Quantitative Analysis of Hallucination Bias in LLM Counting Tasks and Suppression Effects via Structured Protocol (KIS)

Hiroyasu Hasegawa
Mirairzu Lab Kobo
Experiment conducted: April 24-25, 2026 | Submitted: April 26, 2026

Abstract

This study presents an exploratory quantitative analysis of hallucinations arising when large language models (LLMs) count items in large volumes of unstructured text data, and examines the suppression effects of the Knowledge Innovation System (KIS), a proprietary structured protocol.

Three models — GPT-5.3 Instant, Gemini 3 Flash, and Claude Sonnet 4.6 — were evaluated on a three-label (Yes / No / Pending) text dataset ranging from 200 to 2,000 items under four conditions: Baseline (no protocol), KIS Level 4 / Logic: Strict, Chain-of-Thought (CoT) prompting, and a KIS + CoT hybrid.

Results showed that Gemini overcounted the Pending category by +38 items at 1,000 entries under Baseline, exhibiting what we term harmonic hallucination, yet achieved 100% accuracy across all scales with KIS applied. Claude maintained perfect accuracy up to 2,000 items without any protocol. ChatGPT abandoned the task beyond 800 items under Baseline but recovered to 100% accuracy at 1,000 items under the KIS + CoT hybrid. Notably, applying CoT alone to ChatGPT induced distribution fabrication even at 200 items, demonstrating a counter-productive effect.

Based on these findings, we propose a three-type taxonomy of LLM hallucination: Confabulation Type (Gemini), Avoidance Type (ChatGPT), and Process-Opaque Type (Claude). We further demonstrate that KIS functions as an external scaffold — structurally separating the counting, verification, and reporting phases via its log: full output — thereby leveling inter-model performance gaps and providing the audit trails required in practical deployments.

Keywords: LLM, Hallucination, Counting Task, Prompt Engineering, KIS, Chain-of-Thought, Model Comparison, Audit Trail

1. Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in natural language processing, yet a serious concern remains: hallucination — the fluent and plausible generation of factually incorrect information (Huang et al., 2023). Prior research has classified hallucinations along two axes: intrinsic (contradicting the input context) versus extrinsic (contradicting the model's training knowledge), and factuality versus faithfulness (Huang et al., 2023). However, these frameworks have primarily targeted knowledge-retrieval tasks and have left largely unexplored the hallucination behaviors that arise in practical, iterative data-processing tasks.

This study focuses on counting tasks — the aggregation of category frequencies from large volumes of unstructured text data — a task that is deceptively simple yet ubiquitous in real-world operations, and in which LLMs exhibit distinctive hallucination patterns. While NumericBench (Zhao et al., 2025) has shown that LLMs are surprisingly poor at numerical processing in general, no quantitative study to the authors' knowledge has specifically targeted large-scale text-based counting tasks.

This study addresses three research questions. First, what hallucination patterns do LLMs exhibit in counting tasks, and how do these differ across models? Second, how do externally structured prompts such as Chain-of-Thought (CoT) and structured protocols such as KIS each affect counting accuracy? Third, why does process transparency hold equal or greater practical significance than output accuracy alone?

To answer these questions, we conducted a controlled experiment using 200 to 2,000-item three-label text datasets with human-verified ground truth, testing three models across four conditions. Results support a three-type hallucination taxonomy and demonstrate the efficacy of KIS, yielding a practical recommendation for the standardization of structured protocols in LLM-based counting workflows.

2. Related Work

2.1 Hallucination Research in LLMs

Huang et al. (2023) provided a comprehensive survey of LLM hallucinations and established the intrinsic/extrinsic classification framework. Anh-Hoang et al. (2025) further proposed two quantitative metrics — Prompt Sensitivity (PS) and Model Variability (MV) — to distinguish whether hallucinations are attributable to prompt design or model-inherent characteristics. The three-type taxonomy proposed in this study extends these prior frameworks to describe hallucination behaviors specific to counting tasks.

2.2 Prompt Engineering and Chain-of-Thought

Wei et al. (2022) introduced Chain-of-Thought Prompting, demonstrating that eliciting step-by-step reasoning substantially improves accuracy on arithmetic and logical tasks. However, Meincke & Mollick et al. (2025) argued in "The Decreasing Value of Chain of Thought in Prompting" that the effectiveness of CoT varies

significantly by model type and task, showing negligible benefit for reasoning models and, in some cases, a counter-productive effect. The counter-productive CoT finding observed with ChatGPT in this study aligns with this conclusion.

2.3 Numerical Processing Capabilities of LLMs

NumericBench (Zhao et al., 2025) demonstrated that LLMs perform surprisingly poorly on basic numerical tasks including counting, comparison, and arithmetic. EC-Bench (2025) reported that even the best-performing model achieved only 29.98% accuracy on multimodal counting tasks. Unlike these studies, this research adopts an experimental design focused specifically on large-scale counting of plain text data, offering evidence on hallucination behaviors and countermeasures not captured by existing benchmarks.

2.4 The KIS Protocol

The Knowledge Innovation System (KIS) is an AI-augmented thinking framework developed by Hasegawa & Kamogawa (2026a, 2026b), centering on question generation and optimization. It has been published as defensive publication papers on Zenodo (DOI:10.5281/zenodo.18730671 and DOI:10.5281/zenodo.18951932). KIS provides structured control via JSON or natural language directives, sequential log output (log: full), and level-based control (Level 0–5). This study uses Level 4 (Logic: Strict).

3. Methodology

3.1 Dataset

The experimental dataset consists of text items taking one of three values:Yes, No, or Pending. Items were randomly generated using Excel's CHOOSE(RANDBETWEEN(1,3),...) function, with ground truth verified by the authors prior to experimentation. Dataset sizes ranged across 200, 400, 600, 800, 1,000, and (for Claude only) 1,500 and 2,000 items. Evaluation metrics were Mean Absolute Error (MAE) and task completion rate.

3.2 Models and Versions

Model	Official Version	Release Date	Status at Experiment
ChatGPT	GPT-5.3 Instant	March 3, 2026	Standard model, all users
Gemini	Gemini 3 Flash (gemini-3-flash-preview)	December 17, 2025	Gemini app default
Claude	Claude Sonnet 4.6 (claude-sonnet-4-6)	2026	Model used in this study

** Experiment conducted: April 24-25, 2026*

3.3 Experimental Conditions

Four conditions were established:

(1) Baseline (no KIS): Direct counting via a minimal natural language prompt instructing the model to report Yes, No, and Pending counts from the provided data.
(2) KIS (Level 4 / Logic: Strict): Structured protocol with a JSON or natural language directive and mandatory sequential log output (log: full).
(3) CoT prompt: Natural language instruction to write out each item sequentially in groups of 200 before aggregating.
(4) KIS + CoT hybrid: Combination of conditions (2) and (3).

Note on ChatGPT experimental conditions:

When the JSON-structured KIS was applied to ChatGPT, counting errors emerged as early as 400 items. This behavior was interpreted as JSON structure acting as processing noise for this model. Accordingly, the protocol was switched to a natural language equivalent of KIS Level 4 (hereafter KIS-JP). All ChatGPT KIS condition data in this paper reflects the post-switch implementation. This adaptive process is itself recorded as a finding reflecting model-specific characteristics.

** Full prompt text for each condition is provided in Appendix C*

3.4 Evaluation Metrics

For each condition and item-count level, the signed difference (positive or negative) from ground truth and absolute error (MAE) were computed. When the model abandoned the counting task mid-process, the result was recorded as "task not completed" in the completion rate.

4. Results

4.1 Summary of Experimental Results

Table 1 presents the counting results for all models and conditions (Yes column shown as representative; full data in Appendix A). An asterisk (*) denotes task abandonment.

Table 1: Experimental Results Summary (Yes column, representative values)

Condition	200	400	600	800	1,000	1,500	2,000
Ground Truth (Yes)	78	136	189	262	331	525	664
Ground Truth (No)	69	138	208	252	309	487	652
Ground Truth (Pending)	53	126	203	286	360	488	684
Gemini Baseline (Yes)	83	134	202	263	311	—	—
Gemini KIS (Yes)	78	136	189	262	331	—	—
Gemini CoT only (Yes)	—	—	—	—	331	—	—
Claude Baseline (Yes)	78	136	189	262	331	525	664
Claude KIS (Yes)	78	136	189	262	331	525	664
ChatGPT Baseline (Yes)	78	136	203	—*	—*	—	—

ChatGPT KIS-JP (Yes)	78	136	202	273	339	—	—
ChatGPT KIS+CoT (Yes)	—	—	—	—	331	—	—

** Green rows: Ground Truth / Orange: Gemini Baseline (with error) / Blue: Claude / Red: ChatGPT Baseline / White: KIS and hybrid conditions*

4.2 Gemini 3 Flash: Harmonic Hallucination

Under the Baseline condition, counting errors appeared as early as 200 items (Yes: 83 vs. ground truth 78). At 1,000 items, the Pending category was overcounted by +38 items (Pending: 398 vs. ground truth 360) while the total count was preserved at exactly 1,000 — a pattern we term harmonic hallucination. Applying KIS (Level 4 / Logic: Strict) yielded MAE = 0 across all item-count levels from 200 to 1,000.

4.3 Claude Sonnet 4.6: High Accuracy with Process Opacity

Under both the Baseline and KIS conditions, Claude maintained perfect accuracy (MAE = 0) across all levels from 200 to 2,000 items. As Table 1 shows, KIS condition results are numerically identical to ground truth, indicating that KIS adds value not through accuracy improvement but through process visibility — the provision of an audit trail (see Section 5.3). Under the Baseline condition, the counting process remained a black box, and the collapse point (the item-count threshold at which accuracy degrades) remains unidentified.

4.4 GPT-5.3 Instant: Task Avoidance and Counter-Productive CoT

Under the Baseline condition, ChatGPT completed counting up to 600 items but abandoned the task beyond 800 items (returning a "cannot count" response). Under the KIS-JP condition, accuracy was high up to 600 items but showed meaningful error at 1,000 items (Yes: 339 vs. ground truth 331).

Under the CoT-only condition at 200 items, the total count was preserved at 200 while the distribution was fabricated (Yes: 94, No: 66, Pending: 40), despite the Baseline condition producing zero error at the same scale. This confirms that the CoT directive acted as noise, degrading accuracy — a finding consistent with Meincke & Mollick et al. (2025). The KIS-JP + CoT hybrid condition achieved 100% accuracy at 1,000 items (Yes: 331, No: 309, Pending: 360).

Note that the ChatGPT KIS condition involved a switch from JSON-based to natural language implementation (KIS-JP), making it not strictly identical to the KIS conditions applied to Gemini and Claude (see Section 3.3). Accordingly, the ChatGPT KIS log output differs in format from the other two models and is excluded from the log figure comparison (see Section 5.2).

5. Discussion

5.1 A Three-Type Taxonomy of LLM Hallucination

Based on the experimental results, we propose the following three-type taxonomy

of LLM hallucination in counting tasks:

[Confabulation Type / Gemini] The model generates a statistically plausible numerical distribution and fabricates values while preserving the total count. This corresponds to models with high Prompt Sensitivity (PS) in the framework of Anh-Hoang et al. (2025), for which KIS structural constraints are highly effective. Notably, Gemini also achieved correct results under the CoT-only condition at 1,000 items, suggesting high sensitivity to prompt structure in general. However, as CoT-only results at other item counts were not collected, generalization requires further validation.

[Avoidance Type / ChatGPT] The model abandons the task or over-summarizes once processing load exceeds a certain threshold. A combination of structural and external constraints is required to maintain performance.

[Process-Opaque Type / Claude] The model achieves high accuracy but its counting process is a black box, leaving the collapse point unpredictable. KIS application makes the audit trail available, addressing this practical limitation.

5.2 KIS Efficacy: The External Scaffold Hypothesis

We propose the External Scaffold Hypothesis to explain the mechanism by which KIS achieves its effects. The attention mechanism of LLMs is prone to losing count-state tracking over long inputs. KIS's log: full function compensates for this internal attention limitation by requiring the model to externalize intermediate counting results as output — functioning as external memory.

This scaffolding effect is clearly visible in the actual log outputs. Under KIS, Gemini produced an explicit four-step output — Data Ingestion, Quantitative Counting, Consistency Check, Structural Transformation (Figure 1) — with consistency verification functioning as a structurally independent phase. Claude, under KIS, produced a three-layer structure of input validation, checksum, and processing notes, and additionally provided a self-organized summary of cumulative patterns across multiple runs — an instance of metacognitive output (Figure 2).

This differs from Wei et al.'s (2022) CoT, which promotes sequential natural language reasoning. KIS enforces phase separation — counting, verification, reporting — through structural constraints, functioning as a practical audit trail. The two approaches reflect complementary design philosophies rather than competing alternatives.

The fact that ChatGPT required a switch from JSON to natural language (KIS-JP) is itself informative: it suggests that KIS's external scaffold function does not depend on any particular format (JSON vs. natural language) but resides in the conceptual phase separation of mode-setting, counting, and verification. This format adaptability is recorded as a characteristic of model-protocol interaction.

Figure 1: KIS log output — Gemini 3 Flash (1,000-item condition, excerpt)

```
Analytical Execution Log (Full Processing Log)

Step 1. Data Structure Ingestion
   Input source: 5-column survey response dataset
   Row count: 200 rows / Total data points: 1,000
   Decision: All 1,000 data points designated for counting under KIS Strict
mode

Step 2. Quantitative Counting
   Exhaustive token-level enumeration executed
   Pending: 360  /  Yes: 331  /  No: 309

Step 3. Consistency Check
   331 (Yes) + 309 (No) + 360 (Pending) = 1,000  checkmark

Step 4. Structural Transformation
   Distribution pattern extraction and ratio formalization complete
   Result: Pending (360) > Yes (331) > No (309)  Total: 1,000

KIS-Genesis v2.2 | Final large-scale analysis complete. [LOG_END]
```

Figure 2: KIS log output — Claude Sonnet 4.6 (1,000-item condition, excerpt)

```
checkmark Input Data Validation
   Specified count: 1,000  /  Actual total: 1,000  ->  Exact match
   Data structure: 5 columns (tab-delimited) x 200 rows = 1,000 items

chart Count Results (all 1,000 items)
   Yes      331 items   33.1%
   No       309 items   30.9%
   Pending  360 items   36.0%
   Other      0 items    0.0%
   Total  1,000 items  100.0%
   Checksum: 331 + 309 + 360 = 1,000  checkmark

[LOG] Processing Notes
   Data structure: 5-column tab-delimited x 200 rows
   Undefined labels: 0 (all values matched exactly)
   Deviation from specified count: none (exact match)

Cumulative Pattern Summary (auto-detected)
   Run 1: 1 col x 200 rows = 200 items    Run 2: 2 cols x 200 rows = 400 items
   Run 3: 3 cols x 200 rows = 600 items   Run 4: 4 cols x 200 rows = 800 items
   Run 5: 5 cols x 200 rows = 1,000 items  <-- Exact match with specified count
   * Was this an intentional stress test?
```

5.3 Practical Implications: Process Disclosure Over Mere Accuracy

As Claude's results illustrate, achieving a correct output alone is insufficient for practical reliability assurance. In contexts involving audit, compliance, or legal accountability, process transparency is a necessary condition. KIS's log: full output

satisfies this requirement, functioning as an audit trail for counting operations. It is further useful for detecting the approach of a collapse point — without process visibility, sudden accuracy degradation is unpredictable. This applies even to high-accuracy models such as Claude, whose collapse point (at 3,000+ items) remains uncharacterized.

5.4 Limitations

This study is exploratory, and the following limitations apply: (1) Each condition was tested n=1 times; statistical testing of reproducibility has not been performed. (2) Model implementations are proprietary, so architectural interpretations remain inferential. (3) The dataset is limited to three-label (Yes/No/Pending) data; generalization to more complex category systems is a topic for future work. (4) The ChatGPT KIS condition involved a switch from JSON to natural language and is not strictly equivalent to the KIS conditions applied to the other models. (5) Claude's collapse point (at item counts above 3,000) has not been examined.

6. Conclusion

This study quantitatively analyzed hallucination in LLM counting tasks and proposed a three-type taxonomy: Confabulation Type, Avoidance Type, and Process-Opaque Type. It further demonstrated, through actual log outputs (Figures 1 and 2), that KIS functions as an external scaffold complementing model-specific hallucination characteristics.

Key findings include: (1) CoT prompting alone degraded accuracy below Baseline for ChatGPT; (2) the KIS + CoT hybrid overcame this degradation while achieving high accuracy; (3) process visualization is practically indispensable even for high-accuracy models such as Claude; and (4) the optimal protocol format (JSON vs. natural language) is model-dependent.

Future work includes reproducibility validation through multiple trials, collapse-point characterization for Claude at 3,000+ items, generalization to more complex category systems, unification of experimental conditions across all three models, and application to KIS-SaaS development.

References

[1]	Wei et al. (2022)	Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903
[2]	Huang et al. (2023/2024)	A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. arXiv:2311.05232. ACM TOIS.
[3]	Anh-Hoang et al. (2025)	Survey and analysis of hallucinations in large language models: attribution to prompting strategies or model behavior. Frontiers in AI, 8:1622292.

[4]	Meincke, Mollick et al. (2025)	Prompting Science Report 2: The Decreasing Value of Chain of Thought in Prompting. Wharton School Research Paper. SSRN:5285532.
[5]	Zhao et al. / NumericBench (2025)	Exposing Numeracy Gaps: A Benchmark to Evaluate Fundamental Numerical Abilities in Large Language Models. arXiv:2502.11075.
[6]	Chandra et al. (2025)	Sycophantic Chatbots Cause Delusional Spiraling, Even in Ideal Bayesians. arXiv:2602.19141.
[7]	Hasegawa & Kamogawa (2026a)	KIS: A Question-Centric Protocol Architecture for Hierarchical AI Thought Control. Zenodo. DOI:10.5281/zenodo.18730671. Published: 2026-02-22.
[8]	Hasegawa & Kamogawa (2026b)	KIS-Genesis v4.2: A Question-Centric Protocol Architecture for Hierarchical AI Thought Control. Zenodo. DOI:10.5281/zenodo.18951932. Published: 2026-03-11.

** Verify DOIs, volume numbers, and page numbers before journal submission*

Appendix A: Full Experimental Data

Table A-1: Complete results — all models, conditions, and item counts (Yes column)

Condition	200	400	600	800	1,000	1,500	2,000
Ground Truth (Yes)	78	136	189	262	331	525	664
Ground Truth (No)	69	138	208	252	309	487	652
Ground Truth (Pending)	53	126	203	286	360	488	684
Gemini Baseline (Yes)	83	134	202	263	311	—	—
Gemini KIS (Yes)	78	136	189	262	331	—	—
Gemini CoT only (Yes)	—	—	—	—	331	—	—
Claude Baseline (Yes)	78	136	189	262	331	525	664
Claude KIS (Yes)	78	136	189	262	331	525	664
ChatGPT Baseline (Yes)	78	136	203	—*	—*	—	—
ChatGPT KIS-JP (Yes)	78	136	202	273	339	—	—
ChatGPT KIS+CoT (Yes)	—	—	—	—	331	—	—

** Add complete Yes/No/Pending data for all conditions; Gemini CoT-only complete data also to be appended*

Appendix B: KIS Protocol Specification

The experimental prompts for KIS Level 4 / Logic: Strict are shown below. Note that the core KIS algorithm is maintained as a trade secret and is not disclosed; this appendix provides only the information necessary for experimental replication.

B-1: JSON-based KIS (applied to Gemini 3 Flash and Claude Sonnet 4.6)

```
{
  "mode": "analyze",
  "sw": {
```

```
    "level": 4,
    "boost": true
  },
  "checks": {
    "source": "required",
    "logic": "strict",
    "sum_check": [insert item count]
  },
  "outputs": {
    "structure": "on",
    "log": "full"
  }
}

Read the following [count] survey responses and accurately count
the number of Yes, No, and Pending answers, then report the totals.

~ Paste data here ~
```

** The sum_check value was adjusted to match the item count for each experimental level (200/400/600/800/1,000)*

B-2: Natural Language KIS (applied to GPT-5.3 Instant / KIS-JP)

Because the JSON-structured KIS produced errors as early as 400 items with ChatGPT, the protocol was switched to the following natural language instruction. The opening directive "Analysis mode, Level 4" functioned as a KIS Level 4 mode-setting equivalent.

```
Analysis mode, Level 4

Read the following [count] survey responses and accurately count
the number of Yes, No, and Pending answers, then report the totals.

~ Paste data here ~
```

** Despite being substantially more concise than the JSON version, KIS-JP demonstrated a positive effect on counting accuracy for ChatGPT*

Appendix C: Experiment Prompts

The full prompts for the Baseline and CoT conditions are provided below.

C-1: Baseline Prompt (no KIS, common to all models)

```
Read the following [count] survey responses and accurately count
the number of Yes, No, and Pending answers, then report the totals.

~ Paste data here ~
```

** The only difference between this prompt and Appendix B-1 is the presence or absence of the JSON block. The natural language instruction is identical, isolating the effect of JSON structure on counting accuracy.*

C-2: CoT Prompt (common to all models)

```
Read the following [count] survey responses and accurately count
the number of Yes, No, and Pending answers, then report the totals.
Write out all responses in numbered groups of 200 before aggregating.
Do not skip any item. Accuracy is the top priority.

~ Paste data here ~
```

** The KIS + CoT hybrid condition combines the JSON block from Appendix B-1 with the natural language instruction from Appendix C-2.*