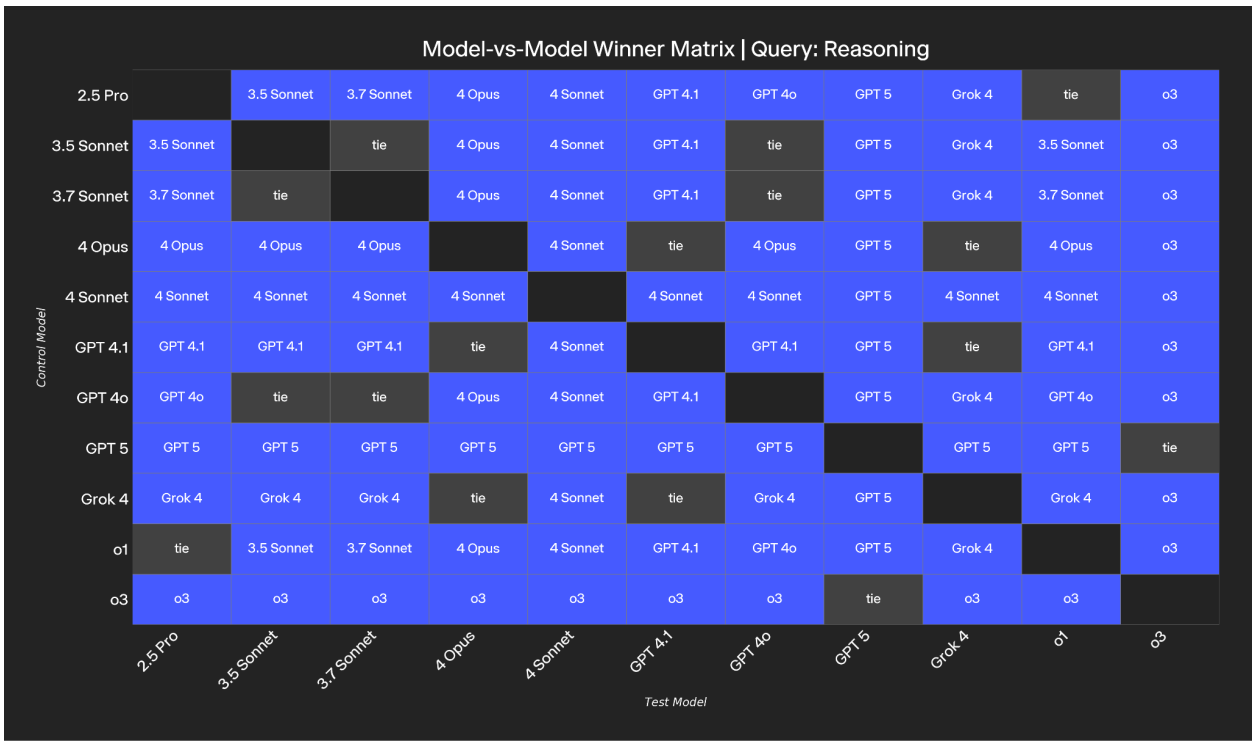
Key observations:

- **OpenAI's 4.1 model** emerges as the extraction champion, particularly excelling at verbatim extraction with scores approaching 5.0.
- **The mini-models** show surprising competence with data extraction, suggesting that structured extraction tasks may not require the computational overhead of larger models.
- **Claude models** maintain middle-of-the-pack performance, neither excelling nor failing dramatically.
- **GPT-5 is Chatty**—it is underwhelming when asked to extract specific information and tends to break formatting instructions due to its verbosity.

Head-to-Head Results and Statistical Significance

Next, we get down to the brass tacks of head-to-head battles and the assessment of statistical significance. As a reminder, we're using a two-sided permutation test with 10,000 iterations and an alpha at 0.05. To produce the following results, we run each model's results against every single one of its peers independently—we focus specifically on cross-category and cross-criterion performance.





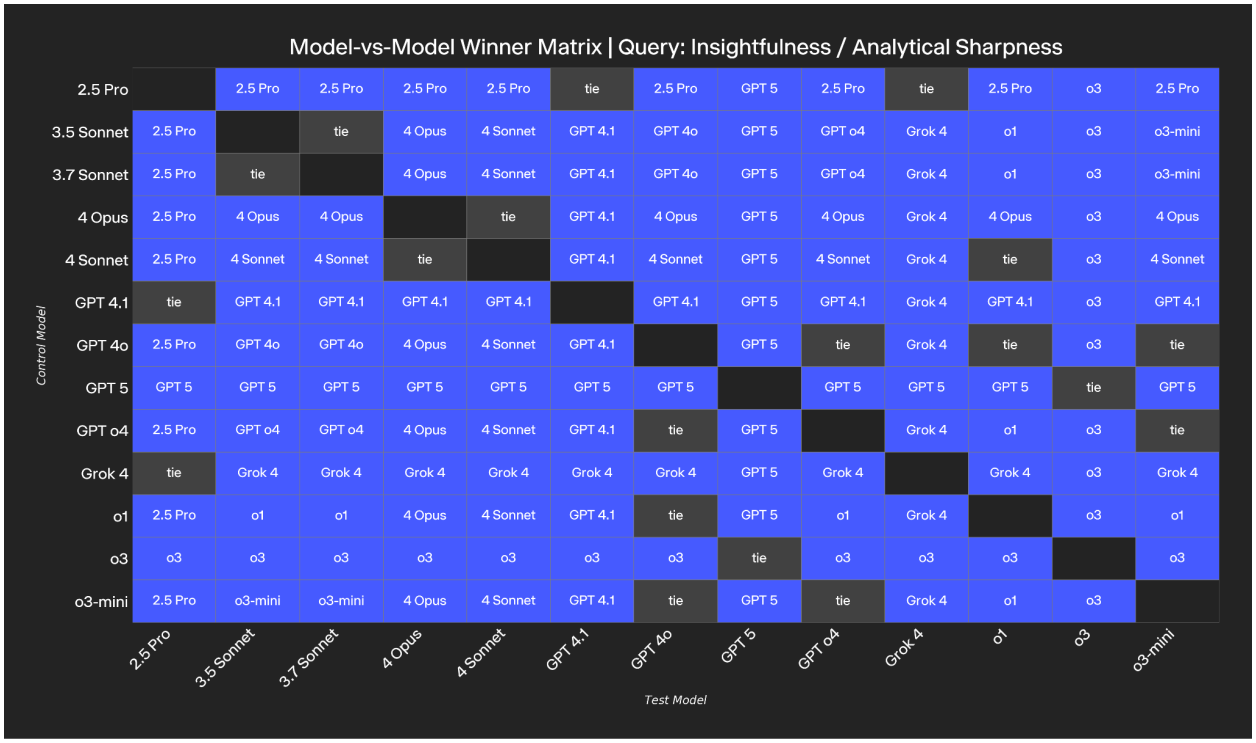
For summarization:

- GPT-5 achieves statistical significance against all models except itself (obviously) and o3.

- 4-opus consistently loses to newer GPT models, suggesting their rapid advancement in summarization capabilities.
- Ties between closely matched models (like 3.5 and 3.7) indicate marginal improvements between versions.

For reasoning:

- The dominance hierarchy is clearer: GPT 5 >= o3 > 4.1 > 4-sonnet > others.
- A higher percentage of ties appear in this category, suggesting reasoning capabilities have plateaued somewhat across model families.
- Note that we removed o4 and 03-mini from this specific category

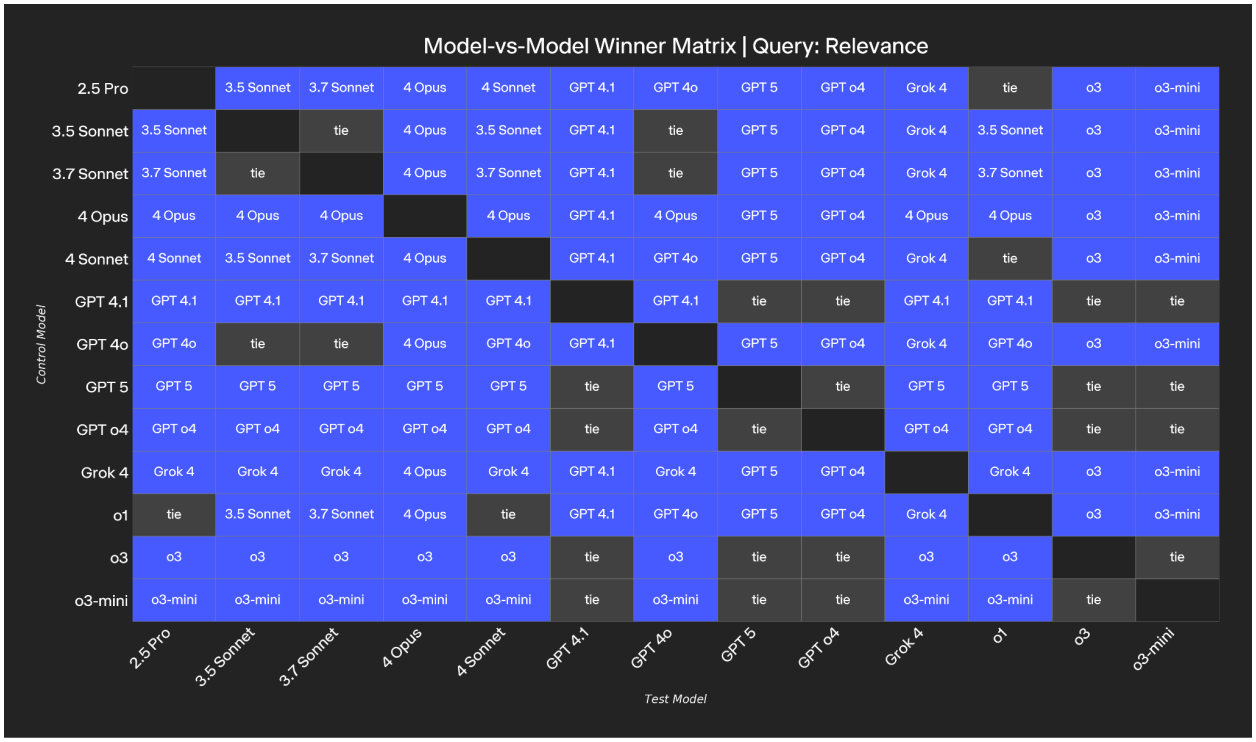


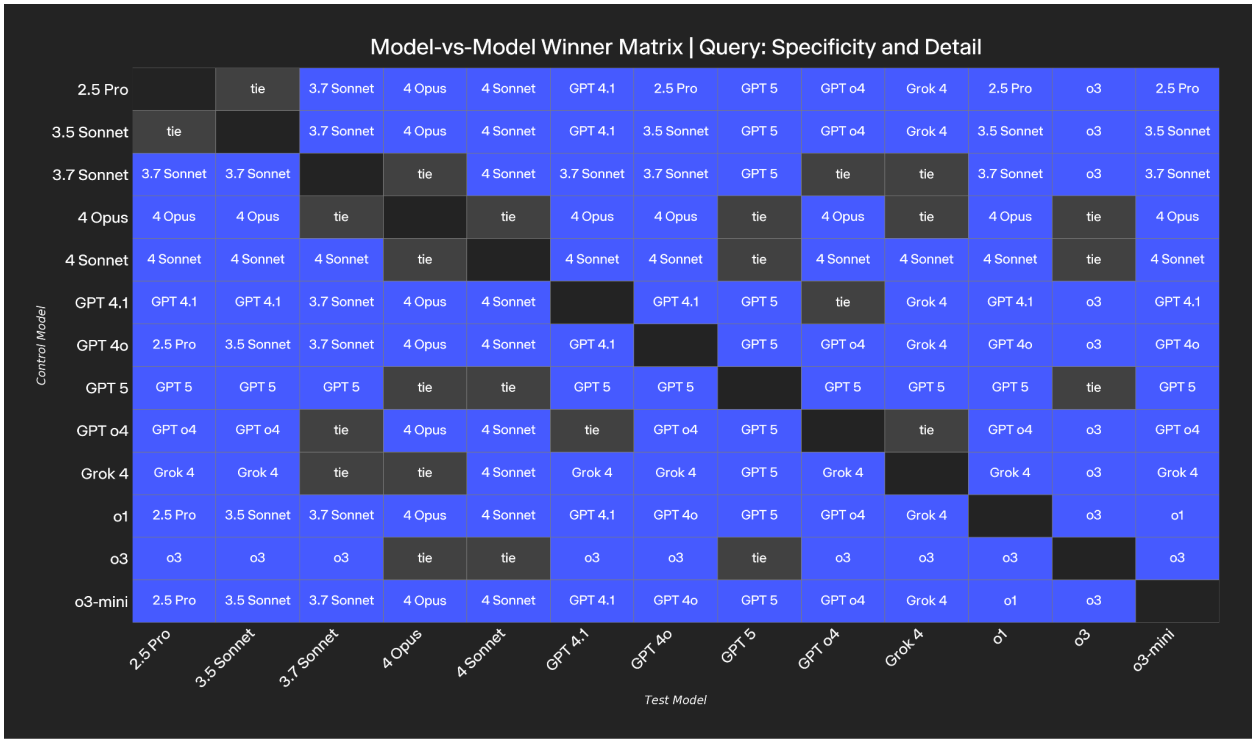
Model-vs-Model Winner Matrix | Query: Logical Coherence

Control Model \ Test Model	2.5 Pro	3.5 Sonnet	3.7 Sonnet	4 Opus	4 Sonnet	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	o1	o3	o3-mini
2.5 Pro		3.5 Sonnet	3.7 Sonnet	4 Opus	4 Sonnet	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	tie	o3	o3-mini
3.5 Sonnet	3.5 Sonnet		tie	4 Opus	tie	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	tie	o3	o3-mini
3.7 Sonnet	3.7 Sonnet	tie		4 Opus	tie	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	tie	o3	o3-mini
4 Opus	4 Opus	4 Opus	4 Opus		4 Opus	GPT 4.1	GPT 4o	4 Opus	GPT o4	4 Opus	4 Opus	tie	o3-mini
4 Sonnet	4 Sonnet	tie	tie	4 Opus		GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	tie	o3	o3-mini
GPT 4.1	GPT 4.1	GPT 4.1	GPT 4.1	GPT 4.1	GPT 4.1		tie	GPT 4.1	tie	GPT 4.1	GPT 4.1	GPT 4.1	tie
GPT 4o	GPT 4o	GPT 4o	GPT 4o	GPT 4o	GPT 4o	tie		GPT 4o	tie	GPT 4o	GPT 4o	GPT 4o	tie
GPT 5	GPT 5	GPT 5	GPT 5	4 Opus	GPT 5	GPT 4.1	GPT 4o		GPT o4	Grok 4	GPT 5	o3	o3-mini
GPT o4	GPT o4	GPT o4	GPT o4	GPT o4	GPT o4	tie	tie	GPT o4		GPT o4	GPT o4	GPT o4	tie
Grok 4	Grok 4	Grok 4	Grok 4	4 Opus	Grok 4	GPT 4.1	GPT 4o	Grok 4	GPT o4		Grok 4	o3	o3-mini
o1	tie	tie	tie	4 Opus	tie	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4		o3	o3-mini
o3	o3	o3	o3	tie	o3	GPT 4.1	GPT 4o	o3	GPT o4	o3	o3		o3-mini
o3-mini	o3-mini	o3-mini	o3-mini	o3-mini	o3-mini	tie	tie	o3-mini	tie	o3-mini	o3-mini	o3-mini	

Model-vs-Model Winner Matrix | Query: Plausibility

Control Model \ Test Model	2.5 Pro	3.5 Sonnet	3.7 Sonnet	4 Opus	4 Sonnet	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	o1	o3	o3-mini
2.5 Pro		3.5 Sonnet	3.7 Sonnet	4 Opus	4 Sonnet	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	tie	o3	o3-mini
3.5 Sonnet	3.5 Sonnet		tie	tie	3.5 Sonnet	tie	3.5 Sonnet	tie	tie	tie	3.5 Sonnet	tie	tie
3.7 Sonnet	3.7 Sonnet	tie		tie	3.7 Sonnet	tie	3.7 Sonnet	tie	tie	tie	3.7 Sonnet	tie	tie
4 Opus	4 Opus	tie	tie		4 Opus	tie	4 Opus	tie	tie	tie	4 Opus	tie	tie
4 Sonnet	4 Sonnet	3.5 Sonnet	3.7 Sonnet	4 Opus		GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4	4 Sonnet	o3	o3-mini
GPT 4.1	GPT 4.1	tie	tie	tie	GPT 4.1		GPT 4.1	tie	tie	tie	GPT 4.1	tie	tie
GPT 4o	GPT 4o	3.5 Sonnet	3.7 Sonnet	4 Opus	GPT 4o	GPT 4.1		GPT 5	GPT o4	Grok 4	GPT 4o	o3	o3-mini
GPT 5	GPT 5	tie	tie	tie	GPT 5	tie	GPT 5		tie	tie	GPT 5	tie	tie
GPT o4	GPT o4	tie	tie	tie	GPT o4	tie	GPT o4	tie		tie	GPT o4	tie	tie
Grok 4	Grok 4	tie	tie	tie	Grok 4	tie	Grok 4	tie	tie		Grok 4	tie	tie
o1	tie	3.5 Sonnet	3.7 Sonnet	4 Opus	4 Sonnet	GPT 4.1	GPT 4o	GPT 5	GPT o4	Grok 4		o3	o3-mini
o3	o3	tie	tie	tie	o3	tie	o3	tie	tie	tie	o3		tie
o3-mini	o3-mini	tie	tie	tie	o3-mini	tie	o3-mini	tie	tie	tie	o3-mini	tie	





At the criterion level, the specific results look as follows:

For specific criteria:

- **Relevance:** Most models achieve statistical ties, suggesting this is a "solved" problem for modern LLMs, though it's interesting GPT-4.1 and 5 tie in this area.
- **Plausibility:** Near-universal ties indicate all tested models have achieved human-level plausibility.
- **Logical Coherence:** Clear stratification emerges, with GPT-5 and o3 dominating, followed by 4.1, then a cluster of Claude models.
- **Insightfulness/Analytical Sharpness:** The most discriminating criterion, with GPT-5 taking the lead over o3 and achieving significance against all competitors.

Multiple Comparisons and False Discovery Rate

An astute reader might notice we're conducting hundreds of hypothesis tests (models × criteria × questions). This raises the specter of multiple comparisons problems. While we don't apply Bonferroni correction (which we felt would be overly conservative), our use of permutation testing with 10,000 iterations provides robust Type I error control at the individual test level.

For future work, we're exploring False Discovery Rate (FDR) control methods like Benjamini-Hochberg, which would allow us to make stronger claims about the proportion of true discoveries among our significant results.

Key Insights and Takeaways

1. **Model Selection Is Task-Dependent:** o3 excels at synthesis and reasoning but struggles with precise extraction. Choose your model based on your primary use case.
2. **The Anthropic Paradox:** Claude models show a peculiar pattern—excellent reasoning capabilities coupled with verbose outputs. This suggests potential for prompt engineering to constrain response length while maintaining quality.
3. **Mini-Models Surprise:** For well-defined verbatim-extractive tasks, mini-models can compete with their larger siblings at a fraction of the computational cost.
4. **Statistical Significance ≠ Practical Significance:** Many statistically significant differences translate to score differentials of 5-10 points. While this may seem small, marginal improvements have significant implications for high-scale use cases. While it wasn't specifically tested in this trial run, the performance in analytical insightfulness could be an interesting proxy for tool-calling use cases.

The results didn't just feel right, they also matched expert opinion. We validated our automated scores against human-labeled examples from former hedge fund analysts and saw strong alignment. What a top-performing analyst would rate as insightful, our framework rated as insightful, too.

Technical Considerations and Limitations

Evaluation Metrics and Biases, Potential Gotchas

While our framework addresses many limitations of existing approaches, several challenges remain:

1. **LLM Evaluator Bias:** Using LLMs to evaluate LLMs introduces potential bias. Models may favor outputs similar to their own style.
2. **Criteria Interdependence:** Although we score each criterion independently, they're not truly orthogonal. High logical coherence often correlates with high plausibility, for instance.
3. **Context Window Effects:** Longer documents may disadvantage models with smaller context windows, even if their actual understanding is comparable.
4. **Prompt Sensitivity:** Small variations in evaluation prompts can lead to different scores. We've standardized our prompts through extensive iteration, but this remains a source of potential variance.

Statistical Power and Sample Size

With J=3 and N=50, we achieve reasonable statistical power for detecting large effects (Cohen's $d > 0.8$). However, smaller effects may go undetected. Power analysis suggests that for detecting medium effects ($d \approx 0.5$) with 80% power, we'd need approximately N=100 bootstrap iterations.

The trade-off is computational cost—each additional bootstrap iteration requires 3 × (number of questions) × (number of criteria) × (number of models) LLM calls. At current API pricing, this becomes expensive quickly, and while we take these steps in production we felt that for this evaluation the that approach would have been excessive.

Conclusion

At Hebbia, we believe that LLMs and the agents they power are central to the future of enterprise workflows, and thus evaluation must evolve alongside them—not just in scale, but in sophistication. In this paper, we introduced Hebbia's framework for consensus-based LLM evaluation with the aim of blending theoretical rigor with statistical and empirical validation.

By synthesizing advances from G-Eval, GPTScore, and BooookScore, and reinforcing them with permutation-based hypothesis testing, we constructed a system that can autonomously evaluate qualitative performance with statistical fidelity. Add in the ability for our users and our internal teams at Hebbia to define criteria ad hoc based on the objectives of a given agentic system, and you have a flexible and adaptable framework that scales with your needs.

Looking ahead, we aim to extend beyond simple model-to-model comparison to encompass system-level evaluations, planning agent benchmarking, and fine-grained debugging of agent behavior. As agentic systems become more complex—incorporating tool use, multistep reasoning, and dynamic context management—so too must the tools we use to measure them.

Precision matters—especially when "good enough" isn't enough. In the high-stakes world of financial analysis, legal discovery, and enterprise intelligence, the difference between 95% and 99% accuracy can be millions of dollars. Our evaluation framework helps ensure we're always moving in the right direction, one statistically significant improvement at a time.

The authors would like to thank the Hebbia engineering team for their patience during the hundreds of thousands of API calls required for this research, and the finance team for not looking too closely at the OpenAI, Google, and Anthropic invoices.