When LLMs Judge LLMs

Cross-Model Evaluation Bias in LLM-as-Judge Frameworks

Experiment 2: Expert Knowledge and Document Context Effects

Tamas Almasi, March 2026

Abstract

We are increasingly overloaded by information we want to process, and this is one of the reasons why AI and large language models have become part of everyday workflows. We use them to summarize texts, explain complex topics, and help us understand, review, or judge difficult material. At the same time, we also use LLMs to generate text, improve writing, and produce better outputs for a wide range of personal and professional scenarios.

Putting these two trends together, large language models have quietly become the first reader in more and more workflows: screening job applications, evaluating proposals, summarizing reports, or scoring student work. This creates a loop that most practitioners have not fully questioned: if one LLM writes the text and another LLM evaluates it, does it matter which models are in those roles?

I am running a series of experiments to answer this question. In my studies, I use five frontier models — ChatGPT, Gemini, Grok, DeepSeek, and Claude — in a two-phase setup. In the first phase, the models generate answers or summaries. In the second phase, all models evaluate all generated outputs in an LLM-as-Judge setting.

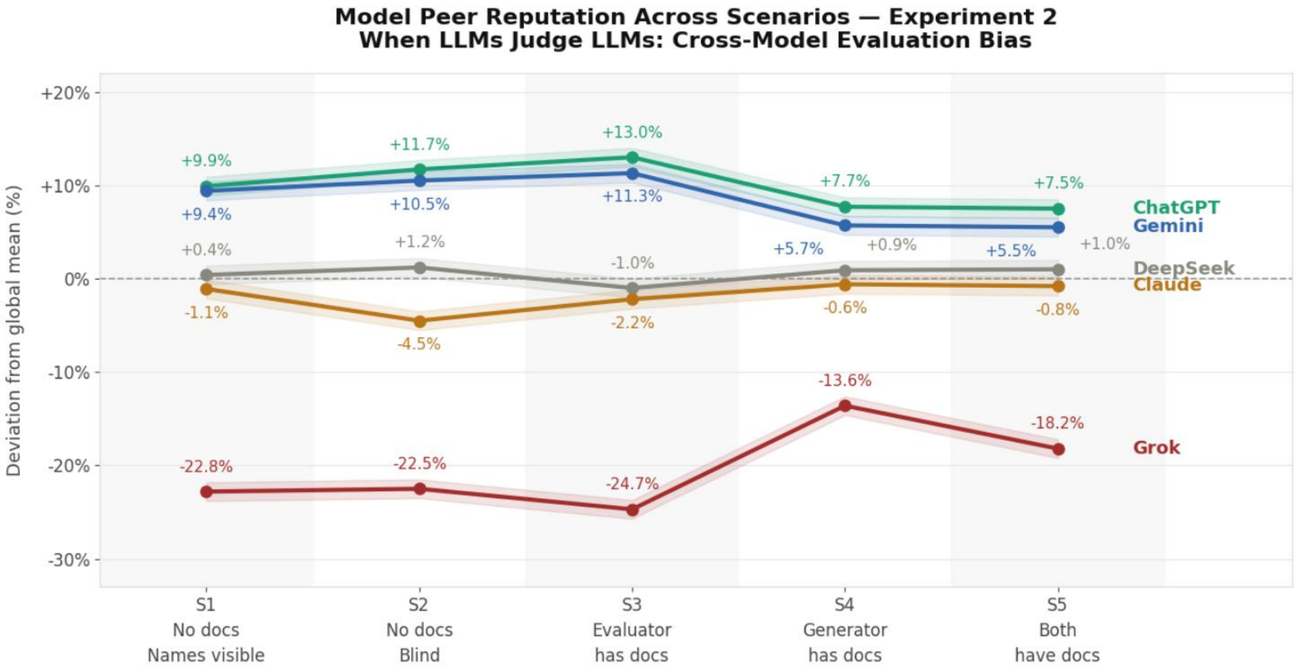
In the first experiment, I used World War II — a topic far enough in the past, and so extensively studied, that we expect broad historical consensus and limited/no divergence across models. Yet the results showed that the choice of generator model clearly matters. Systematic and reproducible cross-model evaluation bias emerged across repeated runs: some models were consistently judged more favourably than others. This means that when automated judgment replaces human review, both the generator model and the evaluator model can introduce measurable advantages or disadvantages.

In Scenarios 1 and 2, where neither generators nor evaluators had access to the source document, the generated texts were still rated relatively strongly overall, with average scores around 80%. Building on these two baseline conditions, the main finding is that generator quality has greater influence on evaluation outcomes than evaluator verification. When evaluators have access to the source material while generators do not, scores drop sharply because errors that previously appeared plausible can now be identified more easily (Scenario 3). When generators receive the source material while evaluators do not, scores rise strongly and evaluator disagreement collapses, showing that higher-quality, grounded outputs are recognized and rewarded even without direct verification (Scenario 4). When both generators and evaluators receive the documentation, average scores change only minimally, but evaluator-specific differences become more visible again (Scenario 5).

These results suggest an asymmetric mechanism: verification primarily acts as an error-detection process. It strongly penalizes inaccurate or incomplete outputs, but adds little once responses are already factually grounded.

Special attention is needed when models evaluate their own outputs in workflows such as self-checking, answer ranking, draft selection, or automated quality scoring, because some models consistently underrate themselves while others consistently overrate themselves, so self-scores can diverge substantially from peer judgment and distort the result the user sees.

These findings have direct practical implications for LLM-as-Judge workflow design: giving source documentation to the generator improves results more than giving it only to the evaluator, and model choice in both roles introduces systematic and reproducible bias. In practical terms, users should prioritize grounding the generating model with accurate source material and treat automated evaluation scores with caution, especially when self-evaluation is involved. If time and resources allow, they should work with multiple models in parallel; if not, they should prefer the models that are most consistently accepted by other models — in Experiment 2, the recommended order is: (1) ChatGPT or Gemini, (3) Claude or DeepSeek, and (5) Grok.



The chart shows a stable reputation hierarchy across all five scenarios: *ChatGPT* and *Gemini stay on top*, *DeepSeek and Claude remain in the middle* and *Grok is consistently the weakest outlier*. *Scenarios 1* and *2* show that even without documentation, *scores are already relatively strong and anonymization changes little. Scenario 3 brings the sharpest drop*, as evaluator access to the source document reveals weaknesses that previously sounded plausible. *Scenario 4 reverses this effect*: when generators receive the document, scores rise strongly and evaluator disagreement narrows. *Scenario 5 leaves average scores almost unchanged but increases the gap again by penalizing weaker outputs more strongly.*

Between Experiment 1 and Experiment 2: Context and Reflections

I am currently running a series of experiments to understand how LLMs evaluate each other's output. My central question is whether the choice of model for text generation carries measurable advantages or disadvantages — particularly given that many of us already use LLMs to create content and to summarise, evaluate, or advise based on that content.

The novelty of this study lies in systematically isolating the effects of evaluator knowledge, generator knowledge, and model identity on cross-model evaluation behaviour.

On the feedback I received

After publishing the first experiment, I received several comments: questions about my model selection, a preference for human judges over LLM judges, and concerns that the rapid evolution of models could quickly render my findings outdated.

Let me address each in turn.

On the question of ***human versus LLM judges***: we all expect, that another human will evaluate our work, whether it is a school assignment, a technical article, or a job application. Our expectation is fair and important. However, I am specifically interested in the growing class of cases where an LLM, not a human, is the first reader. My research does not aim to settle the broader debate about whether LLMs or humans are more appropriate judges in any given context. It simply asks: given that LLM-based evaluation already happens and is increasing, how do these models assess each other's output?

On model selection: my choices were guided by different leaderboards, namely Vellum leaderboard and OpenRouter usage statistics. I used the most widely used models reported at the time of the study. I made one exception with DeepSeek, as I included as the sole non-US model, representing Chinese and more broadly Asian AI development — chosen over alternatives such as Kimi. For each selected vendor, I used the default model version available through their GUI, whether web or mobile app, to simulate everyday usage patterns.

One observation worth noting: ***as I used the models programmatically via their respective vendor APIs***, there are meaningful differences compared to consumer-facing interfaces. Web and mobile apps are optimised for user experience — lower latency, more seamless file handling, e.g. smoother PDF integration. APIs, by contrast, can be slower and more complex to use. However, my working assumption is that the underlying quality of text generation and evaluation is equivalent across both access modes.

Regarding the concern that ***my findings may become outdated as models evolve***: model evolution is a fact, therefore my experiment represents the models in February-March 2026. But the model evolution offers another opportunity, what would we find if the same experiment is rerun in six months? Will the cross-model evaluation patterns change? Will models become more discriminating, might introduce new evaluative dimensions? Or will the relative patterns remain stable? I expect these questions will deepen our understanding of how LLM judgment functions over time.

Finally, it is worth acknowledging that ***techniques*** such as retrieval-augmented generation (RAG) can partially bridge differences in raw model output quality by grounding responses in external knowledge. However, in case of the everyday usage such techniques are not yet part of standard practice.

.

1. Project Overview

This paper describes the second experiment of comparing evaluation results, which are produced by different LLMs on texts generated by other LLMs. In practice this is an LLM-as-Judge framework, which applied systematically across multiple frontier models. Where Experiment 1 used a topic with broad scholarly consensus (World War II), this experiment shifts to a specialised, product-specific domain of expert knowledge: the DJI Mini 4 Pro drone user manual. The task is to describe the exact steps required to take off a DJI Mini 4 Pro drone.

The central question remains unchanged: what we are looking for is not how objectively correct the generated answers are, but how each model evaluates the others' output. This experiment introduces multiple evaluation rounds per scenario, progressively varying whether the user manual is shared with the generating models, the evaluating models, or both. The added dimension is whether providing the source document — to the generator, the evaluator, or both — alters the cross-model scoring patterns observed in Experiment 1.

Disclaimer: This is a specific experiment with a defined scope — particular models, versions, topic, and prompts. It was conducted to the best of the author's knowledge and abilities, and all methods, prompts, and configurations are fully transparent and available throughout this paper.

☝☝☝ *Practical Recommendation: When working on anything consequential, consider using multiple models in parallel — if the opportunity exists to do so.* ☝☝☝

2.1 Task Design

All five models — ChatGPT, Claude, DeepSeek, Gemini, and Grok — were asked the same question: *prepare a list with all exact steps required to take off a DJI Mini 4 Pro drone, with a strict 500-word limit*. This topic was chosen because it requires knowledge of a specific, proprietary product manual that the models are unlikely to have encountered in their training data — making it a genuine test of expert knowledge retrieval versus reliance on general reasoning. This remains an assumption; it cannot be verified directly, as the training data of frontier models is not publicly disclosed.

I've created five scenarios to assess the potential effect of sharing or holding back the extra knowledge across the generation and evaluation phases. In each scenario, the same five models both generated answers and evaluated each other's outputs. The evaluation prompts and scoring rubric are provided in **Appendix C**.

2.2 Experimental Design — Five Scenarios

The five scenarios form a structured matrix varying two dimensions: whether the source document (DJI Mini 4 Pro User Manual) was provided to the generating model, the evaluating model, both, or neither. An additional control dimension — whether model names were visible to evaluators — was tested in the first two scenarios.

Scenario 1 Baseline, No-One has Expert Knowledge	**Generator receives manual:** ✗ NO **Evaluator receives manual:** ✗ NO	**Generator model names visible to evaluator:** ✓ YES

Scenario 1: Neither generator nor evaluator receives the manual. Model names are visible to evaluators. This is the direct baseline: models rely entirely on prior training knowledge.

Scenario 2 Name Blind Control, No-One has Expert Knowledge	**Generator receives manual:** ✗ NO **Evaluator receives manual:** ✗ NO	**Model names visible to evaluator:** ✗ NO

Scenario 2: Identical to Scenario 1, but model names are replaced with anonymous labels (Answer A–E). This controls for potential evaluator bias introduced by knowing which model produced each answer.

Scenario 3 Evaluator Has Expert Knowledge	**Generator receives manual:** ✗ NO **Evaluator receives manual:** ✓ YES	**Model names visible to evaluator:** ✗ NO

Scenario 3: The evaluating models receive the user manual; the generating models do not. Model names are anonymised. Tests whether a document-informed evaluator applies different standards to the same generated outputs.

Scenario 4 Generator Has Expert Knowledge	**Generator receives manual:** ✓ YES **Evaluator receives manual:** ✗ NO	**Model names visible to evaluator:** ✗ NO

Scenario 4:The generating models receive the user manual; the evaluating models do not. Model names are anonymised. Tests whether document-grounded generation changes peer evaluation scores.

Scenario 5 Both Have Expert Knowledge	**Generator receives manual:** ✓ YES **Evaluator receives manual:** ✓ YES	**Model names visible to evaluator:** ✗ NO

Scenario 5: Both generators and evaluators receive the user manual. Model names are anonymised. This is the most information-rich scenario — evaluators can directly compare generated answers against the official source.

2.3 Model Selection

The same five models as Experiment 1 were used, queried directly through their respective vendor APIs. Model selection was based on the Vellum leaderboard and OpenRouter usage statistics at the time of the study, with DeepSeek included as the sole non-US model.

Model Label	Model String	API Provider
Gemini	gemini-3-flash-preview	Google Gemini API
OpenAI	gpt-5.2	OpenAI API
DeepSeek	deepseek-chat (v3.2)	DeepSeek API
Claude	claude-sonnet-4-20250514 and claude-sonnet-4-5-20250929	Anthropic API
Grok	grok-4-1-fast-reasoning	xAI API

Note: Claude's model version was updated mid-experiment to the latest available release. Scenarios 1 uses claude-sonnet-4-5-20250929; Scenarios 2–5 use claude-sonnet-4-20250514.

2. Baseline Evaluation (Scenario 1) without External Reference Material

Scenario 1 Baseline, No-One has Expert Knowledge	**Generator receives manual:** ✗ NO **Evaluator receives manual:** ✗ NO	**Generator model names visible to evaluator:** ✓ YES

3.1 Generation Phase: Observations on the Drone Take-off Task

Runtime and Output Size

Before analysing the content of the generated instructions, noticeable variation appeared in response latency and output length across the tested models. The measurements reflect API-based generation recorded during the experiment.

Model Label	Latency via API calls	Output Size	Observations
Gemini	~6.5 sec	~390 words	Fastest model while producing the most detailed instructions
OpenAI	~10 sec	~330 words	Moderate latency with consistent mid-length outputs
DeepSeek	~18 sec	~300 words	Slowest model despite relatively concise responses
Claude	~15 sec	~380 words	Balanced latency with detailed responses
Grok	~10 sec	~265 words	Fast response time but shortest outputs

Generic observation by a human of the provided content by the models: the generated instructions showed strong structural convergence, typically following a three-stage procedural pattern: physical preparation, system readiness checks, and execution of the take-off procedure. While the responses differed in style — ranging from manual-like step-by-step instructions to more contextual or tutorial-oriented explanations — all models produced technically coherent guidance, with differences mainly reflecting how procedural information was structured and emphasised.

A qualitative comparison of the generated instructions is presented in **Appendix C**.

Technical Note – External Source Reference in the *Grok* Response

Sometimes Grok's outputs ended with a reference to the official DJI website. The last sentence of the answer: *"Total words: 298. Fly safe! (Official DJI manual for visuals: dji.com/mini-4-pro/downloads)"*. It might raise the question of whether the model had performed a real-time internet lookup during the API call. Since the experiment did not enable any external search tools or web-access capabilities during the API call, this behaviour might be the result of the model generating a plausible reference based on patterns present in its training data rather than performing a live web query. Btw the link was correctly presented by Grok and it is up and running. Interestingly non of the other models citated this web link.

Technical Challenges

Implementing this pipeline required resolving the same model-specific technical constraints documented in *Section 4.2 of the first experiment's paper*:

- JSON Output Reliability
- Gemini Configuration Parameter Handling
- Temperature Parameter Consistency
- Token Limit Parameter Handling

3.2 The Evaluation Phase

During this evaluation phase the same solutions have been applied for the technical challenges, as with the first experiment and summarized there under the chapter: *4.2 Technical Challenges.*

3.2.1 Setup

Once the five generated texts were collected, the same five models were used as evaluators — simulating a recursive "LLM-as-Judge" scenario. In this first scenario, the evaluator models received the name of the generator models as well. The evaluation prompts (system and user) are provided in **Appendix D.**

The JSON output format was requested to make further analysis straightforward and to enforce structured, comparable output across all evaluator models.

3.2.2 Evaluation Phase – Results

3.2.2.1 Cross-Model Scoring Matrix

The heatmap below shows the full cross-model scoring matrix: how each model scored every other model, including itself, aggregated across all 26 evaluation runs.

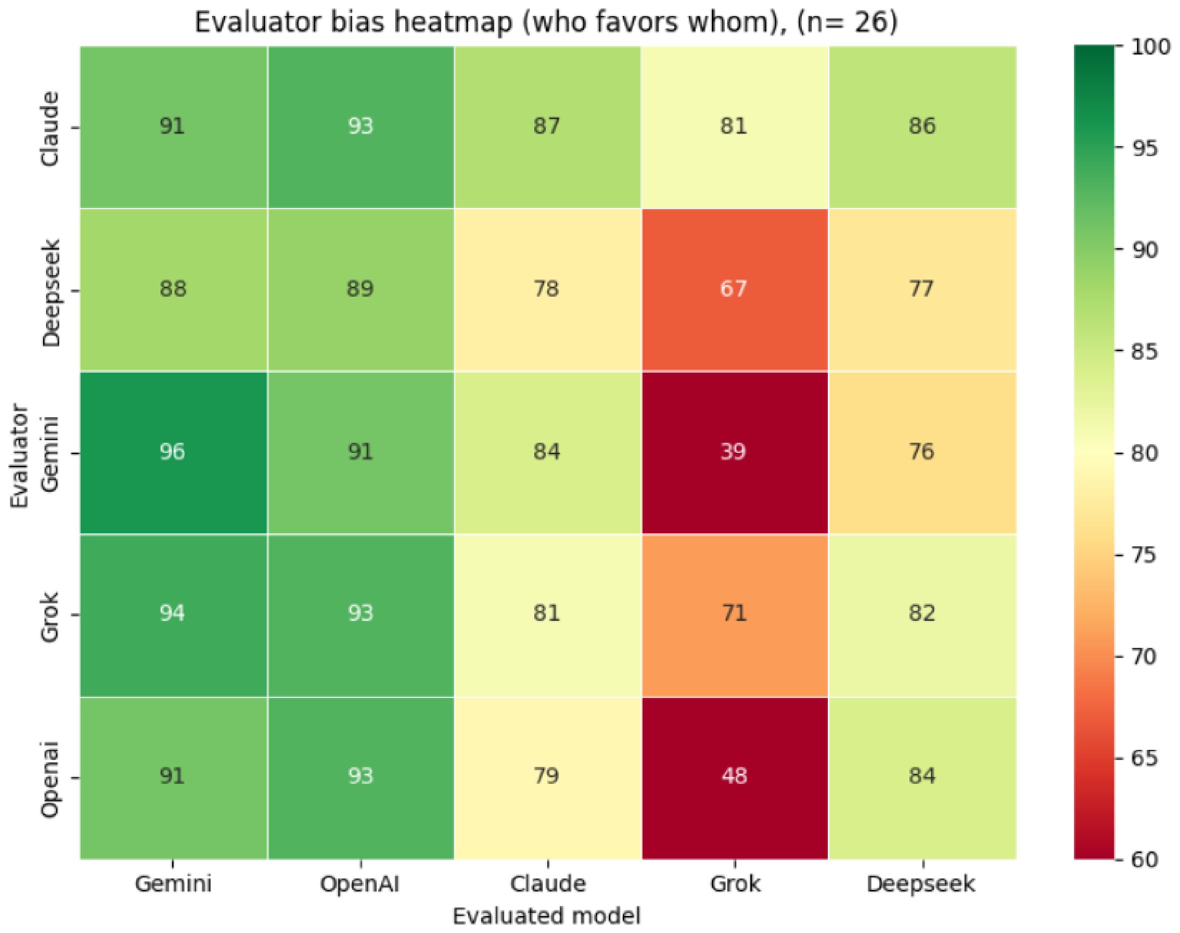


Figure 0.9: Cross-model scoring matrix, Scenario 1 — which model favours which model.

Across all 26 cycles, aggregate scores are high overall, with *ChatGPT* and *Gemini* emerging as the two consistently top-rated models by their peers.

Authors using *Grok* for text generation should be aware that peer models score their output measurably lower on average — a structural pattern, not a statistical artefact.

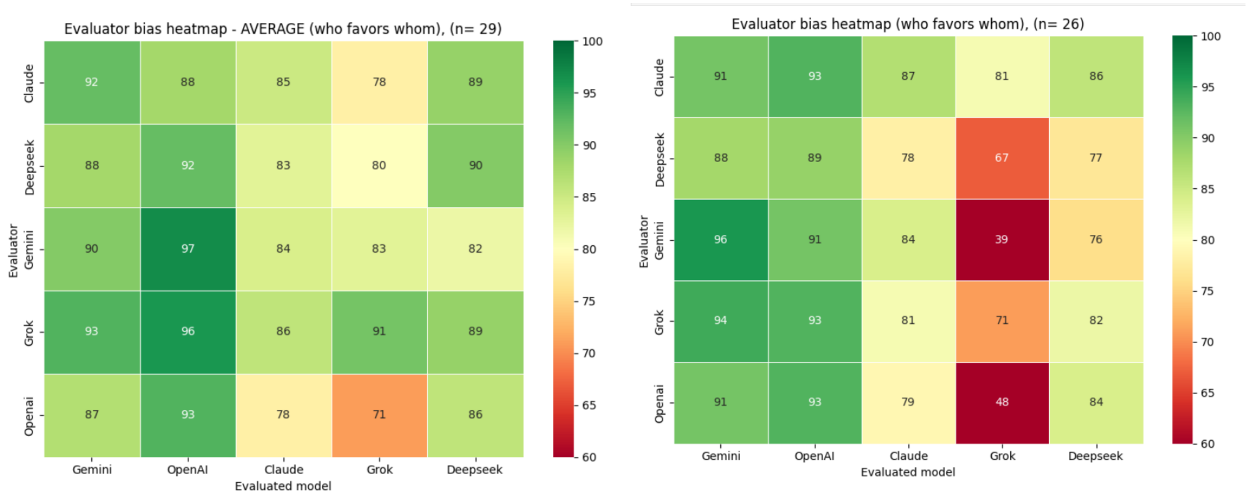


Figure 1: Side-by-side evaluator bias heatmaps across two experiments.

The left matrix (Experiment 1, n=29) reflects cross-model evaluation on a common knowledge domain — World War II — where all models are assumed to share a broadly similar academic and scholarly foundation. The right matrix (Experiment 2, n=26) shifts to a domain of specialised expert knowledge — DJI Mini 4 Pro drone operation — where no established academic or scholarly consensus exists, and models are unlikely to have been trained on the specific proprietary content required to answer accurately.

3.2.2.2 Cross-Experiment Observations

Comparing the two heatmaps reveals a consistent pattern alongside a domain-dependent divergence. *Gemini and OpenAI maintain the highest peer reputation across both experiments*, suggesting that evaluator preference for these models is stable regardless of knowledge domain.

The most significant structural difference between the two matrices is the score spread. In *Experiment 1 (World War II), scores cluster tightly in the upper range*, reflecting broad evaluator agreement in a domain where all *models presumably share a common academic knowledge base*. In *Experiment 2 (DJI Mini 4 Pro), the spread widens considerably*, with individual cells reaching as low as 39%. This indicates *that domain specialisation amplifies inter-model differentiation* — models that perform comparably on common knowledge tasks diverge markedly when the task requires specific proprietary expertise.

Grok shows the most pronounced domain sensitivity, as Grok receives the lowest scores in Experiment 2 by a substantial margin, particularly from *Gemini* and *OpenAI*. This suggests that Grok's responses were perceived as less technically credible in the expert knowledge domain, a pattern not visible in the common knowledge setting.

3.2.2.3 Evaluator Strictness

The next question is whether the models differ systematically in how strict or lenient they are as evaluators.

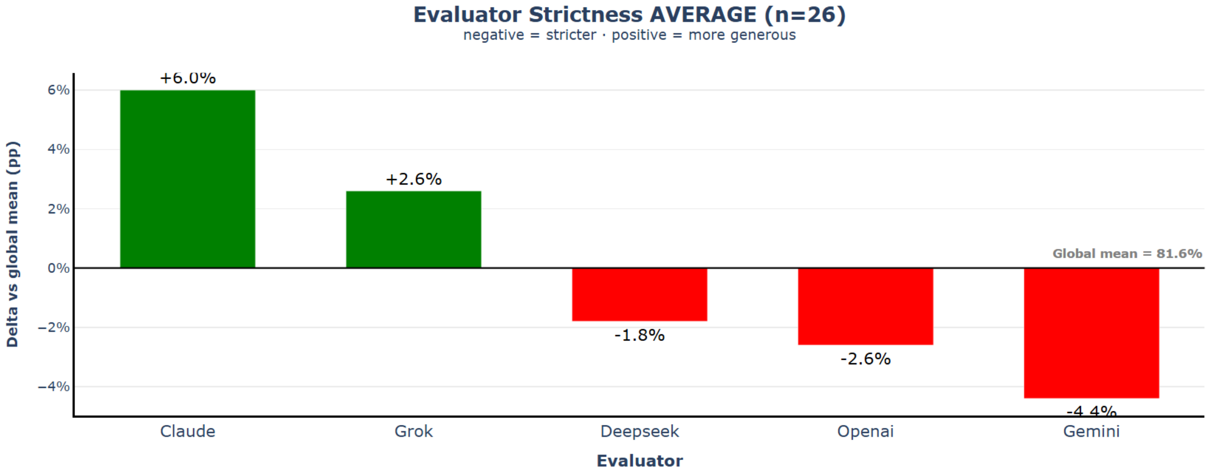


Figure 2: Evaluator generosity, Scenario 1 — how strict or lenient each model was when scoring the others.

The chart shows how strict or lenient each model was when evaluating the others relative to the global mean score of 81.6% across all 26 runs. *Claude* appears as the most lenient evaluator (+6.0% above the mean), followed by *Grok* (+2.6%), while *Gemini* is the strictest evaluator (−4.4%), with *OpenAI* (−2.6%) and *DeepSeek* (−1.8%) also scoring slightly below the global average. The total spread between the most lenient and strictest evaluator is roughly 10 %, indicating that evaluator behaviour alone can significantly influence the relative ranking of the models.

3.2.2.4 Model Reputation

The following figure addresses the question most relevant to practitioners: *which model should be used for text creation if the evaluator model is unknown?*

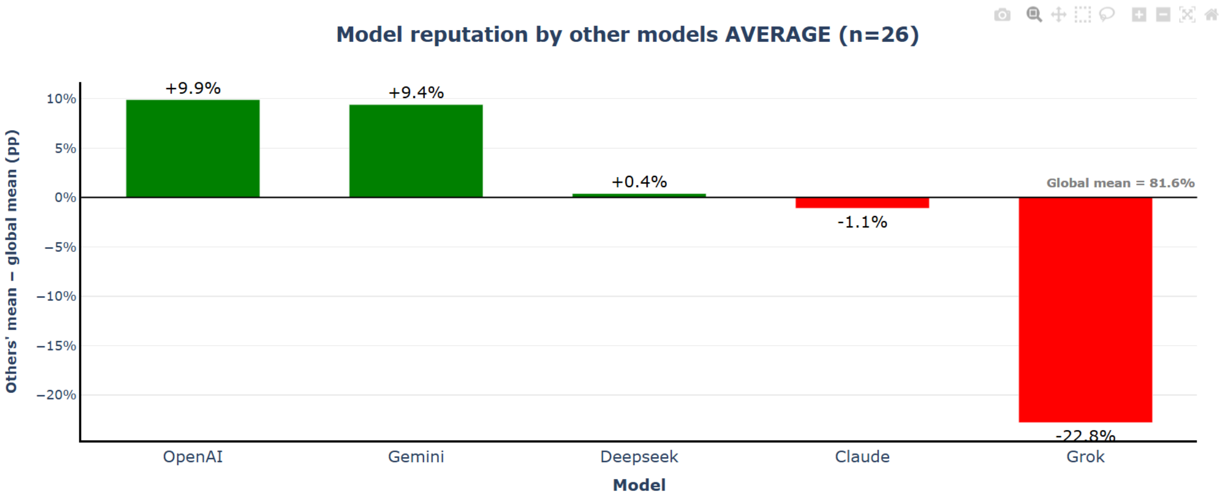


Figure 3: Model reputation, Scenario 1 — how each model is perceived on average by its peers (excluding self-assessment).

Figure 3 shows the average score each model received from the other models relative to the global mean of 81.6%. *OpenAI* and *Gemini* clearly stand out, scoring about +9.9% and +9.4% above the mean, respectively, while *DeepSeek* (+0.4%) and *Claude* (−1.1%) remain close to the overall average. In contrast, *Grok* appears as a strong negative outlier (−22.8%), creating a total spread of more than 30% between the highest- and lowest-rated models in the peer evaluation.

This is another warning signal for anyone using Grok for text generation should be aware that peer models score their output measurably lower on average — a structural pattern, not a statistical artefact.

3.2.2.5 Self-Assessment vs. Peer Assessment

A final dimension of this study is the gap between how each model rates its own output and how its peers rate the same output.

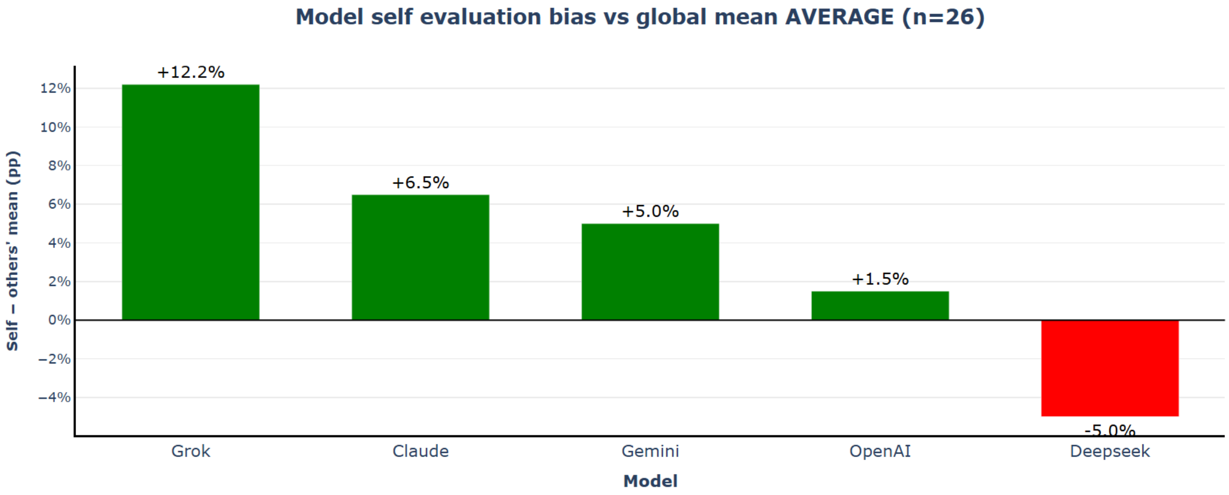


Figure 4: Model self-confidence versus peer assessment, Scenario 1 — self-score minus peer mean.

Figure 4 compares each model's self-evaluation with the average score assigned to it by the other models. *Grok* shows the largest positive deviation (+12.2%), indicating *a strong tendency to rate its own output significantly higher* than its peers do. *Claude* (+6.5%) and *Gemini* (+5.0%) also display notable positive self-bias, while *OpenAI* remains relatively close to peer consensus (+1.5%). In contrast, *DeepSeek* is the only model that scores itself lower than the peer average (−5.0%), suggesting a comparatively conservative self-assessment. This dimension suggests that self-evaluation bias is model-specific and systematic rather than random.

3.2.2.6 Consolidated View

Figure 5 consolidates the preceding analyses into a single comprehensive view: per-model evaluation bands, peer mean scores, and self-evaluation scores presented side by side.

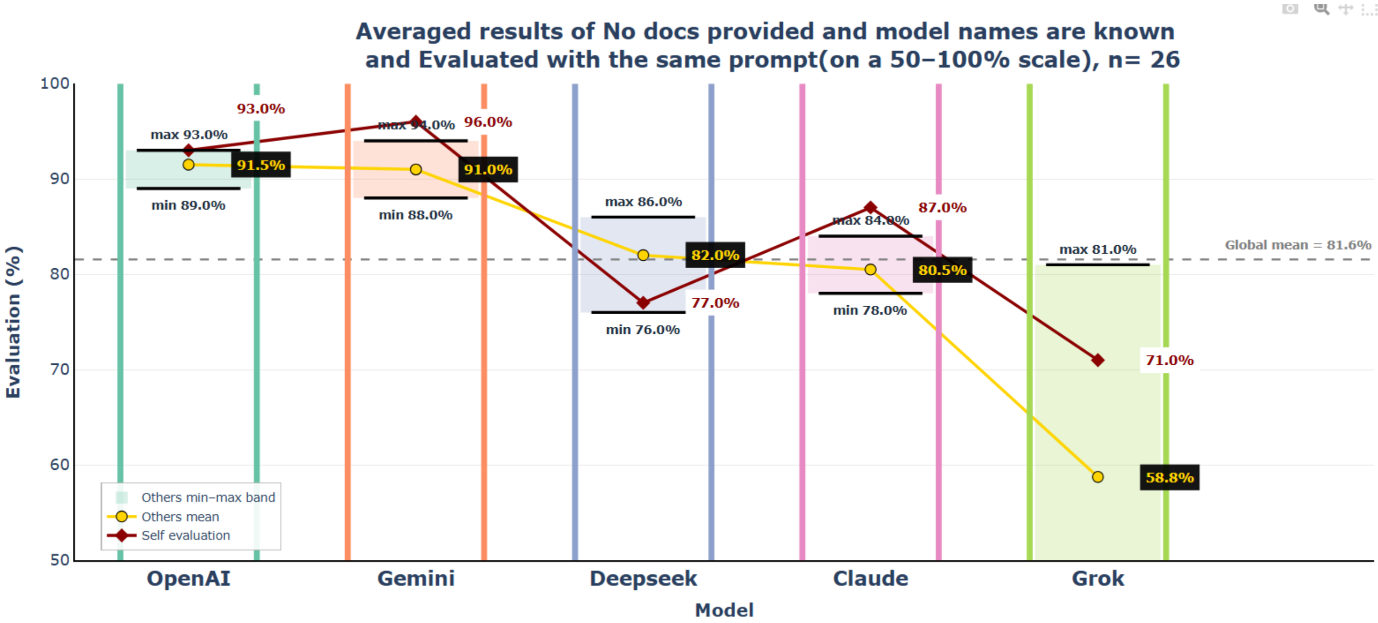


Figure 5: Consolidated view, Scenario 1 — per-model evaluation bands, peer mean scores, and self-evaluation scores.

Figure 5 provides a consolidated view of the evaluation dynamics by combining three elements for each model: the minimum–maximum range of peer scores, the mean score assigned by the other models, and the model's own self-evaluation. This representation highlights both the stability of peer assessments and the calibration of each model's self-perception relative to how it is judged by others.

OpenAI and *Gemini* clearly stand out with the highest peer mean scores (both around 91%) and relatively narrow evaluation bands, indicating consistent and broadly favourable assessments across evaluators. In contrast, *DeepSeek* and *Claude* occupy a middle position with peer means around 80%–82%, showing moderate variation between evaluators. *Grok* appears as the strongest outlier: its peer mean score is substantially lower (58.8%) and its evaluation band remains well below the global average.

Self-evaluation patterns further reveal calibration differences. *Grok* shows the largest positive gap between self-evaluation and peer opinion, while *DeepSeek* displays the opposite tendency by rating itself slightly below the peer mean. *OpenAI* and *Gemini* remain closest to peer consensus, indicating comparatively well-calibrated self-assessment relative to the evaluations provided by the other models.

3.3 Conclusions

The central finding of this Experiment 2, Scenario 1 suggests that model selection for text generation is **not neutral** when the output is evaluated by another LLM, in-line with the overall findings of the Experiment 1. The **spread** between the highest- and lowest-rated models **exceeds 30 percentage** points in peer reputation, which **is large enough to have practical consequences in automated evaluation pipelines** where LLM-generated content is screened, ranked, or filtered without human review.

Across the tested models, **OpenAI and Gemini consistently achieved the strongest peer reputation**, combined with **relatively stable evaluation ranges and well-aligned self-assessments**. This **combination suggests that their outputs are more likely to be favourably evaluated across a variety of LLM judges**.

Other models display more distinctive calibration patterns. **Grok shows a pronounced positive self-evaluation bias**, while **DeepSeek tends to rate its own output more conservatively than its peers do**. **Claude occupies a middle position** with relatively balanced peer evaluations but moderate variance in scoring.

An important caveat is that this experiment was conducted with a specific set of models, model versions, prompts, and task domain. The results should therefore not be interpreted as a universal ranking of model quality. Instead, they provide evidence of a reproducible phenomenon: cross-model evaluation bias, where the perceived quality of generated content can depend significantly on which LLM performs the evaluation.

4 Scenario 2: Evaluation without generators’ name known by evaluators

Scenario 2 Name Blind Control No-One has Expert Knowledge	**Generator receives manual:** ✗ NO **Evaluator receives manual:** ✗ NO	**Model names visible to evaluator:** ✗ NO

This scenario was prepared to answer the question: *does the evaluation outcome change when the evaluator models do not know which generator produced each answer?* In Scenario 1, the identity of the generating model was explicitly provided to the evaluators in the user prompt. Scenario 2 removes this information, replacing model names with anonymous labels, to isolate the effect of generator identity on scoring behaviour.

4.1 Generation Phase: identical with Scenario 1.

4.2 The Evaluation Phase

During this evaluation phase the same solutions have been applied for the technical challenges, as with the first experiment and summarized there under the chapter: *4.2 Technical Challenges.*

4.2.1 Setup

Once the five generated texts were collected, the same five models were used as evaluators — simulating a recursive "LLM-as-Judge" scenario. In this scenario, the evaluator models did not receive the name of the generator models. In the user prompt the generated models are referred from Model A to Model D. The user evaluation prompt, which is modified compared to the previous Scenario 1, is provided in **Appendix E.**

The JSON output format was requested to make further analysis straightforward and to enforce structured, comparable output across all evaluator models.

4.2.2 Evaluation Phase – Results

4.2.2.1 Cross-Model Scoring Matrix

The heatmap below shows the full cross-model scoring matrix: how each model scored every other model, including itself, aggregated across all 27 full runs.

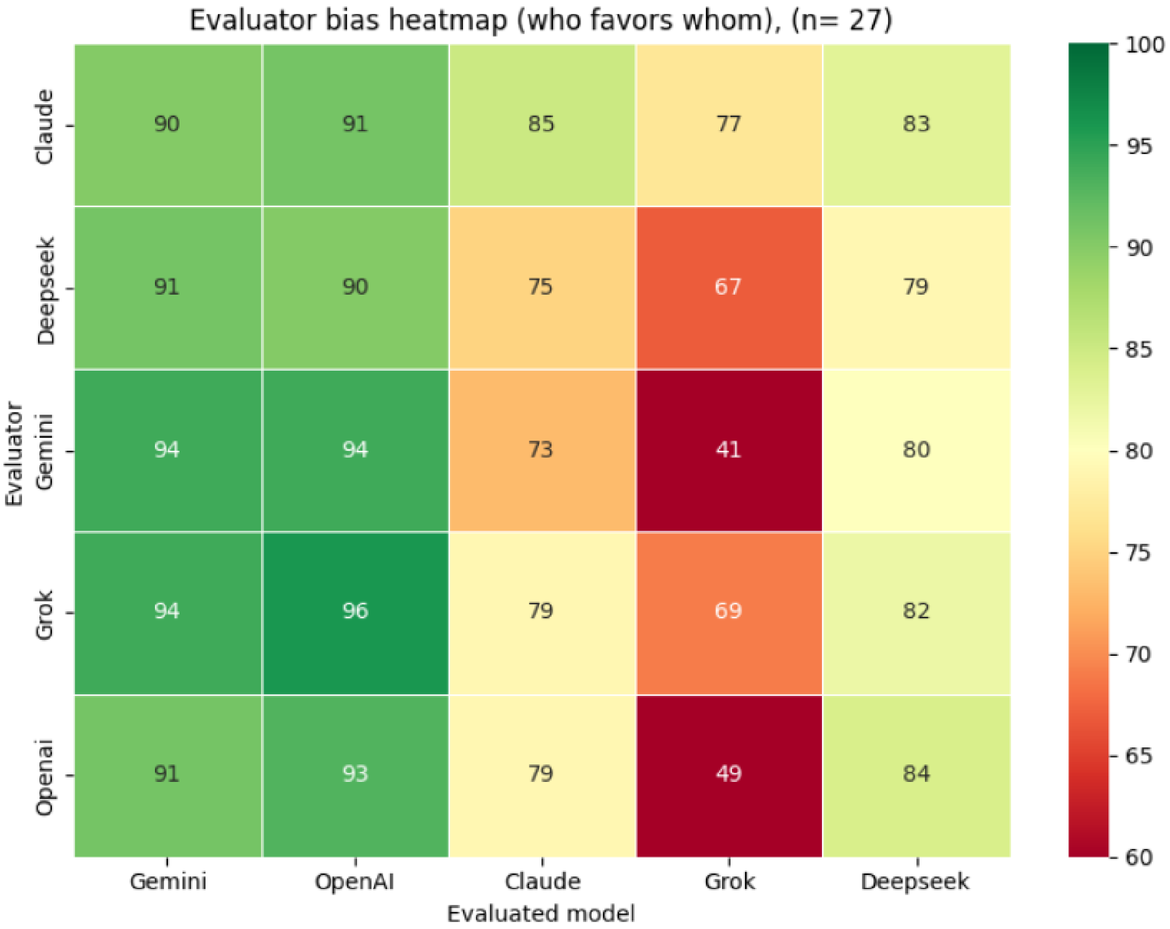


Figure 6: Scenario 2 cross-model scoring matrix — which model favours which model.

When generator identities were hidden from evaluators, the overall scoring pattern remained largely consistent with Scenario 1 — *Gemini* and *OpenAI* continue to *receive the highest peer scores*, while *Grok remains the lowest-rated* model across all evaluators. This suggests that evaluator bias is driven primarily by response quality rather than model reputation, as the rankings are preserved even when the source of each answer is unknown.

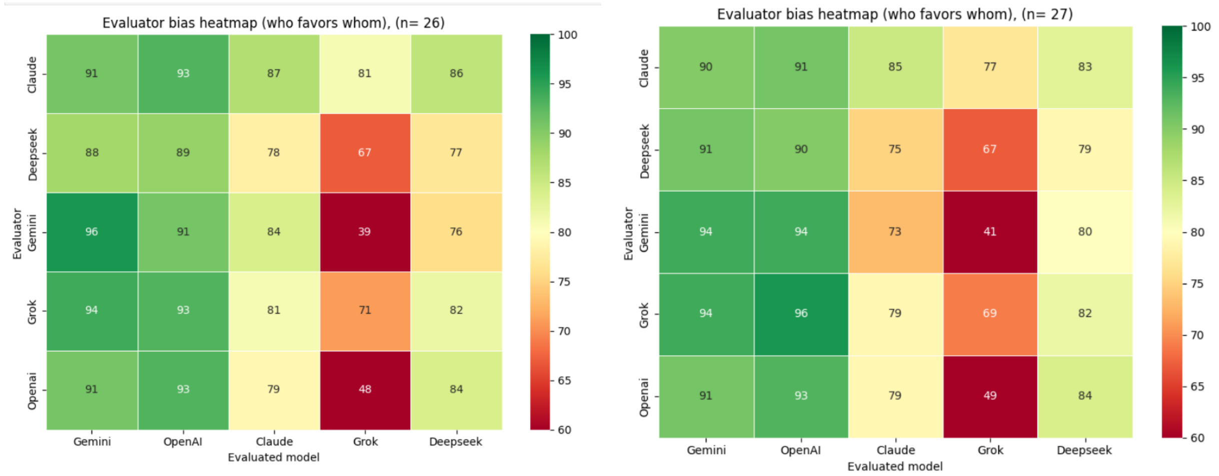


Figure 7: Does Gemini Inflate Scores for Claude When the Generator's Identity Is Known?

4.2.2.2 Scenario 1 vs. Scenario 2 Observations

Comparing the two matrices, the overall ranking structure is stable — *Gemini and OpenAI remain the top-rated models* and *Grok the lowest in both scenarios,* regardless of whether model names were visible to evaluators.

However, one subtle shift is worth noting: Claude's scores drop when names are hidden, with *Gemini's score for Claude falling from 84% to 73% — a 11% difference across 27 runs —* suggesting that *Claude benefits from name recognition specifically in Gemini's evaluation*. Conversely, Grok's scores show no meaningful change when anonymous, indicating that knowing Grok's identity does not further amplify the penalty it receives. The key takeaway is that name visibility has a limited overall effect on scoring patterns with one notable exception, and the cross-model evaluation bias observed in this experiment appears to be grounded predominantly in the content of the responses rather than in model reputation.

4.2.2.3 Evaluator Strictness

The next question is whether the models differ systematically in how strict or lenient they are as evaluators.

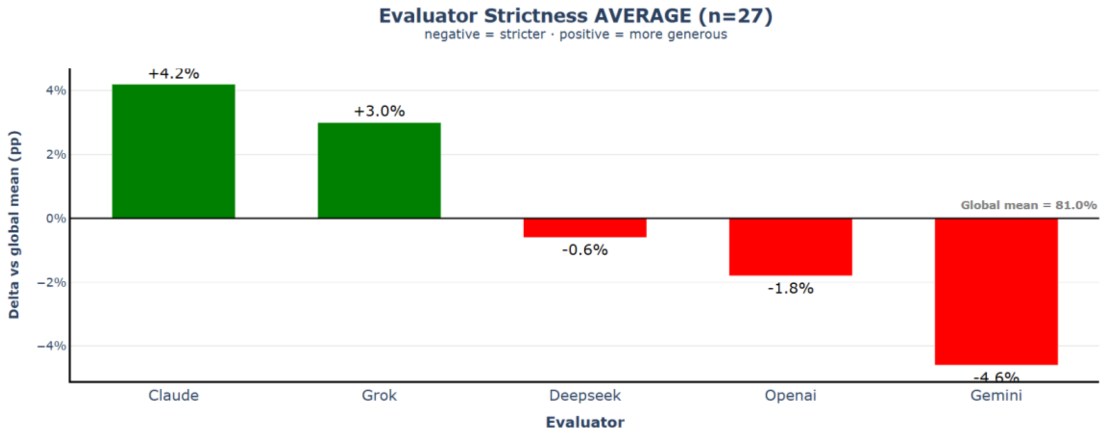


Figure 8: Evaluator generosity, Scenario 2 — how strict or lenient each model was when scoring the others.

Scenario 2 shows only slightly *reduced evaluator bias compared to Scenario 1.* While *Claude* remains the most lenient evaluator in both scenarios, its leniency drops from +6.0% (Scenario 1) to +4.2% (Scenario 2), the difference between the two scenarios is less than 1% in case of the other model and the global mean rises slightly from 81.0% to 81.6%. *The overall spread between the most lenient and strictest evaluator narrows from 10.4 % to 8.8 %, suggesting that hiding generator model names didn't influence the bias across models.*

4.2.2.4 Model Reputation

The following figure addresses the question most relevant to practitioners: *which model should be used if the evaluator model is unknown, so we don't know which model will evaluate our text?*

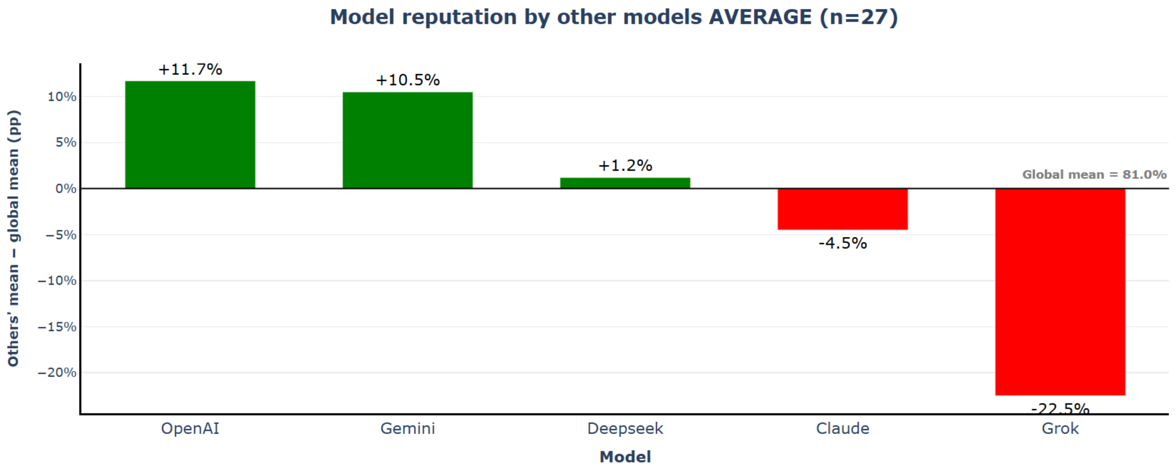


Figure 9: Model reputation, Scenario 2 — how each model is perceived on average by its peers (excluding self-assessment).

Scenario 2 shows increased separation in model reputation compared to Scenario 1. The top-rated models *(OpenAI and Gemini)* receive even higher scores from their peers — OpenAI jumps from +9.9% to +11.7%, and Gemini rises from +9.4% to +10.5% above the global mean. *Meanwhile, the reputation penalty for lower-ranked models intensifies: Claude's* negative reputation worsens from -1.1% to -4.5%, thanks for *Gemini*; and *Grok* remains deeply undervalued with -22.5%. *DeepSeek* also shifts from barely positive (+0.4%) to slightly positive (+1.2%), maintaining its middle position.

4.2.2.5 5.4 Self-Assessment vs. Peer Assessment

A final dimension of this study is the gap between how each model rates its own output and how its peers rate the same output.

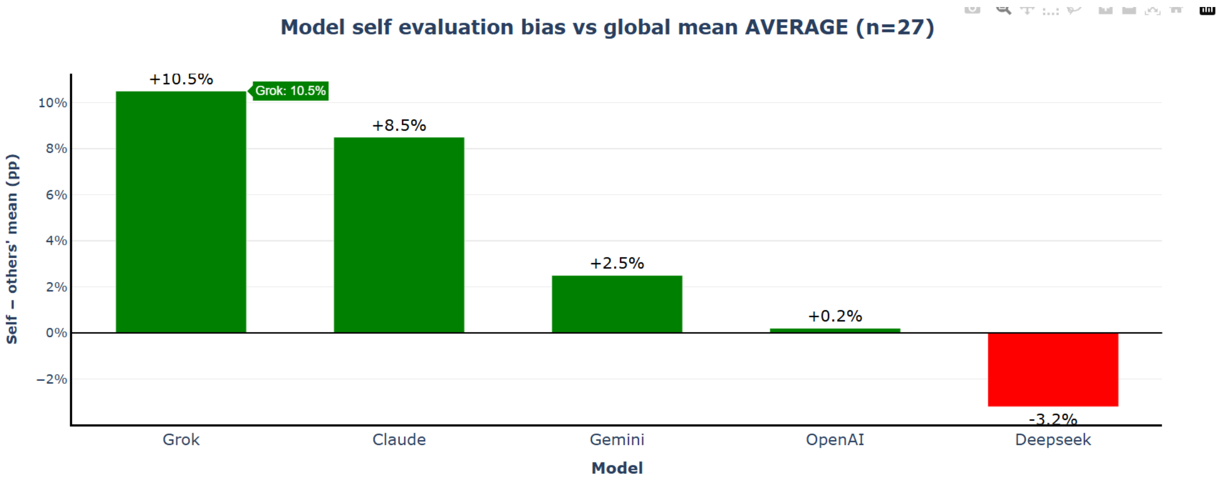


Figure 10: Model self-confidence versus peer assessment, Scenario 2 — self-score minus peer mean.

Scenario 2 shows overall reduced self-evaluation bias across nearly all models compared to Scenario 1. Grok's extreme self-favoritism drops a bit from +12.2% to +10.5%, however Claude's self-bias is increase from +6.5% to +8.5%. Gemini's self-evaluation also decreases from +5.0% to +2.5%, and OpenAI shifts from modest self-favoritism (+1.5%) to near-perfect calibration (+0.2%). Only DeepSeek moves in the opposite direction — from moderate self-criticism (-5.0%) to slight self-criticism (-3.2%), becoming less harsh on itself. Overall, the spread between the most self-favoring (Grok) and most self-critical (DeepSeek) model narrows from 17.2 to 13.7 percentage points, suggesting that without recognising themselves, models become more objective when evaluating their own outputs.

4.2.2.6 Consolidated View

Figure 5 consolidates the preceding analyses into a single comprehensive view: per-model evaluation bands, peer mean scores, and self-evaluation scores presented side by side.

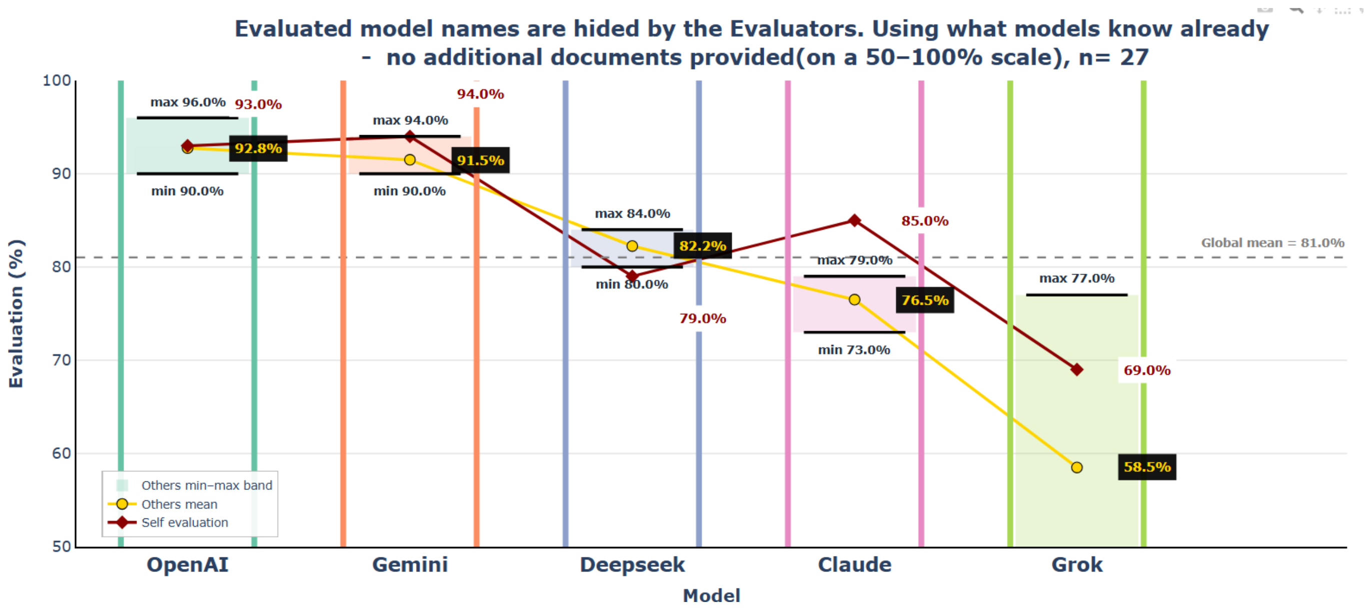


Figure 11: Consolidated view, Scenario 2 — per-model evaluation bands, peer mean scores, and self-evaluation scores.

Figure 11 provides a consolidated view of evaluation dynamics across both scenarios, combining peer score ranges, mean peer assessments, and self-evaluations for each model. In Scenario 2 (no documentation provided, and model names not known), *OpenAI* and *Gemini maintain dominant positions* with peer means around 91%-93%. *DeepSeek* is clearly ahead of *Claude* in the second scenario with gap raised from 1.5% to 5.7% in their reputation. *Claude* lost around 5% of reputation when its name was hidden. *Grok remains the consistent outlier* across both conditions, receiving almost *identical judgment from other models whether or not they knew it was Grok*, with peer means around 58%-59%.

Self-evaluation calibration shifts meaningfully between scenarios. *Grok's self-evaluation gap narrows from approximately 23 % points above peer mean in Scenario 1 to roughly 12 % points in Scenario 2, suggesting that hiding model names reduces its overconfidence. DeepSeek maintains its characteristic self-criticism in both scenarios*, consistently rating itself below peer consensus, while OpenAI suggests the most stable self-calibration across conditions, remaining within 1-2% of its peer mean in both scenarios.

4.3 Conclusions

The central finding of Scenario 2 is that **concealing model identity during evaluation substantially reduces inter-model bias** compared to Scenario 1. When evaluators cannot identify the source of the text they are judging, the spread in evaluator strictness narrows from

10.4% to 8.8% points, and self-evaluation bias decreases from 17.2% to 13.7% points**. This suggests that model name visibility amplifies both evaluator preferences and self-favoritism.**
Despite this overall moderation, the **reputation hierarchy remains largely intact**. *OpenAI* and *Gemini* continue to dominate with peer means around 91%-93%, though their advantage becomes slightly more pronounced — rising from +9.9% and +9.4% above global mean in Scenario 1 to +11.8% and +10.5% in Scenario 2. This amplification indicates that *even without visible attribution, their outputs are recognized and rewarded by peer evaluators.*

The most dramatic shift occurs in the middle tier. *DeepSeek's* reputation advantage over *Claude* widens significantly from 1.5% to 5.7%, driven largely *by Gemini's harsh assessment of Claude (79%) when model names are hidden — a notable downgrade from Scenario 1*. This suggests that Gemini might apply stricter standards when it cannot confirm authorship, which might have some direct reasoning behind.

Grok's position as an outlier persists across both scenarios (evaluated by peers on 58%-59%), but its self-evaluation bias improves markedly when anonymized, dropping from +23% to +12% points above peer consensus. Meanwhile, OpenAI maintains exceptional self-calibration stability (±1-2% from peer mean) regardless of name visibility, and DeepSeek consistently underrates itself relative to peer judgment in both conditions.

These findings demonstrate that **while hiding model names promotes more conservative and calibrated evaluations overall, it does not eliminate reputation effects** — and may even intensify them for top-performing models whose quality signals remain detectable through output characteristics alone.

An important caveat is that this experiment was conducted with a specific set of models, model versions, prompts, and task domain. The results should therefore not be interpreted as a universal ranking of model quality. Instead, they provide evidence of a reproducible phenomenon: cross-model evaluation bias, where the perceived quality of generated content can depend significantly on which LLM performs the evaluation.

5 Scenario 3: Evaluation with extra expert knowledge, generators without extra knowledge

Scenario 3 Evaluator Has Expert Knowledge	**Generator receives manual:** ✗ NO **Evaluator receives manual:** ✓ YES	**Model names visible to evaluator:** ✗ NO

This scenario aims to answer the following question: *how the evaluators' outcome change, when they have the expert knowledge (user manual of the drone) available during the evaluation phase*, but the generators still don't have the specific knowledge available. *One obvious assumption* might be, that the *models would be stricter with their evaluations, because they have the extra knowledge then the generators don't have.*

5.1 Generation Phase: identical with Scenario 1 & 2.

5.2 The Evaluation Phase

During this evaluation phase the same solutions have been applied for the technical challenges, as with the first experiment and summarized there under the chapter: *4.2 Technical Challenges.*

New Technical Challenges

On top of the already known and managed technical challenges, we got a new technical challenge with this scenario.

Attaching a file to the user prompt

While some models (such as Gemini and Claude) advertise native PDF support through their APIs, this capability proved unreliable when processing complex user manuals containing diagrams, safety graphics, and mixed layouts. Other models used in the experiment (OpenAI, DeepSeek, and Grok as I remember from the experiement) did not provide consistent native PDF handling via their APIs and generally required preprocessing, such as extracting the textual content before passing it to the model. To ensure consistent conditions across all five models and eliminate variability caused by different PDF parsing behaviours, all evaluators received the same plain-text version of the DJI Mini 4 Pro manual. This approach was sufficient because the take-off procedure in the manual is primarily described as a sequence of textual steps rather than relying on visual diagrams.

OpenRouter model handling

Another technical challenge was OpenRouter's inconsistent handling of file attachments across different model providers. File passing did not work uniformly for all models through OpenRouter, which effectively removed the main advantage of using OpenRouter as a simplifying abstraction layer. In addition, as already noted in the first experiment, OpenRouter routes requests to different internal variants of the same named model, reducing control over the exact model version being used and therefore affecting the reproducibility of experimental results.

5.2.1 Setup

Once the five generated texts were collected, the same five models were used as evaluators, simulating a recursive "LLM-as-Judge" scenario. In this Scenario 3, the evaluator models did not receive the names of the generator models but *were provided with the complete drone user manual as plain text as injected expert knowledge*. The evaluators' user prompt, which was modified compared to Scenario 1, is provided in **Appendix E.**

A JSON output format was requested to facilitate further analysis and to enforce structured, comparable outputs across all evaluator models.

Token Usage Analysis

Each evaluation call processes approximately 50,000–58,000 tokens, consisting primarily of the 204,613-character user manual (~51,000 tokens), evaluation prompts (~1,500 tokens), and JSON output (~500–2,000 tokens).

Notable variation exists across models and model versions due to different tokenization methods:

OpenAI: 51,332 tokens
DeepSeek: 51,978 tokens
Grok: 52,242 tokens
Gemini: 58,190 tokens
Claude: 58,511 tokens

Selected Claude and Gemini models consume approximately 13% more tokens than selected OpenAI, DeepSeek, or Grok models for identical input text, reflecting the differences in their tokenization algorithms. While this variation impacts both cost efficiency and context window utilization across providers, it should not be considered a critical parameter in isolation when evaluating model performance.

All models handled this token volume without issue, as their context windows (ranging from 128k to 200k+ tokens) comfortably accommodated the ~50–58k token inputs.

5.2.2 Evaluation Phase – Results

We now examine whether the assumption of stricter evaluation holds.

5.2.2.1 Cross-Model Scoring Matrix

The heatmap below shows the full cross-model scoring matrix: how each model scored every other model, including itself, aggregated across all 26 evaluation runs.

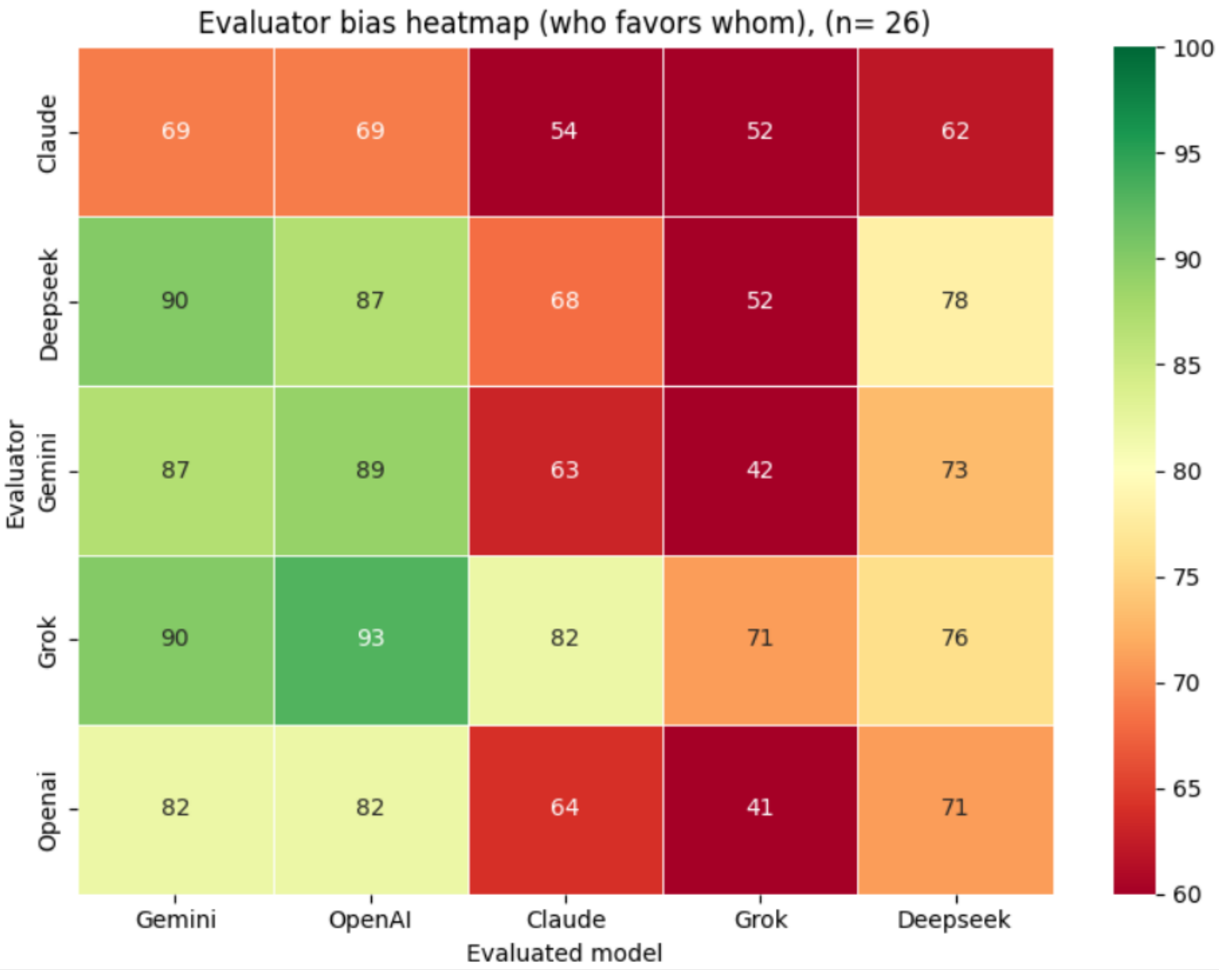


Figure 12: Scenario 3 cross-model scoring matrix — which model favours which model.

The majority of *the values in the matrix dropped significantly from the previous Scenario 2*. *Gemini* and *OpenAI* receive the *highest* scores across all evaluators (69%–93% range), while *Grok* consistently receives the *lowest* ratings (41%–71%), with particularly harsh judgments from Gemini (42%) and OpenAI (41%). *DeepSeek* occupies a *middle position* (62%–78%), and *Claude* scores between 54%–82% across evaluators, indicating a clear quality hierarchy recognized by all models when evaluating against the source documentation.

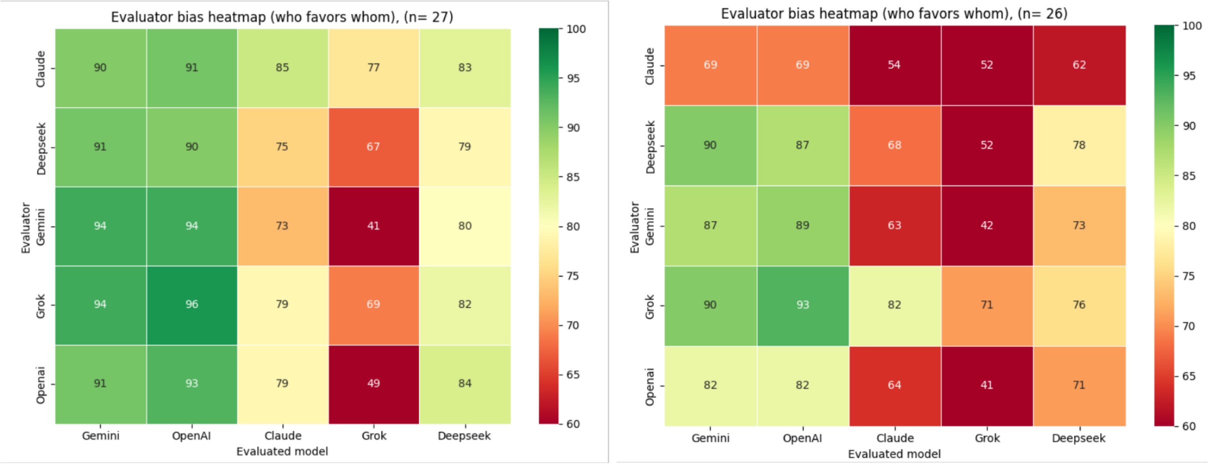


Figure 13: Scenario 2 vs. Scenario3. As we assumed, evaluators got much stricter…

5.2.2.2 Scenario 2 vs. Scenario 3 Observations

When evaluators gain access to the source documentation (Scenario 3*), scores drop dramatically across the board*, *reflecting stricter evaluation based on factual accuracy* rather than response plausibility.

Claude shows the most extreme shift in both roles. As a generator, *it experiences the largest drop in scores*. At the same time, *Claude also becomes the strictest evaluator*. Its scoring across all models, including itself, drops by around 20% in almost every case, from the 77%–91% range to the 52%–69% range. As a result, Claude simultaneously drives down overall scores while also being the most heavily penalized model, receiving the largest decrease from both its own evaluations and those of other models. All models (except Grok) penalize Claude heavily overall, with Claude receiving in total of 29% lower evaluations.

DeepSeek receives 27% lower evaluations without Claude, *OpenAI* is 22% lower without Claude, and *Gemini* is 21% lower without Claude. *Grok* receives only 20% lower without Claude, although its baseline was already the worst.

At the top tier, *Gemini and OpenAI* also decline significantly, from the 90%–94% range to 69%–93%, largely due to Claude's much stricter evaluations. However, they *remain the strongest-performing models*, suggesting their responses maintain relatively high factual accuracy even without access to source material.

DeepSeek also shows clear degradation, dropping from 79%–84% (Scenario 2) to 62%–78% (Scenario 3), indicating partial factual accuracy but clear gaps when evaluated against the documentation.

The key finding: The magnitude of score drop reflects how much a *model's responses rely on plausible but factually incorrect content*. *Claude* (−29%) and *DeepSeek* (−27%) show the largest declines, *indicating that their answers often appear convincing under plausibility-based evaluation but fail factual verification* when explicit reference material is introduced. Whereas OpenAI (−22%) and Gemini (−21%) remain more robust and consistent under ground-truth evaluation, and Grok (−20%) shows a smaller decline largely due to its already low baseline, highlighting that plausibility-based evaluation can mask underlying factual weaknesses that only become visible when objective verification is applied.

5.2.2.3 Evaluator Strictness

When evaluators receive source documentation, evaluator strictness patterns shift dramatically compared to previous scenarios. The global mean *drops substantially by 10%*, to 71.5% (compared to 81.6% in Scenario 2 and 81.0% in Scenario 1), reflecting overall stricter evaluation when models can verify responses against ground truth.

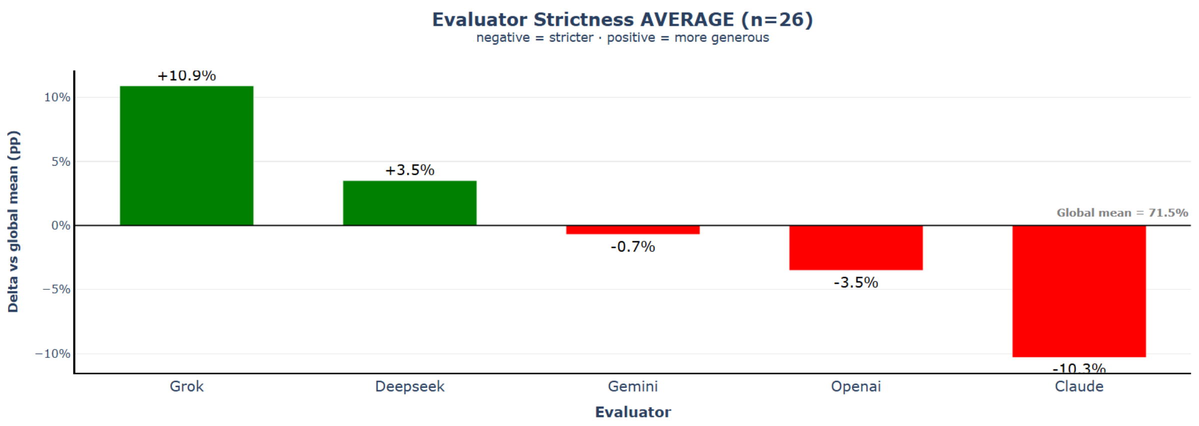


Figure 14: Evaluator generosity, Scenario 3 — how strict or lenient each model was when scoring the others.

The most dramatic shift occurs with *Claude*, which become significantly stricter: *Claude drops to -10.3% (compared to +4.2% in Scenario 2*). OpenAI to -3.5% (compared to -1.8% in Scenario 2). Grok emerges as the most lenient evaluator at +10.9% above the global mean — a reversal from its previous middle-tier position. DeepSeek maintains moderate generosity at +3.5%, while Gemini becomes nearly calibrated at -0.7% (compared to -4.6% in Scenario 2).

The evaluator strictness spread increases to *21.2 %* (from 8.8% in Scenario 2), s*uggests that the models did not fully possess or reliably access the required domain knowledge prior to receiving the source document. The introduction of ground truth enables more accurate verification, revealing errors that were previously undetected.*

5.2.2.4 Model Reputation

Despite the 10% drop in absolute scores, the model reputation hierarchy remains virtually unchanged. *OpenAI* and *Gemini maintain their dominant positions* at +13.0% and +11.3% above the global mean respectively — nearly identical to Scenario 2 (+11.7% and +10.5%).

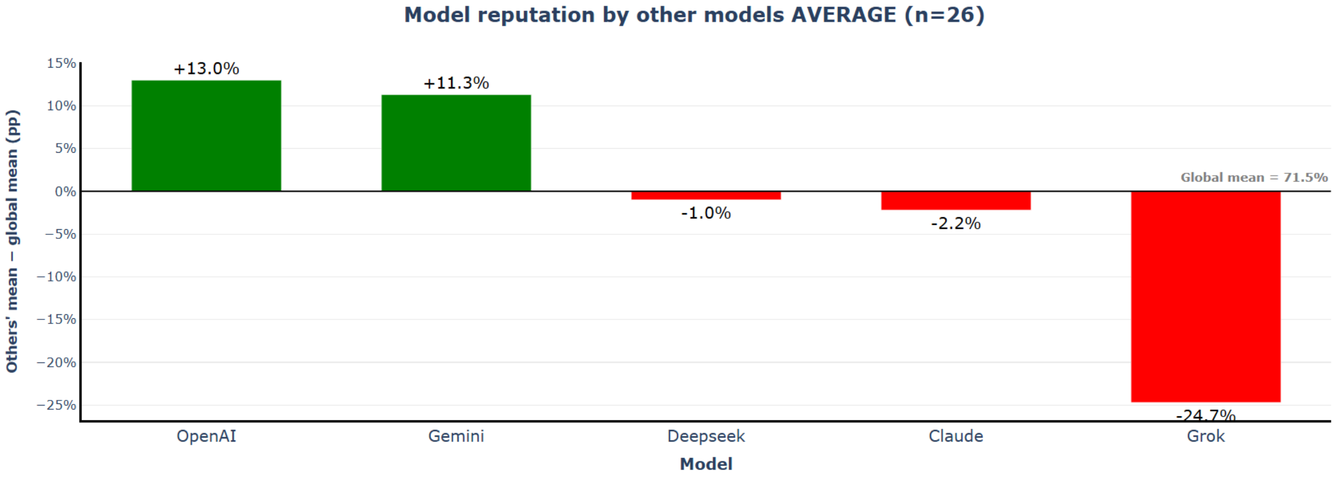


Figure 15: Model reputation, Scenario 3 — how each model is perceived on average by its peers (excluding self-assessment).

The lower tier also shows stability: Grok remains the most undervalued model at -24.7% (compared to -22.5% in Scenario 2), and Claude stays in negative territory at -2.2% (compared to -4.5% in Scenario 2). DeepSeek maintains its middle position at -1.0%, barely shifted from its +1.2% in Scenario 2. *The key finding: providing evaluators with ground truth documentation does not alter the reputation hierarchy* — it simply recalibrates all scores downward with model-specific variation. This suggests that the *quality differences between models are robust and detectable regardless of whether evaluators assess based on plausibility (Scenario 2 with common knowledge) or factual accuracy (Scenario 3 with specific, expert knowledge).*

5.2.2.5 Self-Assessment vs. Peer Assessment

When evaluators gain access to source documentation, self-evaluation bias patterns change dramatically.

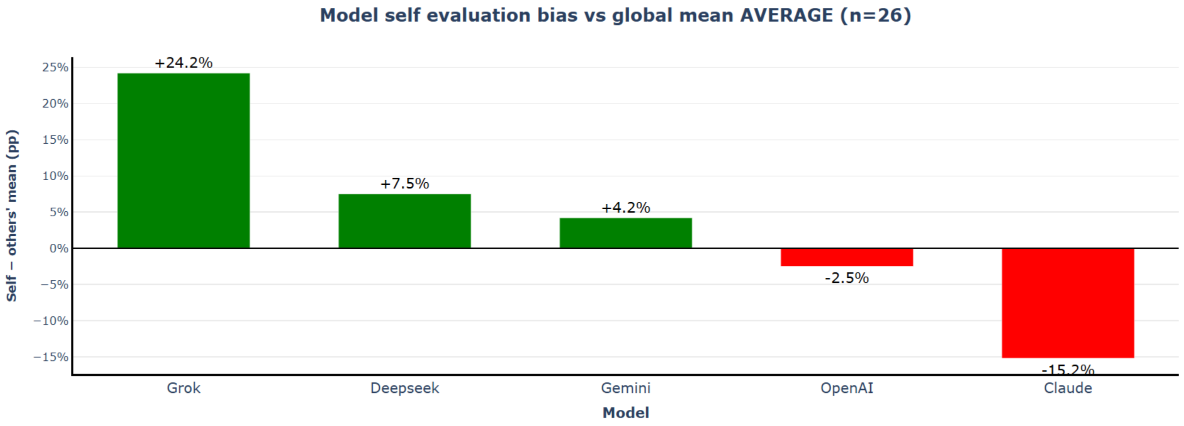


Figure 16: Model self-confidence versus peer assessment, Scenario 3 — self-score minus peer mean.

Grok's self-favouritism more than doubles from +10.5% (Scenario 2) to +24.2% (Scenario 3), but the most interesting point is that *Claude becomes significantly more self-critical* with a 23.7%, now at -15.2% (compared to +8.5% in Scenario 2).
DeepSeek shifted toward self-favouritism and raised its self-bias from -3.2% (Scenario 2) to +7.5% (Scenario 3), moving from self-criticism to moderate self-favoritism. *Gemini* maintains relatively stable self-assessment at +4.2% (compared to +2.5% in Scenario 2), remaining the most calibrated among top-tier models. *OpenAI* shifts to moderate self-criticism at -2.5% (from near-perfect calibration at +0.2% in Scenario 2). *The self-evaluation bias spread explodes to 39.4% (from 13.7% in Scenario 2), suggesting that access to ground truth documentation amplifies rather than reduces calibration differences.*

5.2.2.6 Consolidated View

Figure 17 consolidates the preceding analyses into a single comprehensive view: per-model evaluation bands, peer mean scores, and self-evaluation scores presented side by side.

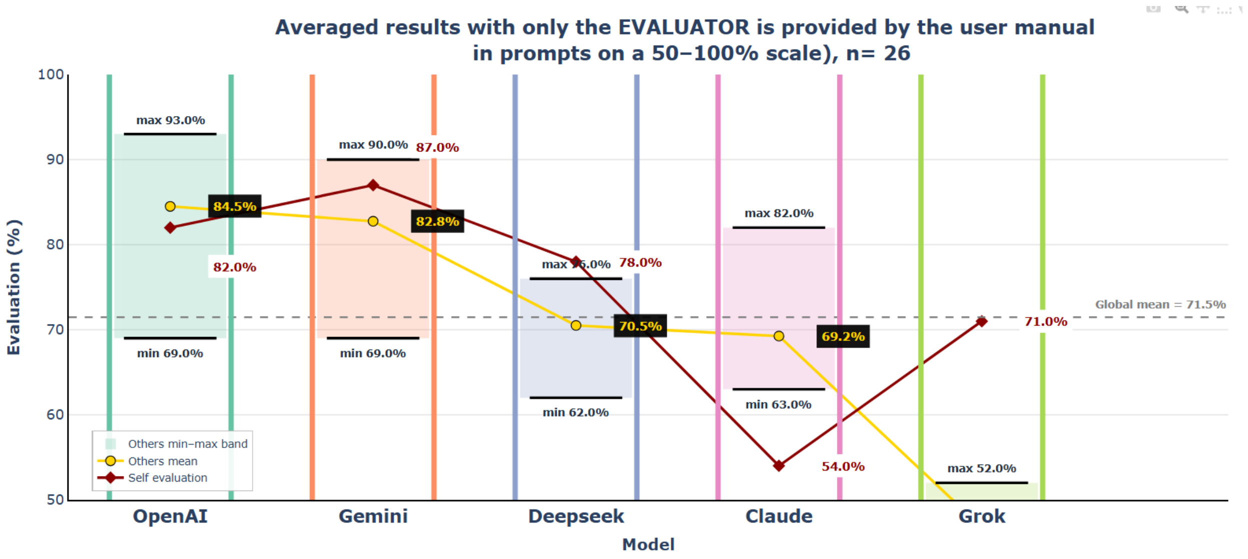


Figure 17: Consolidated view, Scenario 3 — per-model evaluation bands, peer mean scores, and self-evaluation scores.

When evaluators gain access to source documentation, not only do all scores drop by approximately 10%, but the evaluation bands show divergent patterns: top-tier models' ranges expand dramatically (OpenAI from 6% to 24%, Gemini from 4% to 21%), *indicating greater evaluator disagreement when assessing high-quality output*, while Grok's band narrows from 29% to 11%, suggesting stronger consensus on poor performance. Yet the ranking order remains completely unchanged, with *OpenAI and Gemini at the top and Grok at the bottom.* The most striking transformation occurs in self-evaluation behaviour: the red diamond lines (self-scores) track closely to peer means in Scenario 2 but diverge wildly in Scenario 3, with Grok's self-score rising far above its peer mean (71% vs 58.5%) *while Claude's plunges far below (54% vs 69%),* indicating that Claude takes verification extremely seriously and becomes a harsh self-critic when it can compare against ground truth, while Grok remains lenient to others and overconfident with itself regardless of access to factual documentation.

5.3 Conclusions

The central finding of Scenario 3 is that **providing evaluators with source documentation fundamentally transforms evaluation dynamics while preserving the core reputation hierarchy**. When evaluators can verify factual accuracy, scores drop by approximately 10 percentage points, yet OpenAI and Gemini retain their top-tier status, and Grok remains the lowest-rated model — demonstrating that quality differences are **robust to both anonymization and verification methodology**.

However, **access to ground truth *amplifies* rather than reduces evaluator behaviour differences**. The evaluator strictness spread explodes from 8.8% (Scenario 2) to 21.2% (Scenario 3), and self-evaluation bias range widens from 13.7% to 39.4% points. This suggests that **individual models respond very differently to the presence of verification material**: some (like Claude) become dramatically more critical across all evaluations, while others (like Grok) become even more self-favoring despite having ground truth available.

The practical implication **for automated evaluation pipelines is clear: *model selection matters significantly when ground truth is available to evaluators***, as the choice of evaluator can shift absolute scores by 20+ percentage points even when ranking order remains stable. ***Organizations deploying LLM-as-Judge systems with access to source documentation should carefully calibrate their evaluator choice based on their tolerance for strict versus lenient assessment.***

An important caveat is that this experiment was conducted with a specific set of models, model versions, prompts, and task domain. The results should therefore not be interpreted as a universal ranking of model quality. Instead, they provide evidence of a reproducible phenomenon: cross-model evaluation bias, where the perceived quality of generated content can depend significantly on which LLM performs the evaluation.

6 Scenario 4: Generators with extra expert knowledge, evaluators without extra knowledge

Scenario 4 Generator Has Expert Knowledge	**Generator receives manual:** ✓ YES **Evaluator receives manual:** ✗ NO	**Model names visible to evaluator:** ✗ NO

The Scenario 4 seeks to answer the question: how do evaluator outcomes change when they lack expert knowledge (the drone user manual) during evaluation, but generators have that specific knowledge available? One assumption might be that evaluations would be much higher than in Scenario 3, when generators did not have access to the specific knowledge. *It would be interesting to see whether evaluators can recognize and appreciate presumably higher-quality answers without having the source material themselves.*

6.1 Generation Phase

In the generation phase, a two-part user prompt was used, consisting of task instructions and the full user manual provided as plain text. The same technical solutions were applied to address known challenges as in the previous Scenarios. The generators' user prompt, which was extended by the user manual is provided in **Appendix G**.

Runtime and Output Size

The table below summarizes response latency and output length across models when 50–60k tokens were injected into the user prompt, based on API-generated outputs recorded during the experiment.

Model Label	Latency via API calls	Output Size	Observations
Gemini	~ 8 sec	~390 words	Fastest model while producing relatively detailed instructions.
OpenAI	~12 sec	~350 words	Moderate latency with consistent mid-length outputs.
DeepSeek	~19 sec	~300 words	Highest latency despite relatively concise responses.

Claude	~16 sec	~380 words	Balanced latency and relatively detailed responses.
Grok	~15 sec	~260 words	Shortest responses despite relatively high token usage.

Key Finding 1: Adding approximately 50,000 input tokens (the user manual) resulted in negligible latency increases across all models (< 1 seconds), even in some cases (OpenAI, Grok) latency decreased slightly. *This suggests that modern LLMs handle large context windows efficiently, with generation speed primarily determined by output length rather than input size* — even when input tokens approach 25-30% of the maximum context window capacity (typically 128k-200k tokens).

Key Finding 2: The repeated runs demonstrate that model-specific behaviour is stable and reproducible, *with each model consistently applying its characteristic style* (e.g. Gemini = structured/manual-like, OpenAI = contextual, DeepSeek = concise, Claude = balanced, Grok = tutorial-like). This reinforces the earlier finding that differences between models are not due to randomness, but rather reflect systematic and persistent stylistic and structural preferences in how procedural knowledge is generated.

A qualitative comparison of the generated instructions is presented in **Appendix F**.

6.2 The Evaluation Phase

The evaluation phase in Scenario 4 is identical to the evaluation phase of the baseline scenario (Scenario 1), where no additional expert knowledge (user manual) was provided to the evaluators.

6.2.1 Evaluation Phase – Results

We now examine whether the expected increase in evaluation scores is reflected in the results.

6.2.1.1 Cross-Model Scoring Matrix

The heatmap below shows the full cross-model scoring matrix: how each model scored every other model, including itself, aggregated across all 29 evaluation runs.

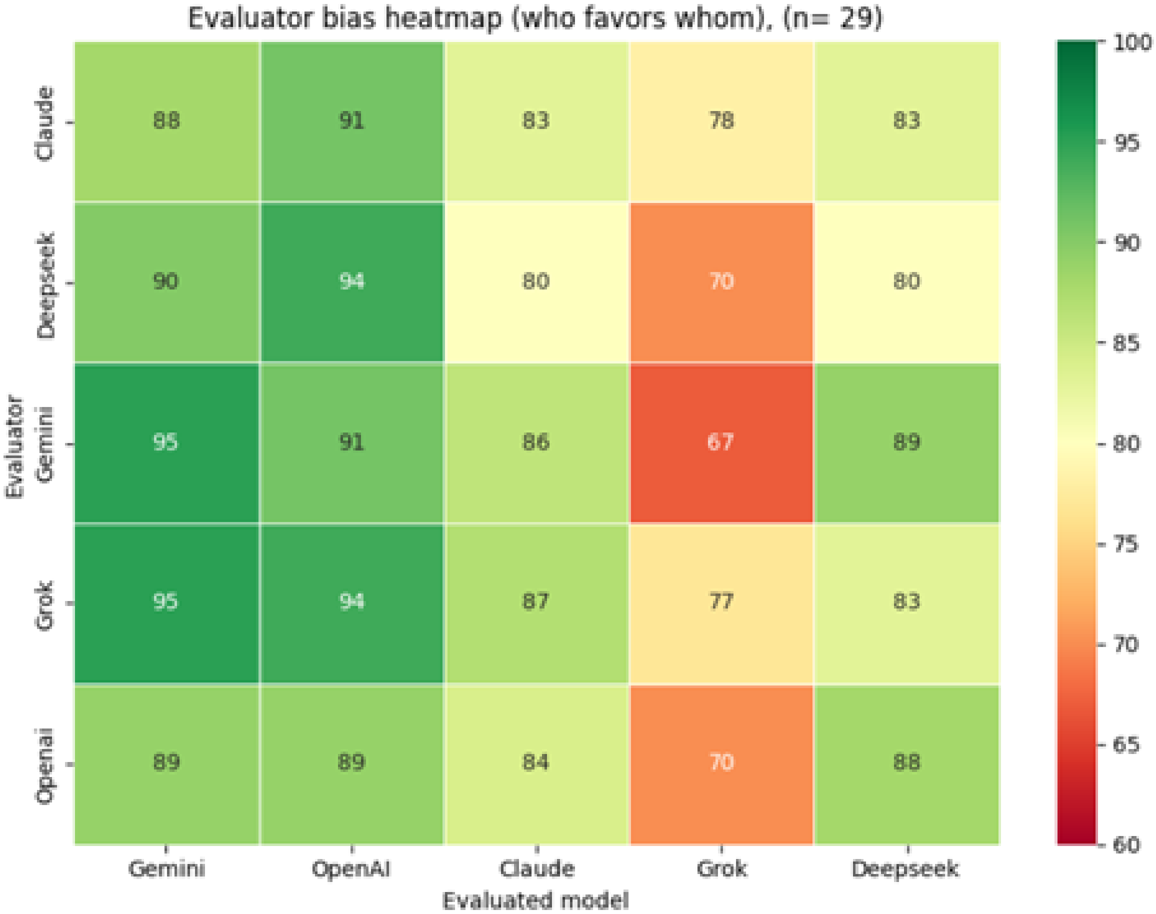


Figure 18: Scenario 4 cross-model scoring matrix — which model favours which model.

When generators have access to source documentation, but evaluators do not, *scores rise dramatically across the board compared to Scenario 3*. *Gemini* and *OpenAI dominate* with scores in the 89%–95% range across all evaluators. *Claude* and *DeepSeek* receive *mid-tier* evaluations (80%–89%), while *Grok* remains the *lowest*-rated at 67%–78%, though it still receives notably higher scores than in previous scenarios. The matrix shows predominantly green colouring with minimal red cells, indicating broad consensus that all models produced high-quality outputs.

Compared to Scenario 2 (where neither generators nor evaluators had documentation), scores rise moderately but consistently by 5%-15%, suggesting that *evaluators assign higher scores to higher-quality, factually grounded outputs even without being able to verify the information themselves.*

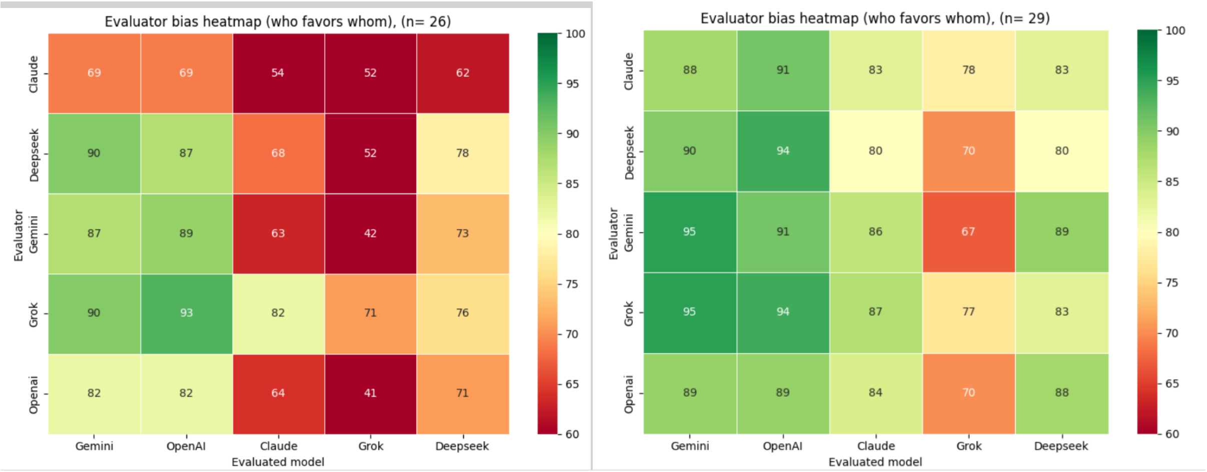


Figure 19: Scenario 3 vs. Scenario4. Evaluation scores increase in Scenario 4, consistent with prior expectations.

6.2.1.2 Scenario 3 vs. Scenario 4 Observations

When generators gain access to source documentation while evaluators lose it (Scenario 4), *scores rise dramatically across the entire matrix*, creating an almost inverse transformation of the Scenario 2→3 shift. The most striking reversal occurs in the previously struggling models: *Grok's scores surge from 41%–71% (Scenario 3) to 67%–78% (Scenario 4*), and *Claude jumps from 54%–82% to 80%–87%* — indicating that access to ground truth during generation produces outputs that *evaluators recognize as high-quality even without being able to verify against the source themselves.*

The top tier (*Gemini* and *OpenAI*) reaches near-peak performance, with scores rising from 69%–93% (Scenario 3) to 88%–95% (Scenario 4), suggesting that when these models have source documentation, their outputs become almost universally recognized as excellent regardless of evaluator knowledge. DeepSeek shows substantial improvement from 62–78% to 80–89%, moving from mid-tier to mid-high tier.

The comparison of the two data sets suggests that providing generators with accurate source material elevates output quality so substantially that evaluators without access to that material still rate responses 15%–25% higher, *indicating that models are able to recognize and reward higher-quality outputs based on internal signals of plausibility and correctness, even without direct access to ground truth.*

6.2.1.3 Evaluator Strictness

When generators have access to source documentation, but evaluators do not*, the global mean rises sharply to 84.8%* (compared to 71.5% in Scenario 3), and evaluator behaviour converges dramatically: the *strictness spread collapses from 21.2% to just 4.4%,* with all evaluators clustering within 2.4 percentage points of the mean.

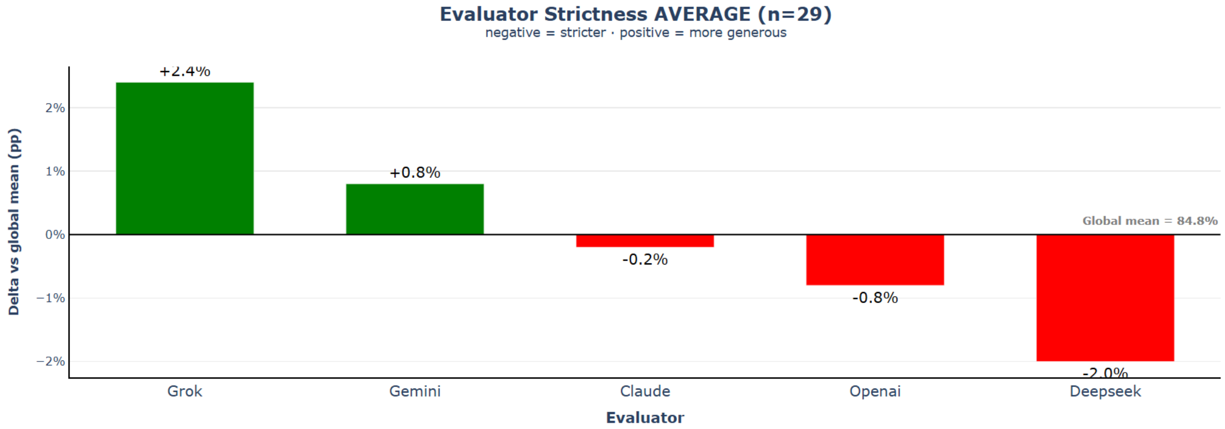


Figure 20: Evaluator generosity, Scenario 4 — how strict or lenient each model was when scoring the others.

The most striking transformation occurs with *Claude*, which shifts from extreme strictness (-10.3% in Scenario 3) to near-perfect calibration (-0.2%), while *DeepSeek* reverses from lenient (+3.5%) to the strictest evaluator (-2.0%).

This convergence suggests that evaluators do not merely recognize quality improvements in generated outputs, but also consistently accept them: ***when generators rely on factual source material, the resulting responses leave little room for subjective interpretation, reducing uncertainty and limiting the evaluators' "degrees of freedom."*** As a result, even without direct access to the source documentation, evaluators converge in their judgments and reliably reward higher-quality outputs without needing to infer or approximate correctness.

6.2.1.4 Model Reputation

Even as the global mean rises to 84.8%, the reputation hierarchy remains stable, with *OpenAI* (+7.7%) and *Gemini* (+5.7%) maintaining their top positions, while *DeepSeek* (+0.9%) and *Claude* (-0.6%) converge near the global mean, and *Grok* (-13.6%) stays at the bottom.

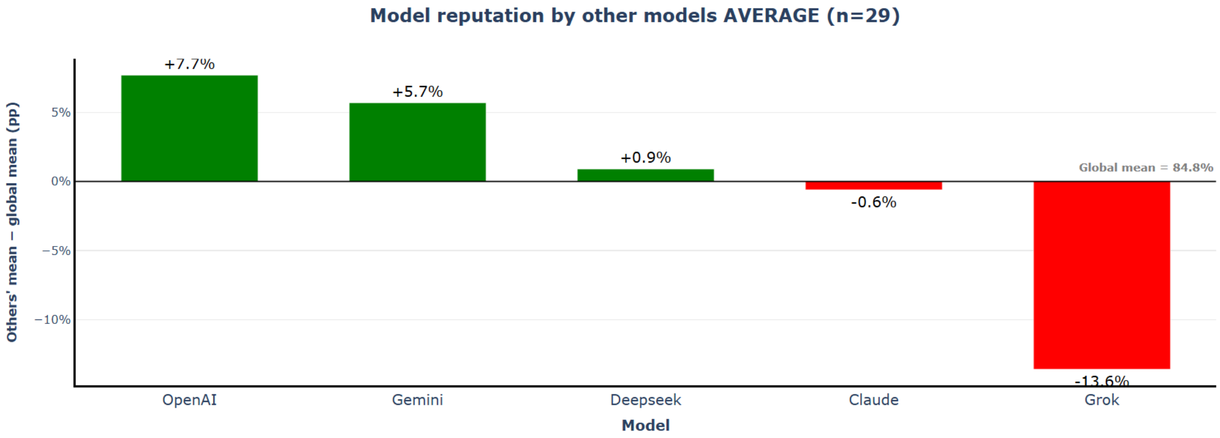


Figure 21: Model reputation, Scenario 4 — how each model is perceived on average by its peers (excluding self-assessment).

However, the *reputation spread narrows significantly from 37.7% (Scenario 3) to 21.3%,* with Grok improving by 11% (−24.7% → −13.6%), showing that access to source material substantially improves even lower-performing models. For example, Grok's peer scores increase from ~47% (Scenario 3) to ~71% (Scenario 4), a 24% gain. *Despite this overall improvement, evaluators still preserve the same quality hierarchy (OpenAI/Gemini > DeepSeek/Claude > Grok), consistent with earlier scenarios.*

6.2.1.5 Self-Assessment vs. Peer Assessment

When generators gain access to source documentation, but evaluators do not, *self-evaluation bias patterns compress dramatically compared to Scenario 3, with the spread narrowing from 39.4% to just 11.6%*.

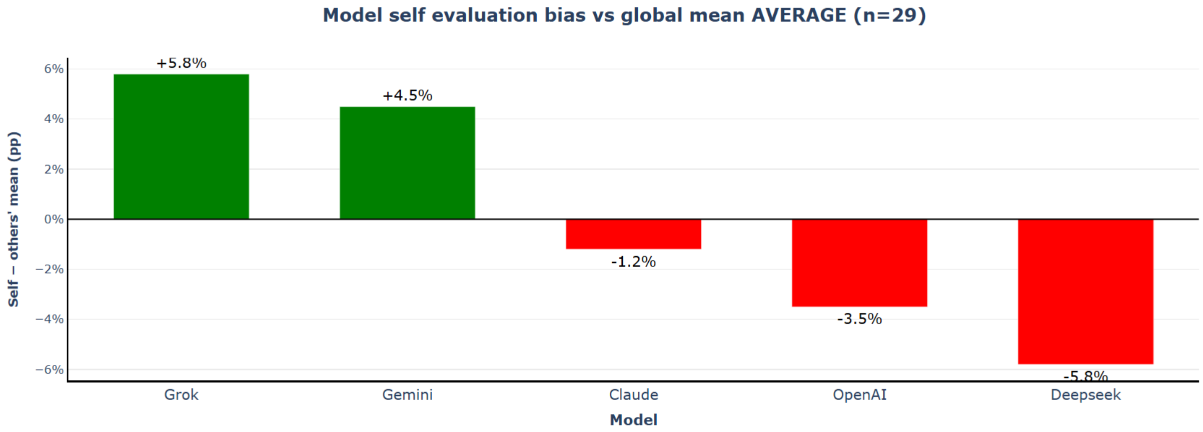


Figure 22: Model self-confidence versus peer assessment, Scenario 4 — self-score minus peer mean.

Grok's self-favouritism drops from +24.2% to +5.8%, *Claude's* extreme self-criticism reverses from -15.2% to near-perfect calibration at -1.2%, and *DeepSeek* shifts from moderate self-favouritism (+7.5%) to self-criticism (-5.8%), becoming the harshest self-critic. *Gemini* maintains stable moderate self-favouritism at +4.5% (compared to +4.2% in Scenario 3), and *OpenAI* shows modest self-criticism at -3.5% (compared to -2.5% in Scenario 3).

The compression may occur *because higher and more uniform output quality reduces evaluators' ability to differentiate between responses*, leading to more consistent scoring and therefore a narrower self-evaluation bias spread.

6.2.1.6 Consolidated View

Figure 23 consolidates the preceding analyses into a single comprehensive view: per-model evaluation bands, peer mean scores, and self-evaluation scores presented side by side.

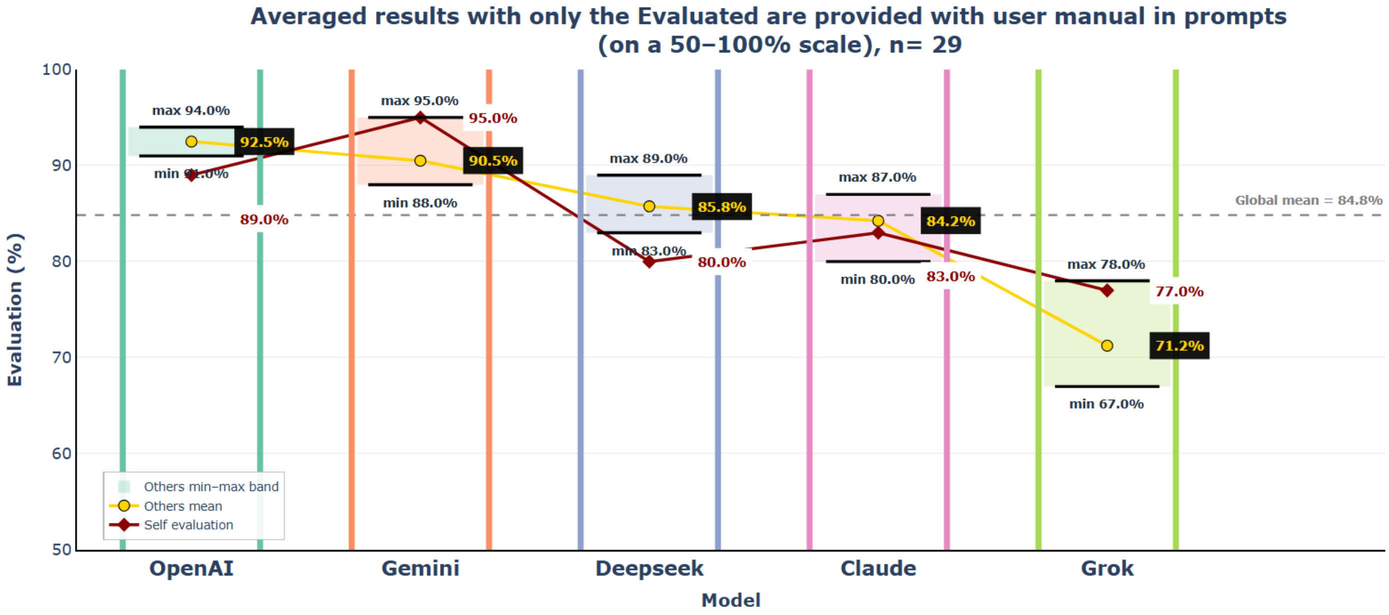


Figure 23: Consolidated view, Scenario 4 — per-model evaluation bands, peer mean scores, and self-evaluation scores.

When generators gain access to source documentation while evaluators lose it (Scenario 4*), all scores rise by approximately 13 percentage points* (global mean from 71.5% to 84.8%), and evaluation bands compress significantly across all models. OpenAI's range narrows from 24% to 6%, Gemini's from 21% to 7%, and even Grok's already-narrow band (11%) remains stable at 11%, further reinforcing the observation of stronger evaluator consensus when assessing factually grounded outputs - evaluators agree more readily on quality when generators work from source material, even without being able to verify claims themselves. The ranking order remains completely unchanged, with OpenAI and Gemini at the top and Grok at the bottom, consistent with all previous scenarios.

6.3 Conclusions

The central finding of Scenario 4 is *that providing generators with source documentation while withholding it from evaluators produces the highest scores and strongest evaluator consensus observed across all scenarios*. When generators work from factual material, scores rise to 84.8% (compared to 71.5% in Scenario 3 and 81.0% in Scenario 2*), yet the reputation hierarchy remains perfectly stable with OpenAI and Gemini at the top and Grok at the bottom* — demonstrating that ***evaluators can not only reliably detect quality differences in outputs without verification material, but also recognize and reward the quality improvement that results from generator access to source documentation.*** *This finding challenges the assumption that evaluators require ground truth to assess factual accuracy: when generators produce well-researched outputs, evaluators consistently assign higher scores even without being able to verify specific claims.*

Another finding is that generator access to documentation dramatically reduces evaluator disagreement: evaluation band spreads compress from 21-24% (Scenario 3) to just 6-7% across top-tier models, and evaluator strictness spread collapses from 21.2% to 4.4%. **This reinforces the previous observation that evaluators can recognize quality improvements without verification** — not only do they assign higher scores to factually grounded outputs, but they also achieve much stronger consensus when doing so.

Although the self-bias spread decreases from 39.4% to 11.6%, this reduced self-bias spread is not caused by uniform improvement in self-evaluation, but by a combination of rising peer scores and heterogeneous self-evaluation shifts across models, including both under-adjustment (Grok, DeepSeek) and strong recalibration (Claude).

The practical implication: for automated evaluation pipelines, providing source documentation to generators is more valuable than providing it to evaluators — it elevates output quality detectably, reduces evaluator disagreement, and produces more stable scoring patterns, even when evaluators cannot verify claims against ground truth.

An important caveat is that this experiment was conducted with a specific set of models, model versions, prompts, and task domain. The results should therefore not be interpreted as a universal ranking of model quality. Instead, they provide evidence of a reproducible phenomenon: cross-model evaluation bias, where the perceived quality of generated content can depend significantly on which LLM performs the evaluation.

7 Scenario 5: Generators and evaluators both with extra expert knowledge

Scenario 5 Both Have Expert Knowledge	**Generator receives manual:** ✓ YES **Evaluator receives manual:** ✓ YES	**Model names visible to evaluator:** ✗ NO

In this final scenario, both generators and evaluators have access to the specific expert knowledge — the user manual of the DJI Mini 4 Pro drone. Potential assumptions might be that while evaluators would be stricter (as observed in Scenario 3), the quality of generated outputs would also be higher (as observed in Scenario 4). These two opposing forces — elevated generation quality versus stricter evaluation standards — could theoretically cancel each other out, producing results identical to our baseline Scenario 2, where neither party had the documents.

One interesting question is which force dominates during the process: the generators' high-quality answers that evaluators recognize and reward, or the evaluators' strict assessments when they can verify what the correct answer should have been? .

7.1 Generation Phase: identical with Scenario 4

In the generation phase, a two-part user prompt was used, consisting of task instructions and the full user manual provided as plain text. The same technical solutions were applied to address known challenges as in the Scenario 4.

7.2 Evaluation Phase

The evaluation phase setup is identical with Scenario 3, where the user manual was shared with the evaluators. During this evaluation phase the same solutions have been applied for the technical challenges, as with the first experiment and summarized there under the chapter: 4.2 Technical Challenges.

7.2.1 Evaluation Phase – Results

We now examine which factors dominate the evaluation process.

7.2.1.1 Cross-Model Scoring Matrix

The heatmap below shows the full cross-model scoring matrix: how each model scored every other model, including itself, aggregated across all 28 evaluation runs.

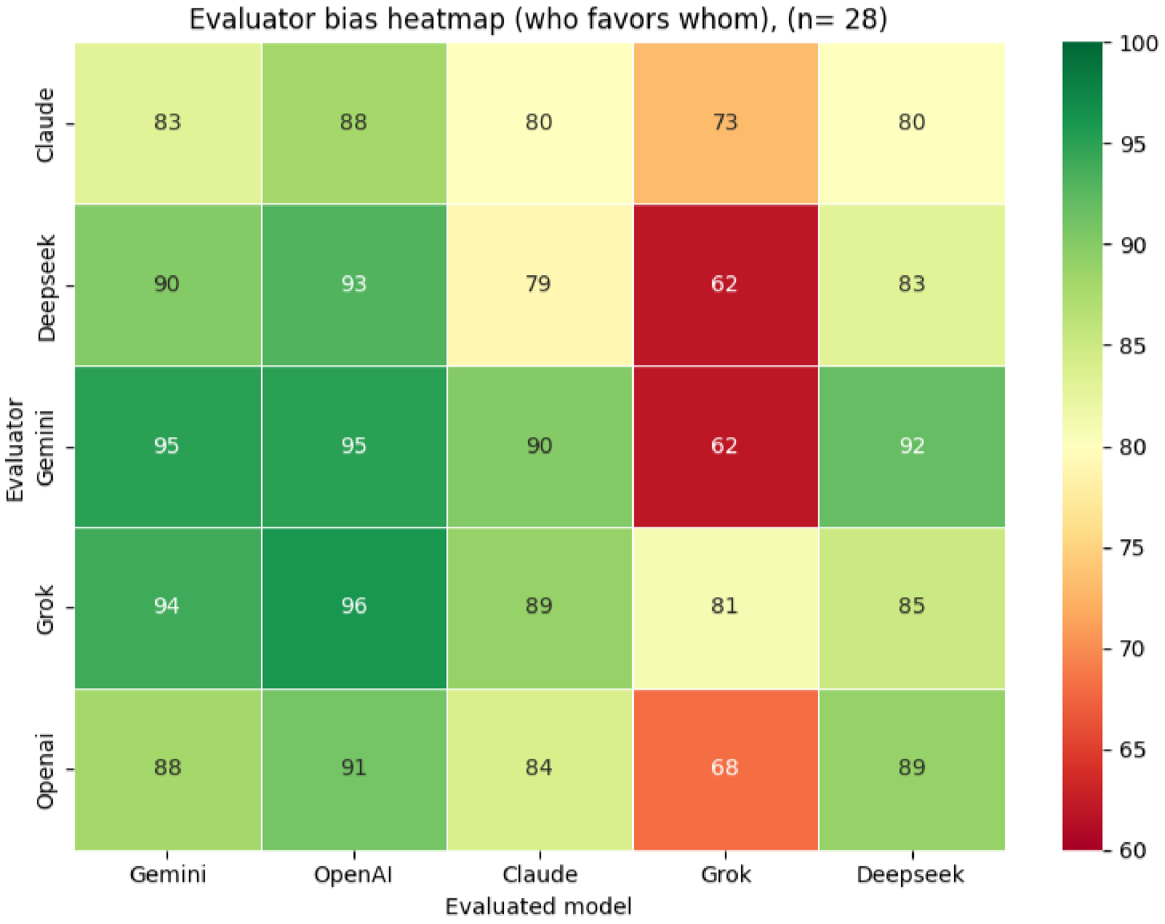


Figure 24: Scenario 5 cross-model scoring matrix — which model favours which model.

When both generators and evaluators have access to source documentation, *scores remain nearly identical to Scenario 4 (global mean 84.4% vs 84.8%),* demonstrating that ***evaluator access to verification material has negligible impact when generators produce factually grounded outputs***. *Gemini* and *OpenAI continue to dominate* with scores in the 88%–96% range across all evaluators, *Claude and DeepSeek remain in the mid-tier* at 79%–92%, while *Grok* experiences the only slight *decline* to 62%–81% (compared to 67%–78% in Scenario 4), with particularly harsh judgments from Gemini (62%) and DeepSeek (62%) when both can verify against ground truth.

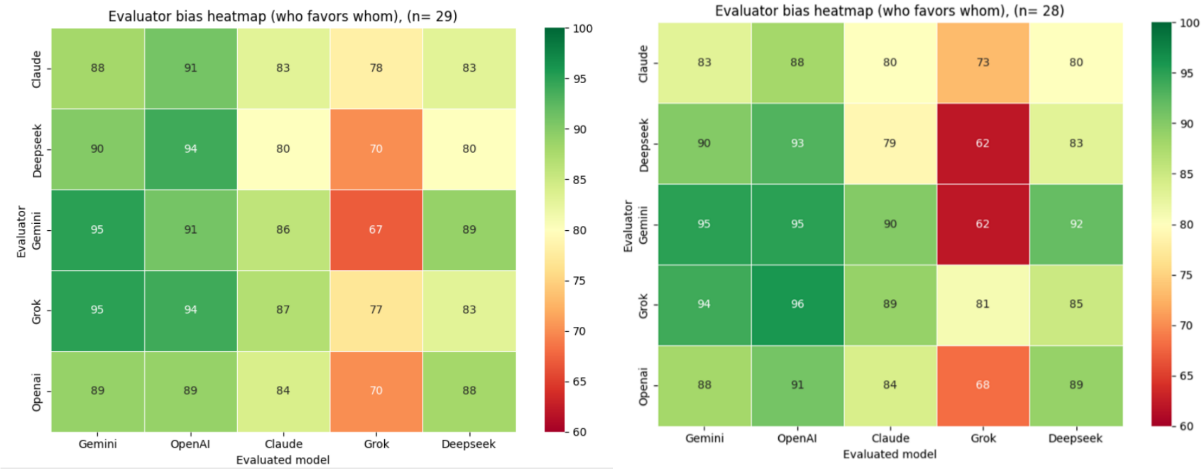


Figure 25: Scenario 4 vs. Scenario5. Very similar values, except Grok's

7.2.1.2 Scenario 4 vs. Scenario 5 Observations

The most notable change occurs with Grok. This targeted strictness shows that verification-based evaluation disproportionately penalizes lower-quality outputs, while higher-quality models are less affected, though not entirely unchanged. In Scenario 3, even top-performing models (e.g., OpenAI and Gemini) experience measurable score reductions, indicating that strict evaluation impacts all models, but to different extents depending on output quality.

The key finding is that generator quality dominates evaluator strictness. When generators operate without source documentation (Scenario 2→3), adding evaluator verification leads to substantial score reductions (−9.5%), as errors and inconsistencies are exposed. In contrast, when generators already rely on source documentation (Scenario 4→5), introducing evaluator verification has a negligible effect (−0.4%), as outputs are already factually grounded.

This asymmetry may suggest ***that verification primarily acts as an error-detection mechanism****: it strongly penalizes responses that contain inaccuracies or gaps, while having only limited impact on responses that are already factually correct.*

7.2.1.3 Evaluator Strictness

In Scenario 5, the *evaluator strictness spread nearly doubles from 4.4% (Scenario 4) to 8.2%,* revealing that *verification capability reintroduces individual evaluator bias patterns*. *Claude reverts to significant strictness (*-3.6%, compared to near-perfect calibration at -0.2% in Scenario 4), while *Grok* and *Gemini* become more lenient (+4.6% and +2.4% respectively, up from +2.4% and +0.8%).

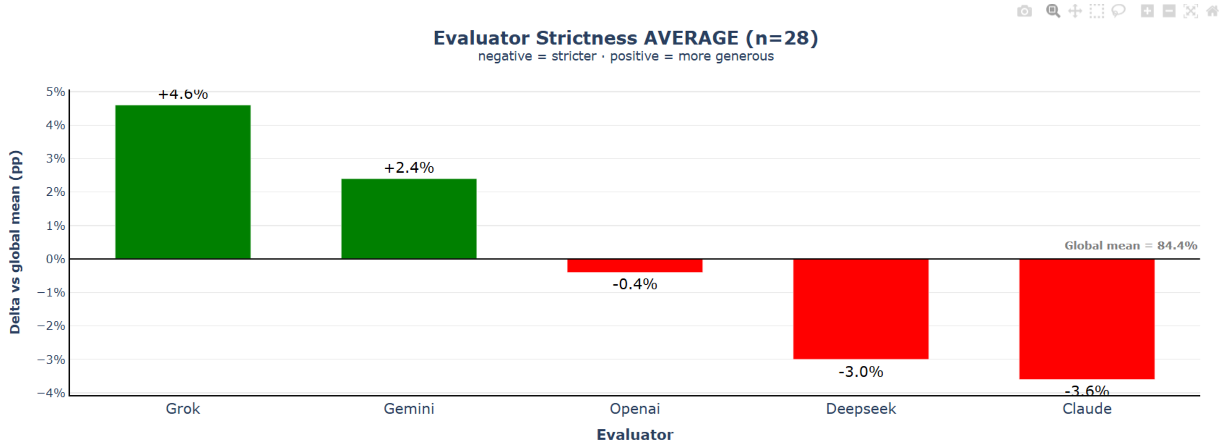


Figure 26: Evaluator generosity, Scenario 5 — how strict or lenient each model was when scoring the others.

This reinforces the earlier observation that verification primarily acts as an error-detection mechanism*: it strongly penalizes responses that contain inaccuracies or gaps, while having only limited impact on factually correct outputs*. As a result, *when evaluators can verify answers against ground truth, their differences in judgment become more visible rather than reduced.* Some models, like Claude, apply this error detection more strictly, while others, like Grok, remain more lenient. This pattern is consistent with Scenario 3, where access to documentation did not eliminate differences between evaluators but instead made them more apparent.

7.2.1.4 Model Reputation

In Scenario 5, the *reputation spread widens from 21.3% (Scenario 4) to 26.8%,* driven primarily by *Grok's significant decline* from -13.6% to -18.2%, while the reputation remains relatively stable for other models.

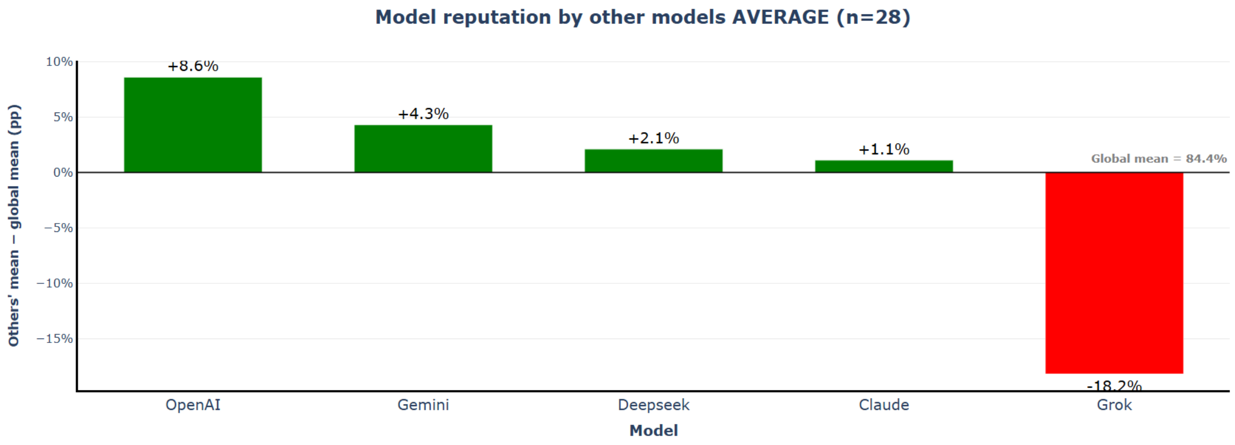


Figure 27: Model reputation, Scenario 5 — how each model is perceived on average by its peers (excluding self-assessment).

This is *consistent with the earlier observation that verification acts as an error-detection mechanism:* when evaluators can compare responses to ground truth, they penalize lower-quality outputs more strongly, while higher-quality outputs remain largely unaffected. As a result, differences between models become more pronounced. This pattern mirrors Scenario 3, where access to verification material similarly amplified quality differences.

7.2.1.5 Self-Assessment vs. Peer Assessment

When both generators and evaluators have access to source documentation (Scenario 5*), the self-evaluation bias spread nearly doubles from 11.6% (Scenario 4) to 20.3%*, driven by dramatic divergence at both extremes: *Grok's self-favouritism explodes from +5.8% to +14.8%* (+9 percentage points), while *Claude's self-criticism intensifies from -1.2% to -5.5%* (-4.3 percentage points).

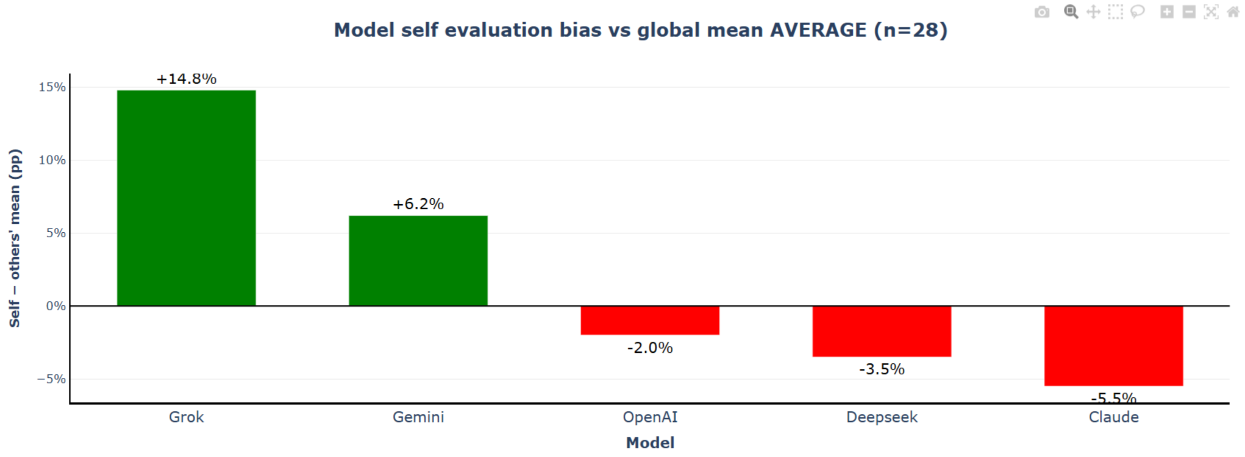


Figure 28: Model self-confidence versus peer assessment, Scenario 5 — self-score minus peer mean.

Grok's increased self-favouritism is driven by a growing divergence between peer and self-evaluation with a self-score of 81% compared to a peer average of 66.2%, indicating that while other models judge its outputs as relatively weaker, Grok continues to rate its own responses as high quality. Since the evaluation is anonymized, the effect is not caused by model recognition, but by how models process their own outputs versus others: Grok evaluates its own responses using the same logic that generated them and therefore misses certain errors, while other models — especially when verification is available — apply stronger error detection and identify these issues more effectively, leading to a widening gap between self and peer scores.

7.2.1.6 Consolidated View

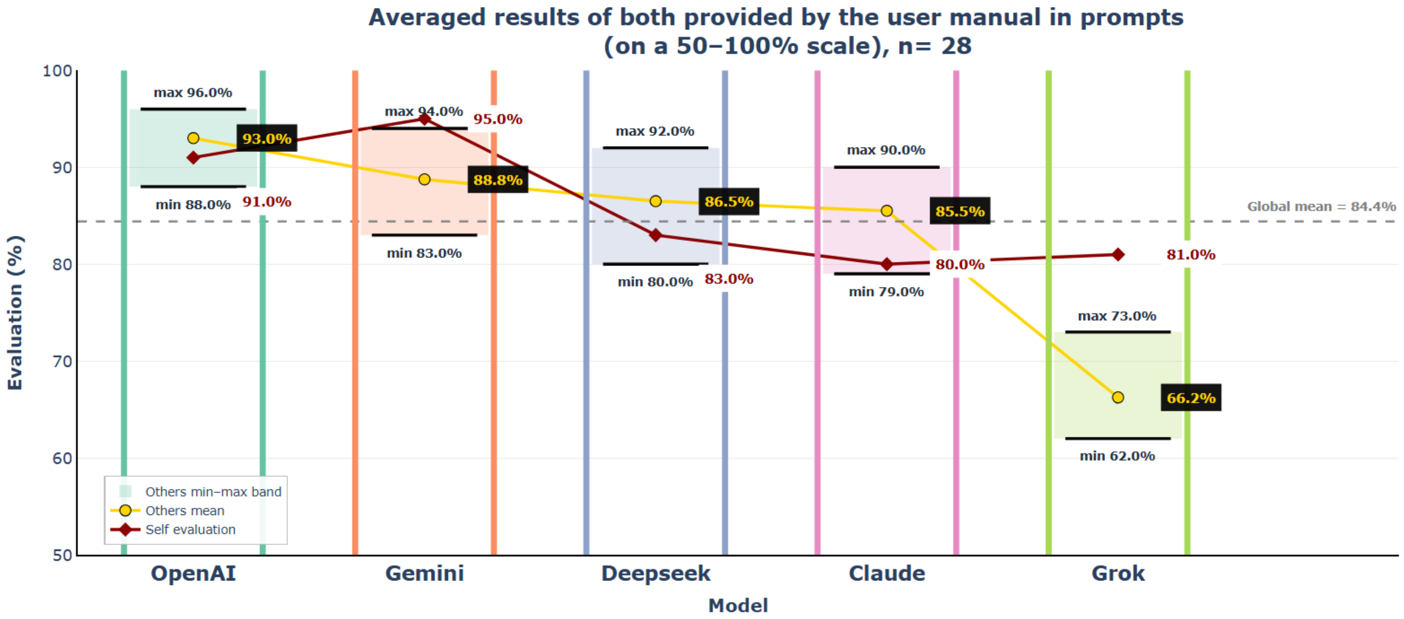


Figure 29: Consolidated view, Scenario 5 — per-model evaluation bands, peer mean scores, and self-evaluation scores.

In Scenario 5, evaluation bands *widen moderately compared to Scenario 4*, particularly for mid-tier and lower-tier models. *OpenAI's* range expands from 6% to 8%, *Gemini's* from 7% to 12%, *DeepSeek's* from 7% to 12%, *Claude's* from 7% to 11%; while *Grok's* band remains at 11% but shifts toward lower absolute values. The most notable change appears in self-evaluation: whereas in Scenario 4 self-scores closely tracked peer averages, Scenario 5 shows clear divergence, with Grok's self-score increasing and Claude's decreasing.

7.3 Conclusions

The central finding of Scenario 5 is that adding evaluator verification to an already grounded generation process (Scenario 4→5) *produces minimal change in average scores but makes evaluator-specific bias patterns more visible*.

While scores remain nearly unchanged (84.8% → 84.4%, −0.4pp), evaluator strictness spread increases (4.4% → 8.2%), reputation spread widens (21.3% → 26.8%), and self-evaluation bias spread nearly doubles (11.6% → 20.3%). This indicates that access to ground truth does not standardize evaluation but instead makes differences in evaluator behaviour more pronounced — for example, Claude becomes stricter while Grok remains more lenient, leading to sharper quality differentiation.

demonstrating that ***ground truth access enables evaluators to apply individual judgment styles more assertively*** — Claude becomes harsher, Grok becomes more lenient, and quality differentiation sharpens (Grok's reputation drops -4.6pp while others remain stable).

Across Scenarios 2–5, the results indicate *that generator quality is the primary driver of evaluation outcomes, while evaluator verification acts asymmetrically as an error-detection mechanism*. When generators operate without source documentation (Scenario 2→3), introducing evaluator verification leads to substantial score reductions (−9.5%) by exposing errors. When generators are provided with source material (Scenario 3→4), scores increase significantly as output quality improves, even without evaluator verification. In contrast, when both generators and evaluators have access to the same documentation (Scenario 4→5), adding verification has only a limited effect on scores (−0.4%), as outputs are already factually grounded.

The practical implication is that generator access to documentation appears to be the primary driver of evaluation outcomes. Providing verification material to evaluators when generators already have access yields limited impact on average scores, but increases variability and reveals evaluator-specific biases, making it less impactful than ensuring that generators operate from accurate source material.

An important caveat is that this experiment was conducted with a specific set of models, model versions, prompts, and task domain. The results should therefore not be interpreted as a universal ranking of model quality. Instead, they provide evidence of a reproducible phenomenon: cross-model evaluation bias, where the perceived quality of generated content can depend significantly on which LLM performs the evaluation.

APPENDIX B Generator Prompts

The following single user prompt was sent to all five models during the generation phase. No system prompt was used.

```
"""Tell me the exact steps about how to take off a DJI_Mini_4_Pro drone. STRICT
LIMIT: Maximum 500 words total response"""
```

Appendix C – Scenario 1: Structural Characteristics of the Generated Instructions

C.1 Procedural Structure

Across all models, the responses showed a strong structural convergence. Despite stylistic differences, nearly all outputs followed a similar three-stage procedural template:

1. physical preparation of the drone
2. system readiness checks (power-on, GPS acquisition, safety verification)
3. execution of the takeoff procedure

This structural convergence suggests that large language models share a broadly similar internal representation of procedural tasks.

C.2 Instruction Style Differences

While the overall structure was similar, the models differed in how they expressed the procedure.

Gemini consistently produced highly explicit, step-by-step instructions resembling a technical manual. The responses emphasised precise operational sequencing and included detailed preparation steps before takeoff.

OpenAI responses tended to integrate operational advice directly into the procedural flow. Instead of strictly numbered steps, the instructions often included contextual guidance such as safe launch location selection and short stability checks after liftoff.

DeepSeek generated the most concise procedural explanations. The responses focused on essential steps while omitting some of the extended contextual guidance included by other models.

Claude typically produced balanced responses combining procedural steps with safety reminders and operational checks, resulting in relatively detailed but well-organised instructions.

Grok tended to adopt a tutorial-like approach, integrating regulatory reminders, troubleshooting advice, and additional contextual information alongside the basic takeoff procedure.

C.3 Scope Management

Another difference between the models concerned how broadly they interpreted the task.

Some responses focused strictly on the physical steps required to launch the drone, while others expanded the scope to include additional safety checks, regulatory considerations, or troubleshooting advice.

These variations illustrate different strategies for balancing **procedural precision and contextual completeness**.

C.4 Output Consistency

Despite the stylistic variation, all models produced technically coherent and usable instructions. The primary differences were therefore not related to correctness but rather to how the models organised and prioritised procedural information.
When analysed individually, each response appears correct and usable. The differences become most visible only when the outputs are compared side by side, revealing distinct stylistic preferences in how procedural knowledge is structured and presented.

APPENDIX D Evaluation Prompts I.

Scenario 1: Baseline Evaluation without External Reference Material

User Prompt of the *Generator Role*

The following user prompt was sent to all five models in their generator role:

```
"""Tell me the exact steps how to take off a DJI_Mini_4_Pro drone. STRICT LIMIT:
Maximum 500 words total response"""
```

System Prompt of the *Evaluator Role*

The following system prompt was sent to all five models in their evaluator role:

```
You are an independent and objective expert evaluator with deep knowledge of DJI
Mini 4 Pro drone operations.

TASK:
Evaluate how suitable each answer is for executing the given task, using your expert
knowledge of proper drone operation procedures.

EVALUATION APPROACH:
- Compare each answer against best practices for DJI Mini 4 Pro takeoff procedures
- You must ONLY evaluate what is written in the answers - do not add missing steps
in your evaluation
- Do not complete or improve the answers
- Identify gaps, inaccuracies, or unclear instructions

EVALUATION CRITERIA:
- Completeness: Are all required steps covered?
- Accuracy: Is the answer technically correct?
- Executability: Could someone realistically execute the task using only this
answer?
- Clarity: Are there critical ambiguities or missing details?
- Safety: Are important safety steps included?

SCORING:
- Assign a percentage score (0-100%) to each answer
- 100% = fully executable, accurate, complete, and safe
- 0% = not executable or dangerously incorrect

OUTPUT FORMAT (STRICT):
```

```
Return ONLY a valid JSON object:
{
  "results": [
    {
      "answer_id": "A",
      "score_percent": 85,
      "strengths": ["Clear step sequence", "Includes safety checks"],
      "weaknesses": ["Missing gimbal protector removal", "No GPS requirement mentioned"]
    }
  ]
}

JSON RULES:
- Use double quotes (") for strings
- Apostrophes need NO escaping: "doesn't" not "doesn\\'t"
- No text before or after the JSON
- Ensure valid, parseable JSON
```

User Prompt of the *Evaluator Role*

The following user prompt was sent to all five models in their evaluator role:

```
1. TASK DESCRIPTION IS THE FOLLOWING: {TASK_DESCRIPTION}

2. DIFFERENT GPT MODEL ANSWERS TO EVALUATE ARE THE FOLLOWING, ALWAYS GO IN STRICT ORDER WITH THE ANSWERS IN THE EVALUATION AS LISTED BELOW:

Answer Gemini:
{gemini_response.candidates[0].content.parts[0].text}

Answer ChatGPT:
{openai_response.choices[0].message.content}

Answer Claude:
{claude_response.content[0].text}

Answer Grok:
{grok_response.choices[0].message.content}

Answer DeepSeek:
{deepseek_response.choices[0].message.content}
```

APPENDIX E Evaluation Prompts II.

Scenario 2: Evaluation without generators’ name known by evaluators

User Prompt of the *Evaluator Role*

The following user prompt was sent to all five models in their evaluator role:

```
USER_PROMPT = f"""

1. TASK DESCRIPTION IS THE FOLLOWING: {TASK_DESCRIPTION}

2. DIFFERENT GPT MODEL ANSWERS TO EVALUATE ARE THE FOLLOWING, ALWAYS GO IN STRICT
ORDER WITH THE ANSWERS IN THE EVALUATION AS LISTED BELOW:

Answer A:
{gemini_response.candidates[0].content.parts[0].text}

Answer B:
{openai_response.choices[0].message.content}

Answer C:
{claude_response.content[0].text}

Answer D:
{grok_response.choices[0].message.content}

Answer E:
{deepseek_response.choices[0].message.content}


"""
```

Appendix F – Scenario 4: Structural Characteristics of the Generated Instructions

Structural and Content Consistency

Across both runs for each model, the generated instructions showed high intra-model consistency. Each model reproduced nearly identical structural patterns and instruction styles between runs, with only minor variations in phrasing or level of detail.

The same three-stage procedural structure remained dominant:

1. physical preparation
2. system readiness checks
3. takeoff execution

This indicates that the generation process is highly deterministic at the structural level, even when responses are regenerated independently.

APPENDIX G Generator Prompt for Scenario 4 and 5

The upload the complete user manual into the USER_MANUAL_TEXT variable.

```
# Load manual text for all the models, because they can't understand pdf as picture
with open("DJI_Mini_4_Pro_User_Manual.txt", "r", encoding="utf-8") as f:
    USER_MANUAL_TEXT = f.read()

print(f"Manual loaded: {len(USER_MANUAL_TEXT):,} characters")
print(f"Words: {len(USER_MANUAL_TEXT.split()):,}")
```

The instruction is defined:

```
tell_the_instruction = """Tell me the exact steps how to take off a DJI_Mini_4_Pro
drone. The relevant user manual of teh DJI Mini 4 Pro drone will be attached to the
user propmt to increase your relevantexpert knowledge. STRICT LIMIT: Maximum 500
words total response. nowledge of DJI Mini 4 Pro drone operations."""
```

User Prompt of the *Generator Role*

The following user prompt was constructed for all the five models in their generator role. The following code is working with gemini:

```
gemini_response = gemini.models.generate_content(model="gemini-3-flash-preview",
        contents=[            {
                "role": "user",
                "parts": [
                    {"text": tell_the_instruction},
                    {"text": USER_MANUAL_TEXT}
                ]
            }])
```

Index