

Article

Hallucination Mitigation in Large Language Model-Based Tool Recommendation: A Cross-Provider Architectural Ablation Study Across Two Model Generations [†]

Lavdim Menxhiqi *  and Galia Marinova * 

Faculty of Telecommunications, Technical University of Sofia, 8 Kliment Ohridski Blvd., 1000 Sofia, Bulgaria

* Correspondence: lavdim.menxhiqi@ubt-uni.net (L.M.); gim@tu-sofia.bg (G.M.)

[†] This article is a revised and expanded version of a paper entitled “Database-Grounded LLM Recommendations for Engineering Tool Selection: An Anti-Hallucination Architecture”, which was presented at the 1st International Conference on Industrial & Systems Engineering (ICISE-2026), Tirana, Albania, 16 April 2026.

Abstract

In a closed-inventory large language model (LLM) system such as Online-CADCOM, which recommends engineering tools from a verified inventory, we measure inventory non-compliance, that is, a mention-level event in which the model recommends a tool not present in the verified inventory. We use this inventory-relative sense of hallucination throughout: an out-of-inventory mention may be a fabricated tool or a real commercial tool absent from the curated inventory, so the metric reports inventory non-compliance rather than factual fabrication. We evaluate a three-mechanism mitigation stack consisting of database-grounded context injection, fixed vocabulary constraints, and enforced JavaScript Object Notation (JSON) output across three commercial LLM providers (OpenAI, Anthropic, Google), two model generations, and two output modes (standard and reasoning), totaling 6912 Application Programming Interface (API) calls over 12 configurations. Under a recall-equalized detector adopted as the primary metric, the inventory non-compliance rate, which we denote the hallucination rate (HR) following common usage, decreases from roughly 69–80% to 4–13% under the full architecture. The cross-provider average is similar across the two generations tested (8.5% Generation 1 (Gen1), 6.9% Generation 2 (Gen2)), although per-provider directions diverge. We also examine the C3 configuration, in which only JSON output enforcement is active without grounding. A naive detector reports a large hallucination increase over the unconstrained baseline (+10.1 percentage points (pp) Gen1, +15.1 pp Gen2), but we show this gap is largely a detection-format artifact: structured JSON fields make out-of-inventory tools easy to extract, whereas the same real tools are frequently missed in free text. Under a recall-equalized detector the gap narrows to +2.6 pp (Gen1) and +4.8 pp (Gen2) and remains statistically significant only for two current-generation models, indicating a small, current-generation effect rather than a universal one. Reasoning-mode models provide no statistically significant improvement under architectural constraints. A frequency-weighted audit shows that the majority of remaining out-of-inventory mentions correspond to real engineering tools absent from the platform’s inventory. Under the full architecture, roughly half of responses (pooled $P_{\text{any}} \approx 49.5\%$) still contain at least one such mention, indicating that handling unseen tools remains an open challenge for closed-inventory recommendation systems. Our evidence comes from a single engineering platform with four related electronic-design and power-electronics domains, so the findings characterize this setting rather than recommendation domains in general.



Academic Editor: Liviu-Adrian Cotfas

Received: 13 May 2026

Revised: 12 July 2026

Accepted: 20 July 2026

Published: 22 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](#)[Attribution \(CC BY\) license](#).

Keywords: large language models; hallucination mitigation; cross-provider evaluation; retrieval-augmented generation; electronic design automation tools; printed circuit board design tools

1. Introduction

Over the last few years, it has become increasingly common to deploy large language models (LLMs) in production systems. Enterprise spending on generative Artificial Intelligence (AI) reached \$13.8 billion in 2024 [1]. Recommendation and decision-support systems are among the areas where LLMs can have significant impact, for example by providing relevant product recommendations within a given domain based on their broad knowledge. This capability can complement engineering decision-support workflows that rely on validated and verified tool inventories. One of the key issues affecting the reliability of current LLM deployments is hallucination, in the general sense the generation of highly plausible but factually incorrect information [2,3]. In the closed-inventory setting studied here, we use a narrower, inventory-relative sense throughout: a recommendation is non-compliant when the named tool is not present in the verified inventory, regardless of whether that tool exists elsewhere in the world.

In the context of domain-specific recommendations, we encounter a specific form of hallucination. Here, a tool is recommended by the system but is not present in the verified inventory of the target platform. We study this phenomenon in the context of the Online-CADCOM system introduced in our previous work [4]. Online-CADCOM is a web-based system designed to assist engineers in selecting appropriate tools for their needs. In Menxhiqi and Marinova [4], we provided a detailed description of the prototype implementation, the PostgreSQL database design, and the initial validation trial conducted with university students. The system maintains a curated database of 82 verified engineering tools across multiple domains. An out-of-inventory recommendation may therefore correspond either to a fabricated tool name or to a real commercial tool that is not included in the platform's curated inventory. The broader problem of tool hallucination, where models select inappropriate tools or misuse them, has recently received dedicated attention [5]. Inventory-relative hallucination in domain-specific recommendations differs from general-purpose hallucination. In the latter case, hallucination may involve subtle factual inaccuracies that require human judgment to detect. In contrast, inventory-relative hallucination can be evaluated automatically via database lookup, achieving high precision and high (though not perfect) recall. Determining whether a recommended tool exists in the verified database is therefore an objective and largely automatable process.

Various techniques have been proposed to mitigate hallucination in LLMs. Retrieval-augmented generation (RAG) grounds outputs in retrieved factual content [6]. Chain-of-thought prompting decomposes reasoning into intermediate steps, which may reduce errors [7]. Grammar-constrained decoding restricts outputs to valid syntactic structures [8]. However, most empirical evaluations of these techniques focus on a single model family, leaving open the question of whether mitigation strategies generalize across providers.

This question is increasingly important as enterprises adopt multi-provider LLM strategies. A mitigation architecture that performs well on one provider may not exhibit the same behavior on another. In addition, LLM capabilities continue to evolve, including the introduction of “reasoning” or “thinking” modes that involve extended internal deliberation before producing output. Preliminary evidence (preprint; not yet peer-reviewed) suggests that reasoning models may exhibit higher hallucination rates in certain contexts [9], challenging the assumption that additional computation necessarily improves accuracy.

In this work, we empirically investigate the interaction of three constraint-based mechanisms in commercial LLMs: database-grounded context injection, closed-vocabulary constraints, and structured JavaScript Object Notation (JSON) enforcement. While each mechanism has been studied individually in prior work, to the best of our knowledge this is among the first systematic empirical characterizations of their combined behavior across multiple commercial providers and model generations. An alternative approach is post-processing via a “generate freely then filter” strategy, where tool names are validated against the database after generation. Although valid, this approach requires full response generation prior to filtering and may necessitate regeneration, whereas prompt-level constraints prevent out-of-inventory content from being produced in the first place.

The present paper extends our prior conference work [10], which evaluated the three-mechanism architecture on a single LLM provider (OpenAI GPT-4o-mini, 576 queries). The novelty delta over that conference paper is the cross-provider, cross-generational evidence: where the conference paper established that the architecture works for one model, this paper tests whether that effectiveness, the ablation ordering, and the C3 detection-format effect hold across three providers and two model generations, and adds a recall-equalized detector and a temperature control that were not part of the conference study. Specifically, we evaluate models from three major commercial providers (OpenAI, Anthropic, and Google) across two generations, totaling 6912 controlled Application Programming Interface (API) calls.

Multiple works investigate the concept of hallucination in natural language generation [2,3,11]. Ji et al. [2] classify hallucinations as intrinsic vs. extrinsic; intrinsic hallucinations are those that contradict the original source material while extrinsic hallucinations add information that is unverifiable. Out-of-inventory recommendations in a domain-specific setting would be classified as extrinsic hallucination since a structured database query can confirm their accuracy.

Reasons for hallucination in LLMs have been listed by Huang et al. [3], such as memorization of the training data, exposure bias, and the decoding algorithms. The fact that hallucination rates vary significantly across model architectures and scales suggests that cross-provider evaluation is needed, and that, as Zhang et al. [11] show, hallucination is a general characteristic of autoregressive language models not specific to any one architecture, occurring across a wide range of tasks.

Hallucination in conversational systems can be reduced substantially as Shuster et al. [12] show, by grounding the generation process in external knowledge. This principle can be extended beyond document retrieval to structured tool databases that encode hierarchical entities, attributes, and relationships.

Retrieval-augmented generation (RAG) [6] alleviates hallucination by incorporating retrieved documents into the generation process. By grounding outputs in verifiable information, RAG reduces unsupported claims. Gao et al. [13] survey RAG architectures and compare their effectiveness across hallucination types, revealing that tightly coupled retrieval-generation pipelines are particularly effective in domain-specific applications.

Peng et al. [14] extend retrieval-based methods through automated fact-checking feedback loops. Instead of fetching text snippets from databases, this retrieval method inserts structured entity data covering attributes, features, and associations of engineering tools, realizing a form of retrieval grounding tailored to relational engineering databases.

Geng et al. [8] show that grammar-constrained decoding can enforce valid output structures without requiring additional fine-tuning. By restricting the output space, this approach also helps reduce hallucination. In our architecture, we apply closed-vocabulary constraints (M2) at the prompt level by providing the model with a list of valid tool names. Structured JSON enforcement (M3) is implemented at the application layer.

Ayala and Bechard [15] examine how structured output specifications interact with hallucination in retrieval-augmented systems. Their results show that structured output formats can either improve or reduce factual accuracy depending on whether grounding context is available. Their study focuses on a single-model question-answering setting, suggesting that the phenomenon reflects a broader interaction between structured output and grounding rather than being limited to a model-specific artifact.

The advancement in the design of LLMs that are capable of reasoning represents a major step forward in model design. This is illustrated by the way scaling model capabilities, as demonstrated by GPT-4 and subsequent models, improves performance across diverse tasks [16]. Additionally, instruction-following training with human feedback (i.e., Reinforcement Learning from Human Feedback, RLHF), as shown by Ouyang et al. [17], improves alignment and serves as a basis for modern reasoning models.

It has been conjectured that greater reasoning ability should result in improved accuracy. However, recent evidence suggests that reasoning-mode models may achieve similar or even higher hallucination rates in certain settings, possibly due to over-elaboration during extended deliberation that generates plausible but unfounded reasoning chains.

The use of large language models (LLMs) in engineering fields such as Electronic Design Automation (EDA) has recently been a prominent topic within EDA research. Wu et al. [18] introduce ChatEDA, an LLM-based autonomous agent developed for EDA tasks, and show how domain-adapted language models can support circuit design workflows. Liu et al. [19] created ChipNeMo, a domain-adapted LLM for chip design (preprint; not yet peer-reviewed); this system uses retrieval-augmented generation and proprietary design data to achieve significant advances compared to general-purpose models. The studies in these references demonstrate how LLMs are increasingly being integrated into engineering disciplines, and emphasize the importance of mitigating hallucination when generated outputs refer to actual tools, specifications or technical entities. However, these domain-adaptation approaches require significant training investment and are tied to specific model families, leaving open the question of whether provider-portable architectural constraints can achieve comparable hallucination reduction without fine-tuning.

A complementary line of work mitigates hallucination by reasoning over explicit relational structure rather than by constraining the prompt. Luo et al. [20] ground LLM reasoning in knowledge-graph paths, retrieving relation chains that the model must follow to produce faithful and interpretable answers. Graph-structured representations have also been used outside text generation to detect inconsistencies automatically; for example, Cimino and Deufemia [21] build interaction graphs over IoT automation rules to flag conflicting behaviors without supervision. These approaches enforce consistency through a learned or constructed graph that the model traverses at inference time. Our work differs in the locus and cost of the constraint: rather than guiding reasoning over a graph, we impose vocabulary and schema constraints at the prompt level against a flat verified inventory, which requires no graph construction, no retrieval-path training, and no per-provider model adaptation, and is therefore directly portable across commercial APIs.

Online-CADCOM is a research web platform for engineering tool selection that has evolved through several research phases, from earlier Multi-Criteria Decision Analysis (MCDA)-based automatic tool selection [22] to the current LLM-powered tool recommendation. Its Dynamic Expert Module (DEM) architecture is implemented using React JS on the frontend, ASP.NET Core 8.0 on the backend, and PostgreSQL as the database [4].

In this study, Online-CADCOM functions as a closed-world recommendation system. It maintains a curated inventory of 82 verified engineering tools with objectively verifiable membership; thus, inventory non-compliance can be measured as a binary classification problem: a recommended tool either exists in the inventory or it does not. The four evaluation domains

all lie within electronic-design and power-electronics tooling, so results should be read as characterizing this related family of domains rather than recommendation tasks at large.

Bommasani et al. [23] and Zheng et al. [24] advanced systematic LLM assessment through holistic benchmarking and combined automated-human evaluation, and recent frameworks evaluate multi-step tool and function calling under controlled, contamination-free conditions [25]. However, these assessments focus on general performance and do not investigate whether mitigation architectures transfer across provider ecosystems. A systematic cross-provider evaluation of hallucination-mitigation architectures spanning multiple providers and model generations remains an open gap in the literature. In this work, we aim to address this gap by evaluating a three-mechanism constraint-based architecture across three commercial providers and two model generations.

The rest of the paper is organized as follows: Section 2 presents the methodology, Section 3 reports the results, Section 4 discusses the findings, and Section 5 concludes the paper.

2. Methodology

2.1. System Architecture

Online-CADCOM is a research application that is fully implemented and deployed as an online system for managing and selecting engineering tools. It is implemented using ASP.NET Core 8.0 and React 18 [4]. The application uses a PostgreSQL database containing 82 verified engineering tools, developed through prior work on the platform's knowledge base and tool inventory [26]. The tools are structured into a hierarchy of categories and subcategories, and each tool record includes characteristics, parameters, and inter-tool relationships. The names of the 82 verified tools used as the verification inventory are provided in the Supplementary Materials.

Figure 1 shows the anti-hallucination architecture with three sub-modules that can be switched on and off.

Mechanism M1: Database-Grounded Context Injection. We retrieve PostgreSQL records for tools and use a hierarchical context builder with six stages that generates the following contexts in sequence: (1) attributes of the target tool including feature attributes, input and output connections, and verification tools; (2) sibling tools in the same subcategory; (3) tools in the parent category; (4) tools in adjacent subcategories under the same parent category; (5) cross-category tools that share feature overlap with the target tool; and (6) a global list of all tool names. The number of tokens in the grounding context depends on domain density and is typically between 2000 and 7000. Because M1 includes tool records that contain tool names alongside their attributes, tool names are implicitly disclosed in the context. Retrieval-based grounding of this kind has been shown to reduce hallucination in large language models [27]. The M2 constraint can be seen as a refinement on top of this, which we discuss further in Section 2.3.

Mechanism M2: Closed-Vocabulary Constraint. We enforce a constraint using the following instruction placed before a list of tool names: "You MUST ONLY recommend tools from the following verified database. Do NOT invent, assume, or hallucinate any tool names." This constraint is reinforced by an enumerated list of tool names (names only, without additional attributes) repeated at multiple points in the prompt. In contrast to M1, the names are shown without contextual details. With 82 tool names, the complete vocabulary fits within the prompt context. For substantially larger inventories, a retrieval step would be required to select a top-K candidate subset from the full vocabulary. Unlike grammar-constrained decoding [8], which operates on the decoder, M2 is a prompt-level constraint, the deployable approximation available for commercial APIs that do not expose decoder access.

Mechanism M3: Structured JSON Enforcement. The model is prompted to output its response in a fixed JSON format, populating fields such as `toolName`, `toolId`, `features`, `strengths`, `limitations`, and `workflowSteps`. The reliability with which models adhere to such JSON schemas has itself been systematically benchmarked [28]. M3 forces structured output but provides no grounding; in the C3 configuration it is applied without M1 or M2. Whether ungrounded JSON enforcement increases out-of-inventory recommendations, and how much of any apparent increase is a detection-format artifact, is examined in Sections 2.10 and 4.2.2.

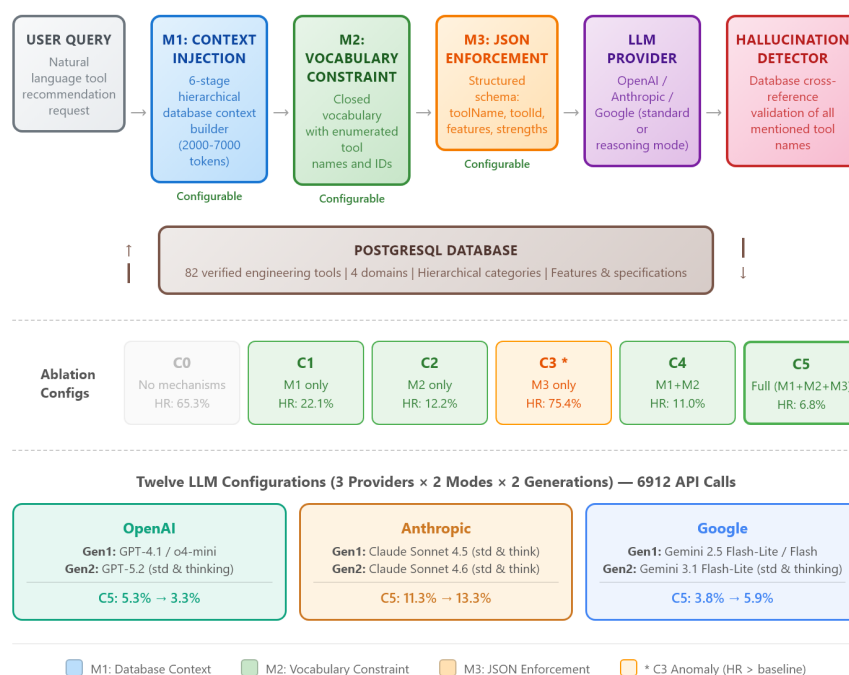


Figure 1. Three-mechanism anti-hallucination architecture. User queries pass through configurable combinations of M1, M2, and M3 before reaching any of twelve large language model (LLM) configurations (three providers, two generations). Responses are validated against the tool database. The architecture is evaluated unchanged across 6912 Application Programming Interface (API) calls. Horizontal arrows indicate the direction of data flow through the pipeline; the vertical arrows next to the database denote retrieval from and validation against the tool inventory. Standard models are run on all six configurations (C0 to C5); thinking variants use a partial factorial design and are evaluated only at C0 and C5, plus a C3 check for Generation 2 (see Section 4.3).

2.2. Ablation Design

We define six configurations representing all meaningful combinations of the three mechanisms (Table 1):

Table 1. Ablation configuration matrix. Each configuration represents a unique combination of mechanisms M1 (Context), M2 (Vocabulary), and M3 (JSON).

Config	Description	M1	M2	M3
C0	Ungrounded baseline	—	—	—
C1	Context only	✓	—	—
C2	Vocabulary only	—	✓	—
C3	JSON only	—	—	✓
C4	Context + Vocabulary	✓	✓	—
C5	Full architecture	✓	✓	✓

✓ = mechanism active; — = mechanism not active.

The three binary mechanisms yield $2^3 = 8$ possible combinations. We evaluate six configurations and exclude M1 + M3 (context + JSON, no vocabulary) and M2 + M3 (vocabulary + JSON, no context). These two configurations are excluded for three main reasons. First, our main question for M3 is whether it adds benefit on top of the minimum effective configuration M1 + M2, which is tested directly by comparing C4 and C5. Second, the weak performance of C3 shows that JSON enforcement without grounding is counterproductive, and the partially grounded intermediate cases are of lower mechanistic interest for this paper. Third, adding the two excluded configurations would require 1152 additional API calls per generation (two configurations $\times$ three providers $\times$ four domains $\times$ 12 prompts $\times$ three repetitions $\times$ two generations), which increases experimental cost by 33% for limited additional insight. We therefore postpone M1 + M3 and M2 + M3 to future work, where they can more directly test whether partial grounding affects out-of-inventory generation under JSON enforcement. The six evaluated configurations allow us to isolate each mechanism (C1, C2, C3), test the minimum effective combination (C4 = M1 + M2), and evaluate the full architecture (C5 = M1 + M2 + M3).

2.3. Ablation Overlap Considerations

A methodological note about mechanism isolation is necessary because M1's tool records include tool names. As a result, C1 (M1 only) provides implicit tool-name exposure through the injected context, while C2 (M2 only) provides explicit tool-name exposure combined with an explicit behavioral constraint, but without contextual details. The ablation therefore reflects practical deployment configurations rather than perfectly isolated mechanisms. All mechanism-attribution language should be interpreted at the configuration level. For example, "C1 is associated with X% reduction" refers to the effect of the C1 configuration (M1 enabled, M2/M3 disabled), not to M1 in isolation. Empirically, C2 outperforms C1 in both generations (Gen1: 12.2% vs. 22.1%; Gen2: 8.2% vs. 19.1%), suggesting that explicit behavioral constraints in M2 are associated with lower hallucination than implicit name exposure in M1, despite M1 providing richer context. A cleaner ablation (M1 with and without the tool-name list) could isolate this effect further in future work. Full prompt templates for all six configurations are provided in the Supplementary Materials.

2.4. Model Selection

We evaluate twelve LLM configurations from three providers in two operational modes and across two model generations (Tables 2 and 3).

In our prior conference paper, we demonstrated the effect of this architecture on a single provider model (OpenAI GPT-4o-mini, 576 queries). Here we examine robustness across providers. Specifically, we evaluate the three major commercial LLM API providers, OpenAI, Anthropic, and Google, which represent the dominant practical options for deploying LLM-based recommendation systems. Switching providers may affect results due to differences in model families, alignment through RLHF, and tokenization.

Two-generation evaluation. A single cross-provider evaluation provides only a snapshot in time. Because LLMs evolve rapidly, we repeat the full experimental protocol on each provider's next-generation models as of February 2026 (Generation 2), in addition to the models available as of January 2026 (Generation 1). With three providers, two modes (standard and thinking), and two generations, this yields a $3 \times 2 \times 2$ factorial design. This design allows us to assess whether the mitigation architecture remains effective as models evolve and provides practitioners with evidence that deployment investments are not invalidated by a model upgrade.

Reproducibility parameters. The specific API parameters applied in each configuration are provided in Supplementary Table S1. All experiments were carried out in Sofia, Bulgaria, between 25 January and 15 February 2026.

Supplementary Materials. All experimental data and source code are publicly available. The repository includes: (1) the 82 tool names that are part of the verification inventory; (2) the 197-name H1/H2 classification CSV with evidence URLs; (3) the evaluation framework source code (prompt builder, hallucination detector, metrics calculator); (4) the prompt templates for all configurations (C0–C5); (5) analysis scripts for all reported statistics and figures; and (6) the database schema of the tool framework plus a sample subset of tool records for reproduction using synthetic data. The repository README includes reproduction instructions.

Table 2. Generation 1 LLM configurations. These models represent the initial cross-provider evaluation baseline.

Provider	Standard Model	Reasoning Model	Reasoning Mechanism
OpenAI	GPT-4.1	o4-mini	Separate model family
Anthropic	Claude Sonnet 4.5	Claude Sonnet 4.5	Extended thinking toggle
Google	Gemini 2.5 Flash-Lite	Gemini 2.5 Flash	Model variant

Table 3. Generation 2 LLM configurations. These are each provider’s next-generation models (February 2026). All three providers now use a single model with a configurable thinking toggle, eliminating the cross-provider reasoning heterogeneity present in Generation 1.

Provider	Standard	Reasoning (Thinking)	Mechanism	Parameter
OpenAI	GPT-5.2	GPT-5.2	Reasoning effort	reasoning.effort
Anthropic	Claude Sonnet 4.6	Claude Sonnet 4.6	Adaptive thinking	thinking.type: adaptive
Google	Gemini 3.1 Flash-Lite	Gemini 3.1 Flash-Lite	Thinking level	thinkingLevel

In Generation 1 there is considerable variability in the implementation of the reasoners across the different providers (i.e., different model families, different API parameters, different models). This makes it very difficult to make any meaningful comparison of reasoners across the different providers. Generation 2 resolves this: all three providers converge on a single-model paradigm with configurable thinking toggles (see Supplementary Table S1 for parameters).

2.5. Experimental Design

We use a factorial design experiment utilizing the Provider dimension (OpenAI, Anthropic, and Google), Mode dimension (standard and thinking), and Generation dimension (Generation 1 (Gen1), Generation 2 (Gen2)) conducted on 3 × 2 × 2 combinations of providers, modes, and generations. By “generation” we refer to each provider’s cohort of models evaluated at a given time under the same protocol (Gen1: January 2026; Gen2: February 2026). Models were identified through APIs provided by the three providers during the measurement period and are listed in Tables 2 and 3.

Ablation configuration (C0–C5) and domain (D1–D4) are additional within-subject factors. We use a partial factorial design for cost–information trade-off: standard models are evaluated on all six configurations, while thinking models are evaluated only on boundary configurations C0 and C5. A full ablation for thinking models would require 5184 additional API calls (3 providers × 6 configs × 4 domains × 12 prompts × 3 reps × 2 generations). The boundary-only design suffices, as the key question is if thinking increases performance

at both (C0) the ungrounded baseline and (C5) the full architecture; in this case, all of the observable $\Delta C5$ scores were less than 2%.

Phase 1: Gen1 Full Ablation (Standard Models). $3 \times 6 \times 4 \times 12 \times 3 = 2592$ API calls.

Phase 2: Gen1 Thinking Comparison. $3 \times 2 \times 4 \times 12 \times 3 = 864$ API calls.

Phase 3a: Gen2 Full Ablation (Standard Models). $3 \times 6 \times 4 \times 12 \times 3 = 2592$ API calls.

Phase 3b: Gen2 Thinking Comparison. $3 \times 2 \times 4 \times 12 \times 3 = 864$ API calls.

In total, the experiment consists of 6912 API calls across twelve model configurations spanning two generations.

Accounting of N. Here, N denotes the total number of API calls (queries) in each reported analysis. A single query is one API call to one model with one prompt under one configuration (Supplementary Table S2). All 6912 calls returned valid responses; no calls were excluded due to timeouts, malformed responses, or API errors.

Each cell has $12 \text{ prompts} \times 3 \text{ repetitions}$, yielding 36 queries. Within-prompt standard deviation was below 2% for C5 configurations. In the Results tables, N (the number of API calls included in each specific analysis) is reported alongside each result.

2.6. Evaluation Domains

These four domains represent variations in tool density and level of specialization (Table 4). Here, “tools” refers to engineering software products such as EDA applications, calculators, and simulators, not UI functions. In Domain D1 (Printed Circuit Board (PCB) design tools), some names are proper product names (e.g., “KiCad”, “Eagle”), while others are more generic. More generic names are handled through structured extraction and multi-word matching within the hallucination detection pipeline.

Table 4. Evaluation domains with tool counts and category hierarchy.

ID	Domain	Tools	Related	Examples
D1	PCB Design Tool	8	21	KiCad, EasyEDA, EAGLE
D2	PCB Design Calculator	15	20	Saturn PCB Toolkit, Technick Impedance Calc.
D3	Switched-Mode Power Supply (SMPS)/Converters	8	8	PowerEsim, WEBENCH (TI), ADIsimPower
D4	Transformers	10	19	goodcalculators, jcalc, alfatransformer

The four evaluation domains include 41 primary tools; the remaining 41 tools in the 82-tool inventory belong to non-evaluated categories. The “Related” column indicates tools from sibling and parent categories included via the M1 hierarchical context builder. M2 always includes all 82 tool names regardless of domain, and hallucination detection operates against the full 82-tool inventory.

2.7. Prompt Generation

For each domain, 12 test prompts are generated to cover diverse query types, including workflow recommendations, tool comparisons, simulation requests, verification needs, manufacturing workflows, calculator recommendations, alternatives, student-oriented guidance, advanced analysis tools, and multi-domain integration scenarios. The resulting 48 prompts (12 per domain across D1 to D4) are author-constructed to span these query types rather than sampled from production interaction logs, so their representativeness of the real Online-CADCOM user query distribution has not been validated; this is noted as a threat in Section 4.3. As an automated, prompt-independent check on relevance, we observe that under C5 the recommended tools predominantly fall within the queried domain category (Section 4.1.2); a full domain-relevance proxy against the platform’s MCDA rankings is left to future work.

2.8. Metrics

We compute four metrics for each configuration-model-domain combination.

Hallucination Rate (HR). HR is the proportion of tool-name mentions that are absent from the verified database:

$$HR = \frac{|\text{Hallucinated Tools}|}{|\text{Total Mentioned Tools}|} \quad (1)$$

The counting unit is the distinct tool-name mention. Tool names are deduplicated within each response (case-insensitive). Each provider-configuration cell rate is mention-pooled: hallucinated and total mentions are summed over the 144 queries in the cell before dividing, so queries that list more tools contribute proportionally more weight. Throughout the paper, rows and columns labeled “Avg” (cross-provider or cross-domain averages) are unweighted arithmetic means of the underlying cell rates, with each cell counting equally regardless of its mention volume. Pooled cell rates and unweighted means of those cell rates can therefore differ; we use pooled rates for individual cells and unweighted means for the summary “Avg” entries.

Grounding Rate. The complement of hallucination rate ($1 - HR$).

P_{any} (Query-Level Hallucination Probability). The proportion of queries containing at least one out-of-inventory tool mention:

$$P_{\text{any}} = \frac{|\{q : |\text{Hallucinated Tools}(q)| \geq 1\}|}{|\text{Total Queries}|} \quad (2)$$

This metric captures what users experience directly: whether a response contains any out-of-inventory recommendation. Under C5, P_{any} in Gen1 standard models ranges from 35.4% (GPT-4.1) to 59.0% (Claude Sonnet 4.5), and in Gen2 standard models from 32.6% (Gemini 3.1 Flash-Lite) to 91.7% (Claude Sonnet 4.6). These values can be higher than HR because a single hallucinated tool among many grounded ones yields a low mention-level rate but still produces a “contaminated” response from a user perspective. We also define $P_{\text{any,H2}}$, the probability of encountering at least one H2 (external-real) tool recommendation:

$$P_{\text{any,H2}} = \frac{|\{q : |\text{H2 Tools}(q)| \geq 1\}|}{|\text{Total Queries}|} \quad (3)$$

An operationally significant failure mode can be isolated: how frequently does a user of the platform encounter real, competing tools (that have not been vetted by the platform)? $P_{\text{any,H2}}$ is reported under Section 4.2 in addition to P_{any} .

H1/H2 Taxonomy. Two types of out-of-inventory mentions are identified: H1 (fabricated) refers to names of tools that do not have an external referent (e.g., PCB Pro Designer 3000) and H2 (external-real) refers to tools that do exist commercially or as open-source software but are not listed in the curated inventory (e.g., Altium Designer, LTspice). These represent failures relative to the closed inventory, but they have different epistemic status. Frequency-weighted decomposition of the most frequent out-of-inventory names by manual verification is presented under Section 4.2.

Tools Per Query (T/Q). The average number of distinct tool names mentioned per response. This is essential for interpreting HR.

Expected Hallucinated Tools Per Query ($E[H/Q]$):

$$E[H/Q] = \frac{\sum_q |\text{Hallucinated Tools}(q)|}{|\text{Total Queries}|} \quad (4)$$

Unlike HR , $E[H/Q]$ is not affected by “denominator dilution” (listing more grounded tools). It captures the absolute user impact per response.

2.9. Hallucination Detection

Hallucination detection can be decomposed into two sub-tasks: (1) the tool mention detection task, which aims to locate the candidate tool mentions in the response, and (2) the tool mention grounding task, which determines whether each mentioned tool is grounded or hallucinated.

2.9.1. Candidate Tool Name Extraction

Candidate tool names are extracted using three complementary strategies applied to every response:

Strategy 1: JSON field extraction. A considerable number of results can be parsed as valid JSON (mainly C3 and C5). We distinguish two notions used throughout the paper: a response is *API-valid* if the provider returned it without error, whereas it is *JSON-schema-compliant* if its body parses as JSON and exposes the expected tool fields. The JSON% column in the response-characteristics tables (Sections 3.5 and 3.6.5) reports JSON-schema compliance, not API validity; all 6912 calls were API-valid. Tool names are extracted from specific schema fields (`toolName`, `tool`) via recursive JSON traversal, ignoring the meta-description fields (`phase`, `step`, `action`, `verdict`). We filter the string values by length, which must be between 3 and 100 characters.

Strategy 2: Known-name matching. The 82 known tool names are loaded when the application starts. For each full name, the tool searches the raw response for the full tool-name string using case-insensitive contain search (`response.Contains(fullToolName, OrdinalIgnoreCase)`). The names are matched as full phrases which are composed of multiple words. Known-name matching only matches grounded mentions and cannot catch hallucinated names.

Strategy 3: Software product pattern detection. Regex patterns derived from engineering software naming conventions are applied to detect hallucinated tool names not present in the database. Patterns include common software suffixes (“Designer”, “Pro”, “Suite”, “Studio”, “Toolkit”, “Calculator”, “Simulator”, “Viewer”, “Editor”), version number patterns (e.g., “OrCAD 17.4”), and a curated list of known commercial EDA tools outside our inventory (e.g., Altium Designer, OrCAD, Cadence, Mentor Graphics, Proteus, Multisim, LTspice). These patterns aim to match the common naming conventions of tool names while excluding common English words.

When we combine the three strategies, we get the set of candidate tool mentions for each response. For free-form text responses (C0, C1, C2, C4), Strategies 2 and 3 are the dominant strategies. For JSON responses (C3, C5), all three strategies are used, but we prefer Strategy 1 due to its higher precision. The precision of Strategy 1 is due to its use of the structural information of the JSON fields.

2.9.2. Grounding Classification

Each extracted candidate is classified as grounded or hallucinated via a three-level matching cascade:

- **Exact match** (case-insensitive): candidate exactly matches a database tool name.
- **Fuzzy containment**: if either string contains the other and the length ratio (shorter/longer) exceeds 0.55, classify as grounded (e.g., “KiCad EDA” matching “KiCad”).
- **Edit distance**: for names longer than 4 characters, if Levenshtein distance ≤ 2 , classify as grounded (handles minor typos such as “KiCaad”).

Candidates failing all three levels are classified as hallucinated.

2.9.3. Scope and Limitations of Detection

Ambiguous names. We found three ambiguous names in D1–D4: “EAGLE” (a common English word), “jcalc” (a short, generic-sounding name), and “Maddox” (a common surname). For Strategy 2, we match the full tool-name strings against the input strings in a case-insensitive containing fashion rather than in a case-insensitive substring fashion. Sensitivity analysis confirms that excluding these changes hallucination rates by <1 percentage point and that our findings are unchanged.

False negatives. Hallucinated tool names that do not fit the patterns described in Strategy 3 may be missed, making our hallucination estimates conservative for free-form configurations (C0, C1, C2, C4). In JSON configurations (C3, C5), Strategy 1 substantially reduces this risk by extracting tool names directly from schema fields.

False positives. A non-tool word matching a regex pattern could be incorrectly identified as a hallucinated tool. There were zero false positives from validating 50 stratified responses (25 C0, 25 C5, evenly balanced by provider) across 485 extracted tool mentions, with 28 undetected tool names (27 from C0 free-form responses and 1 from C5 JSON responses). Overall recall is estimated at approximately 94.5%, and above 99% for C5 due to JSON field extraction. Because recall is lower for free-form baselines, C0 hallucination rates are conservative lower bounds, and the C0-to-C5 reduction is likely equal to or greater than reported.

Detection pipeline asymmetry. Recall is higher for JSON-enforced configurations (C3, C5) than for free-text configurations due to Strategy 1. This asymmetry biases against our improvement claims, since it makes uncontrolled baseline rates (C0) conservative lower bounds while capturing nearly all hallucinated mentions under JSON enforcement. Any bias therefore works against our reported reduction, making the measured improvement conservative.

Finally, our precision measure is about whether an extracted string behaves like a tool mention (i.e., looks like a tool name entity). A post hoc analysis under Section 4.2 identifies a subset of out-of-inventory mentions that are generic or non-specific entities (e.g., “Practical Tips”, “Score”) that pass syntactic extraction but lack a specific product referent. These are still correctly classified as out-of-inventory, but they inflate headline *HR* relative to the operationally relevant H2 (external-real) category.

2.10. Detection Validation and Recall Equalization

We validated the production detector on a stratified manual audit. Precision is 100% (485 of 485 extracted mentions; 95% CI [99.2, 100]) and overall recall is 94.5% (485 of 513; 95% CI [92.2, 96.2]). Recall is above 99% for JSON-enforced configurations (C3, C5) because Strategy 1 extracts tool names directly from schema fields, but lower for free-text configurations (C0, C1, C2, C4), where Strategy 3 relies on a curated regex list. This asymmetry is benign for the C0-to-C5 reduction (it makes C0 a conservative lower bound), but it inflates the apparent C3-minus-C0 gap, because the same real out-of-inventory tools are caught under JSON yet missed in free text (about 96% missed in C0; 1322 of 1374 mentions for ANSYS, FEMM, COMSOL, MATLAB, Simulink, and PSIM).

To remove this asymmetry we constructed a recall-equalized detector that raises C0 recall to the level of C3 rather than degrading C3. It retains the production grounding and classification and adds, to every response, any high-precision out-of-inventory tool from a curated gazetteer that appears in the raw text but was missed by production, de-duplicated against already-flagged tools. A 40-sample stratified audit of the added matches found 100% precision (40 of 40 genuine out-of-inventory recommendations). With an empty gazetteer the detector reproduces the published production numbers exactly, so the two metrics are directly comparable. We report results under both detectors and adopt the recall-equalized detector as the primary metric for the C3 comparison (Sections 3.3 and 3.6.2).

2.11. Statistical Methods

All comparisons use non-parametric tests. We treat hallucination rate (*HR*) as non-parametric because (i) *HR* is bounded on $[0, 1]$, and (ii) under effective configurations the empirical *HR* distributions are non-normal and typically right-skewed.

- **Within-model comparisons** (C0 vs. C5, C0 vs. C3): Wilcoxon signed-rank test with query-level paired observations ($n = 144$ per provider per configuration). The C0 vs. C5 tests are two-sided. The C0 vs. C3 tests are also two-sided since we had no a priori belief about the direction of the anomaly.
- **Cross-mode comparisons** (standard vs. thinking) are evaluated using the Wilcoxon signed-rank test with query-level pairing by (domain, prompt, repetition), since both variants are evaluated on the same 144 queries for each provider ($n = 144$). This paired design, also used in the within-model ablation and in the multi-group ordering discussed below, ensures that identical queries are compared across conditions, with pairing by query identity controlling for prompt-difficulty effects. As a robustness check, we also report Mann–Whitney *U* tests on independent samples. When the two tests differ, the higher statistical power of the paired test reveals significant differences favoring the standard models (Gen1 Google, $p < 0.001$; Gen1 cross-provider aggregate, $p = 0.008$). Differences between standard and thinking modes are computed as $HR_{\text{thinking}} - HR_{\text{standard}}$; therefore positive $\Delta C5$ and positive r_{rb} indicate higher hallucination under thinking mode.
- **Cross-generational comparisons** (Gen1 vs. Gen2): Mann–Whitney *U* test on independent samples ($n = 144$ per group), because Gen1 and Gen2 models are architecturally distinct (e.g., GPT-4.1 vs. GPT-5.2), and we expect the assumption of comparable within-query error structure to be weaker than for within-generation mode comparisons.
- **Multi-group ordering** (C0–C5): Friedman test followed by three planned pairwise Wilcoxon signed-rank tests (paired, consistent with the repeated-measures design) to assess contrasts aligned with the study hypotheses: (i) C2 vs. C4 to test whether adding M1 to M2 improves performance; (ii) C2 vs. C5 to test whether the full architecture outperforms vocabulary alone; and (iii) C4 vs. C5 to test whether adding M3 to M1 + M2 improves performance.

For planned contrasts, we performed a Bonferroni correction for $k = 3$ tests. Other tests that show up in the results (e.g., the C3 effects, standard-vs.-thinking, and cross-generational comparisons) are not multiplicity corrected and should be treated as exploratory and merely indicative. The significance threshold for all tests is $\alpha = 0.05$. Tables report mention-aggregated *HR*, while all statistical tests operate on per-query *HR* values (one observation per query; $n = 144$ per cell). Zero-difference pairs (queries with identical *HR* under both conditions) are excluded from the Wilcoxon signed-rank test and from the computation of r_{rb} . For the primary C0 vs. C5 contrast, the proportion of zero-difference pairs is $\leq 10.4\%$ per cell. For ablation pairwise contrasts among low-*HR* configurations (e.g., C2 vs. C4), this proportion can reach 50–64% because both conditions frequently yield zero hallucinations for the same query.

Effect sizes. We report matched-pairs rank-biserial correlation (r_{rb}) as the primary paired effect size, Cliff’s delta (δ) as a descriptive non-parametric measure [29], and Cohen’s *d* for comparability. Cliff’s delta thresholds follow Cliff [29]: $|\delta| < 0.147$ (negligible), < 0.33 (small), < 0.474 (medium), and ≥ 0.474 (large). Formulas and relationships between measures are provided in the Supplementary Materials.

Confidence intervals. We report 95% bootstrap confidence intervals derived using the percentile method with 10,000 resamples, with the individual query as the resampling unit ($n = 144$ query resamples per provider-configuration cell). Each resample draws 144 query-level *HR* values with replacement, recomputes the mention-aggregated rate,

and records the 2.5th and 97.5th percentiles. When bootstrap and parametric intervals diverge, we report the bootstrap interval. Additional bootstrap details are provided in the Supplementary Materials.

3. Results

We report results in two parts. The first part presents the Generation 1 cross-provider baseline (Phases 1–2: GPT-4.1, Claude Sonnet 4.5, Gemini 2.5 Flash-Lite/Flash; 3456 API calls). The second part presents Generation 2 results (Phases 3a–3b: GPT-5.2, Claude Sonnet 4.6, Gemini 3.1 Flash-Lite; 3456 additional calls).

Throughout this section we use the hallucination rate (HR) as a short name for the inventory non-compliance rate defined above; where the bare term “hallucination” appears it refers to this inventory-relative quantity, not to factual fabrication.

Note: Cross-provider absolute rate comparisons should be interpreted carefully, since default temperatures differ (OpenAI/Anthropic: 1.0; Google: 0.7). To bound this, we re-ran Google standard at temperature 1.0 for the C0 and C5 endpoints: the C5 change is small in both generations (Gen1 3.8% to 3.4%, $\Delta -0.4\text{pp}$, 95% CI $[-1.7, +0.8]$; Gen2 5.9% to 6.1%, $\Delta +0.1\text{pp}$, 95% CI $[-2.0, +2.2]$), both intervals include 0, and the confound is bounded to roughly 2pp against an approximately 10pp cross-provider spread. Within-provider and cross-generational comparisons are unaffected. The extreme cross-provider ratio cited later (GPT-5.2 3.3% vs. Claude Sonnet 4.5 11.3%) is between two temperature-1.0 providers and is therefore not temperature-confounded.

3.1. Cross-Provider Baseline Hallucination (C0)

Under C0 (no anti-hallucination mechanisms), all models exhibit high hallucination rates across all domains (per-domain rates in Supplementary Table S3).

The C0 hallucination rates range from 59.2% to 74.0%. This means that around two-thirds of suggested tools do not appear in the confirmed tool box. Reasoning models show consistently equal or higher rates than their standard counterparts (cross-provider averages: 65.3% standard vs. 70.1% thinking). Thus, extended internal deliberation alone does not decrease ungrounded output.

3.2. Full Architecture Effectiveness (C5)

When all three mechanisms are in place (C5), there is a substantial reduction in inventory non-compliance (Table 5).

Table 5. Generation 1 hallucination rates (%) under C5 with C0-to-C5 reduction (36 queries per cell; $N = 864$). Cross-provider rate comparisons should be interpreted cautiously due to temperature differences (see Section 4.3).

Model	D1	D2	D3	D4	Avg	Reduction
<i>Standard Models</i>						
GPT-4.1	8.7	9.1	3.6	0.0	5.3	91.5%
Claude Sonnet 4.5	26.3	14.9	2.1	1.9	11.3	84.7%
Gemini 2.5 Flash-Lite	9.3	4.5	1.4	0.0	3.8	93.6%
<i>Reasoning Models</i>						
o4-mini	14.8	6.2	4.6	0.9	6.6	90.8%
Claude S. 4.5 (thn)	25.8	15.7	1.9	1.8	11.3	84.7%
Gemini 2.5 Flash	9.6	9.1	1.4	1.9	5.5	91.5%
<i>Cross-prov. avg (std)</i>	14.8	9.5	2.4	0.6	6.8	89.6%
<i>Cross-prov. avg (thn)</i>	16.7	10.3	2.6	1.5	7.8	88.9%

Italics mark the model-group labels and the cross-provider average rows.

The C0-to-C5 reduction is statistically significant for all twelve model configurations (Wilcoxon signed-rank, all $p < 0.001$; $r_{rb} = +0.994$ to $+1.000$; Cliff's $\delta = +0.824$ to $+1.000$, all large). This is the primary preplanned contrast and survives Bonferroni correction for the $k = 3$ planned tests (Section 2.11).

For standard Gen1 models, residual C5 rates are as low as 3.8% (Gemini Flash-Lite, 95% CI [2.7, 4.4]) and as high as 11.3% (Claude Sonnet 4.5, 95% CI [8.6, 12.5]). The lower absolute rates may be due to Google's lower temperature setting (0.7 vs. 1.0, see Section 4.3). All models come close to zero hallucination in domains D3 and D4 ($\leq 1.9\%$ for five of six models), which suggests that smaller, well-defined domains enable stronger grounding. The C0-to-C5 reduction holds across all 48 domain-model cells individually ($\alpha = 0.05$).

Under C5, reasoning models perform comparably, but slightly worse (cross-provider: 6.8% standard vs. 7.8% thinking; $\Delta C5 = +1.0pp$, $p = 0.008$, $r_{rb} = +0.189$, small). These standard-versus-thinking comparisons are not multiplicity corrected and are reported as exploratory (Section 2.11). None of the providers show a statistically significant gain for the reasoning mode. The only per-provider statistically significant result (Google, $r_{rb} = +0.414$) favors the standard configuration. The same conclusion is also confirmed when applying the Mann-Whitney U test as a robustness check, and is unchanged under the recall-equalized detector (Section 2.10).

3.3. Ablation Analysis

Table 6 presents the mention-aggregated hallucination rates for each ablation configuration in Generation 1.

Table 6. Generation 1 ablation results: mention-aggregated HR (%) per configuration, averaged across four domains (144 queries per cell; $N = 2592$), under the production detector (as published) and the recall-equalized detector adopted as the primary metric. The recall-equalized C0 baseline is higher, so the absolute C0-to-C5 reduction is larger in every cell.

Config	Mechanisms	GPT-4.1	Claude 4.5	Gemini Lite	Avg
<i>Production detector (as published)</i>					
C0	None	62.7	74.0	59.2	65.3
C1	M1 (Context)	17.6	35.5	13.2	22.1
C2	M2 (Vocab.)	9.7	22.1	4.8	12.2
C3	M3 (JSON)	74.0	80.0	72.1	75.4
C4	M1 + M2	7.5	22.6	2.9	11.0
C5	M1 + M2 + M3	5.3	11.3	3.8	6.8
C0→C5 reduction (pp)		57.3	62.7	55.4	58.5
<i>Recall-equalized detector (primary)</i>					
C0	None	73.0	79.6	69.4	74.0
C1	M1 (Context)	18.7	35.4	14.2	22.8
C2	M2 (Vocab.)	8.9	18.0	5.3	10.7
C3	M3 (JSON)	75.5	81.4	73.0	76.6
C4	M1 + M2	5.4	19.6	2.6	9.2
C5	M1 + M2 + M3	6.4	13.3	5.9	8.5
C0→C5 reduction (pp)		66.5	66.3	63.6	65.5

Italics mark the detector-panel labels.

Under the production detector, C3 (JSON only) appears strictly larger than C0 for all providers (+10.1pp cross-provider average). However, this gap is largely a detection-format artifact (Section 2.10): JSON fields make out-of-inventory tools easy to extract, while the same real tools are missed in free-text C0. Under the recall-equalized detector the C3-minus-C0 gap collapses to a cross-provider average of +2.6pp and is non-significant for all Gen1 providers (Wilcoxon, all $p > 0.1$; pooled Cliff's $\delta = +0.113$, negligible); these C3-

versus-C0 tests are not multiplicity corrected and are reported as exploratory (Section 2.11). Configuration ordering for the grounded configurations is statistically significant (Friedman test, all $p < 0.001$). The largest reduction occurs at C1, which goes from 74.0% to 22.8% (corrected), while C2 further reduces the rate to 10.7%. C4 approaches C5 (9.2% vs. 8.5% corrected); the C2–C5 and C4–C5 contrasts are planned and Bonferroni-corrected for $k = 3$, and the C2–C5 contrast is significant for OpenAI and Anthropic ($p_{\text{adj}} < 0.001$) but not Google ($p_{\text{adj}} = 1.0$).

3.4. Standard Models vs. Reasoning Models

We report two distinct reasoning comparisons that must not be conflated. The Generation 1 comparison is cross-family: each provider’s standard and reasoning conditions use different model checkpoints (e.g., OpenAI GPT-4.1 vs. o4-mini), so any Gen 1 difference confounds reasoning depth with model identity and is only a coarse, between-model contrast (Table 7). The Generation 2 comparison is a clean within-model toggle: the same checkpoint is evaluated with reasoning off and on, isolating the effect of reasoning itself (Section 3.6.4). Both comparisons use a partial factorial design that evaluates the C0 and C5 endpoints (and, for Generation 2, the C3 endpoint; see below), not the full intermediate ladder C1–C4; we therefore restrict reasoning claims to these anchor configurations.

Table 7. Generation 1 standard vs. reasoning model comparison under C0 and C5. $\Delta C5 = HR_{\text{thinking}} - HR_{\text{standard}}$ (percentage points (pp)); positive = higher hallucination under thinking. $N = 288$ per row.

Provider	Mode	C0 HR (%)	C5 HR (%)	Reduction	$\Delta C5$
OpenAI	Standard	62.7	5.3	91.5%	+1.3pp
	Reasoning	72.3	6.6	90.8%	
Anthropic	Standard	74.0	11.3	84.7%	$\pm 0.0\text{pp}$
	Reasoning	73.6	11.3	84.7%	
Google	Standard	59.2	3.8	93.6%	+1.7pp
	Reasoning	64.5	5.5	91.5%	
<i>Average</i>	<i>Standard</i>	65.3	6.8	89.6%	+1.0pp
	<i>Reasoning</i>	70.1	7.8	88.9%	

Bold marks the better value within each standard–reasoning pair; italics mark the cross-provider average rows.

3.5. Response Characteristics

Table 8 summarizes the response characteristics of Generation 1 models under the full architecture (C5).

Table 8. Generation 1 response characteristics per query under C5 ($n = 144$ per model).

Model	Tools/Q	Out Tok.	Latency (s)	Think Tok.	JSON%	Lat. Mult.
GPT-4.1	8.8	1162	19.5	—	99.3	1.53×
o4-mini	9.9	1026	29.8	1673	98.6	
Claude Sonnet 4.5	15.6	2843	65.6	—	97.9	0.95×
Claude S. 4.5 (thn)	12.7	2424	62.0	529	99.3	
Gemini 2.5 Flash-Lite	12.7	1467	6.0	—	100.0	2.70×
Gemini 2.5 Flash	14.0	374	16.2	1717	78.5	

Claude Sonnet 4.5 produced 77% more tools per query than GPT-4.1 (15.6 vs. 8.8), though the verbosity–HR correlation is weak (see Section 4.1.1). Under C5, $E[H/Q]$ ranges from 0.47 (GPT-4.1) to 1.76 (Claude Sonnet 4.5), a 4–8× reduction from C0. Gemini 2.5 Flash

shows degraded JSON validity (78.5% vs. 100% for Flash-Lite), and the thinking tokens (529–1717) add overhead without reducing hallucination.

3.6. Generation 2 Results

Generation 2 evaluates GPT-5.2, Claude Sonnet 4.6, and Gemini 3.1 Flash-Lite using the same protocol, with 3456 additional API calls. All three providers now use a single model with a configurable thinking toggle.

3.6.1. Gen2 Full Architecture Effectiveness (C5)

Table 9 presents the Generation 2 hallucination rates under C5 with the corresponding C0-to-C5 reduction.

Table 9. Generation 2 hallucination rates (%) under C5 with C0-to-C5 reduction (36 queries per cell; $N = 864$).

Model	D1	D2	D3	D4	C5 Avg	Reduction
<i>Standard Models (C5)</i>						
GPT-5.2	5.9	6.8	0.2	0.2	3.3	94.6%
Claude Sonnet 4.6	21.9	15.6	10.5	5.2	13.3	81.6%
Gemini 3.1 Flash-Lite	10.4	11.1	2.0	0.3	5.9	91.2%
<i>Reasoning Models (C5)</i>						
GPT-5.2 (thn)	4.3	8.1	0.2	0.6	3.3	94.6%
Sonnet 4.6 (thn)	24.6	16.6	11.6	6.9	14.9	79.3%
Gemini 3.1 Flash-Lite (thn)	8.3	11.2	1.5	0.0	5.2	92.4%
<i>Cross-prov. avg (std)</i>	12.7	11.1	4.3	1.9	7.5	88.8%
<i>Cross-prov. avg (thn)</i>	12.4	12.0	4.4	2.5	7.8	88.7%

Italics mark the model-group labels and the cross-provider average rows.

Gen2 C5 results confirm that the architecture generalizes across providers. Under the production detector GPT-5.2 achieves the lowest residual rate at 3.3%, followed by Gemini 3.1 Flash-Lite at 5.9% and Claude Sonnet 4.6 at 13.3%. Whether Anthropic regresses or improves across generations depends on the detector and is examined under the recall-equalized cross-generational comparison (Section 3.6.3). The near-zero D3/D4 pattern holds (0.0%–11.6%).

3.6.2. Gen2 Ablation Analysis

Table 10 presents the Generation 2 ablation results across all configurations and providers; Figure 2 shows the corresponding heatmap.

Under the production detector the C3-minus-C0 gap appears larger in Gen2 (+15.1pp) than Gen1 (+10.1pp). Under the recall-equalized detector this gap narrows to a cross-provider average of +4.8pp (Figure 3) and is statistically significant only for GPT-5.2 (+7.5pp, $p \approx 2 \times 10^{-4}$, Cliff's $\delta = +0.382$, medium) and Claude Sonnet 4.6 (+5.5pp, $p \approx 2 \times 10^{-5}$, $\delta = +0.329$, small); Gen2 Gemini is small and non-significant (+1.3pp, $p = 0.30$). The corrected C3 effect is therefore a current-generation, model-specific phenomenon, not a universal one. The Gen2 ordering for the grounded configurations remains $\{C5, C4, C2\} < C1 < C0$ (Friedman, all $p < 0.001$). C2 alone (5.2%) approaches C5 (6.9%) and C4 (6.0%, corrected). Under the recall-equalized detector the incremental contribution of M3 (C4 to C5) provides no consistent benefit: it significantly increases hallucination for OpenAI ($p_{\text{adj}} = 0.039$, $r_{rb} = +0.354$) and is non-significant for Anthropic ($p_{\text{adj}} = 0.51$) and Google ($p_{\text{adj}} = 0.61$). The vocabulary constraint alone (C2) is therefore at least as good as the full architecture (C5) for most current-generation provider cells.

Table 10. Generation 2 ablation results: mention-aggregated *HR* (%) per configuration, averaged across four domains (144 queries per cell; $N = 2592$), under the production detector (as published) and the recall-equalized detector adopted as the primary metric. Under recall-equalization the C3-minus-C0 gap narrows and is significant only for GPT-5.2 and Claude Sonnet 4.6.

Config	Mechanisms	GPT-5.2	Sonnet 4.6	Flash-Lite 3.1	Avg
<i>Production detector (as published)</i>					
C0	None	60.3	72.4	67.3	66.7
C1	M1 (Context)	13.7	26.2	17.4	19.1
C2	M2 (Vocab.)	2.9	15.2	6.5	8.2
C3	M3 (JSON)	85.6	82.6	77.3	81.8
C4	M1 + M2	3.6	16.3	4.5	8.1
C5	M1 + M2 + M3	3.3	13.3	5.9	7.5
C0→C5 reduction (pp)		57.1	59.1	61.3	59.2
<i>Recall-equalized detector (primary)</i>					
C0	None	78.5	77.7	77.5	77.9
C1	M1 (Context)	15.4	22.9	18.2	18.8
C2	M2 (Vocab.)	2.5	8.5	4.6	5.2
C3	M3 (JSON)	86.1	83.2	78.8	82.7
C4	M1 + M2	2.4	11.6	4.1	6.0
C5	M1 + M2 + M3	3.8	10.8	6.1	6.9
C0→C5 reduction (pp)		74.7	66.9	71.4	71.0

Italics mark the detector-panel labels.

3.6.3. Cross-Generational Comparison

Table 11 and Figure 4 compare hallucination rates under C5 between the two model generations.

Table 11. Cross-generational C5 hallucination rates (%) under the recall-equalized detector (primary); bold marks the better (lower) rate ($n = 144$ per cell). Under the production detector Anthropic instead appears to regress (standard 11.3 to 13.3); recall-equalization, which adds real out-of-inventory tools missed in Gen1 free text and removes non-specific labels in Gen2, shows it improving.

Provider	Mode	Gen1	Gen2	Δ	Direction
OpenAI	Standard	6.4	3.8	−2.6	↓ Improved
	Reasoning	7.6	3.6	−4.0	↓ Improved
Anthropic	Standard	13.3	10.8	−2.5	↓ Improved
	Reasoning	13.4	11.8	−1.6	↓ Improved
Google	Standard	5.9	6.1	+0.2	≈ Stable
	Reasoning	6.4	5.1	−1.3	↓ Improved
<i>Average</i>	<i>Standard</i>	8.5	6.9	−1.6	↓ Improved
	<i>Reasoning</i>	9.1	6.8	−2.3	↓ Improved

Arrows indicate the direction of change from Gen1 to Gen2 (↓ = improved, ≈ = essentially unchanged); italics mark the cross-provider average rows.

Under the recall-equalized detector, the cross-provider C5 average is comparable across the two generations (standard: 8.5% Gen1 vs. 6.9% Gen2; reasoning: 9.1% vs. 6.8%), with two of three providers improving (OpenAI −2.6pp, Anthropic −2.5pp) and Google essentially flat (+0.2pp). Under the production detector Anthropic instead appeared to regress (+2.0pp); recall-equalization accounts for the difference (Section 2.10). These cross-generation comparisons rest on two timepoints, are not multiplicity corrected, and are reported as exploratory (Section 2.11); they cannot establish a temporal trend. At the query level (P_{any}), 25%–96% of responses contain at least one out-of-inventory mention; the H2-only rate remains approximately 1–4%.

3.6.4. Gen2 Standard vs. Reasoning Models

Table 12 compares standard and reasoning-mode performance for Generation 2 models.

Table 12. Generation 2 standard vs. reasoning comparison under C0 and C5. $\Delta C5 = HR_{\text{thinking}} - HR_{\text{standard}}$ (pp). All Gen2 providers use single-model thinking toggles. $N = 288$ per row. Reasoning is additionally evaluated at C3 (Section 3.6.4); it does not amplify the apparent C3 effect (production C3 HR: standard 85.6/82.6/77.3%, thinking 86.3/83.9/78.7% for GPT-5.2/Sonnet 4.6/Flash-Lite 3.1).

Provider	Mode	C0 HR (%)	C5 HR (%)	Reduction	$\Delta C5$
OpenAI	Standard	60.3	3.3	94.6%	$\pm 0.0\text{pp}$
	Reasoning	61.8	3.3	94.7%	
Anthropic	Standard	72.4	13.3	81.6%	+1.6pp
	Reasoning	72.3	14.9	79.3%	
Google	Standard	67.3	5.9	91.2%	−0.7pp
	Reasoning	69.1	5.2	92.4%	
<i>Average</i>	<i>Standard</i>	66.7	7.5	88.8%	+0.3pp
	<i>Reasoning</i>	67.7	7.8	88.5%	

Italics mark the cross-provider average rows.

Under Gen2’s single-model-with-toggle design, only Anthropic shows a statistically significant standard-vs-reasoning difference (Sonnet 4.6 reasoning +1.6pp worse, $p < 0.005$); OpenAI and Google show no meaningful difference (both $p > 0.20$). The cross-provider aggregate effect remains negligible.

Reasoning and the C3 configuration. To test whether reasoning interacts with the JSON-only constraint, we additionally ran all three Generation 2 models in thinking mode at C3. Reasoning does not amplify the gap: the cross-provider C3-minus-C0 gap is +15.1pp for standard and +15.2pp for thinking, a negligible difference. Per-provider thinking C3 HR values closely track their standard counterparts (GPT-5.2 86.3%, Claude Sonnet 4.6 83.9%, Gemini 3.1 Flash-Lite 78.7%), and the standard-vs-thinking difference is non-significant for every provider (all 95% CIs include 0). Because both endpoints are JSON, this comparison is unaffected by the detection-format correction. This closes the open question of whether reasoning depth could worsen ungrounded structured-output hallucination: it does not, and reasoning provides no benefit at the C3 constraint, just as at C0 and C5.

3.6.5. Gen2 Response Characteristics

Table 13 summarizes the response characteristics of Generation 2 models under C5.

Table 13. Generation 2 response characteristics per query under C5 ($n = 144$ per model).

Model	Tools/Q	Out Tok.	Latency (s)	Think Tok.	JSON%	Lat. Mult.
GPT-5.2	14.2	2057	34.5	—	100.0	1.33×
GPT-5.2 (thn)	14.3	2103	45.9	596	100.0	
Claude Sonnet 4.6	19.0	3109	60.8	—	96.5	1.29×
Sonnet 4.6 (thn)	19.0	3768	78.4	514	100.0	
Gemini 3.1 Flash-Lite	8.0	830	4.2	—	100.0	2.26×
Gemini 3.1 Flash-Lite (thn)	7.2	1 *	9.5	1708	100.0	

* Reported by API; actual thinking content is substantially longer due to a token-counting artifact in the Gemini API.

Gen2 patterns mirror Gen1: Claude Sonnet 4.6 is the most verbose (19.0 tools/query), GPT-5.2 increases verbosity (8.8 to 14.2) yet reduces hallucination, and Gen2 $E[H/Q]$ under C5 ranges from 0.34 (Flash-Lite 3.1 thinking) to 2.89 (Sonnet 4.6 thinking). Gen2 thinking

overhead is comparable to Gen1 ($1.29\text{--}2.26\times$ vs. $0.95\text{--}2.70\times$ in Gen1) and all Gen2 thinking models achieve 100% JSON validity, yet only Anthropic shows a significant thinking penalty, and no provider shows a thinking advantage.

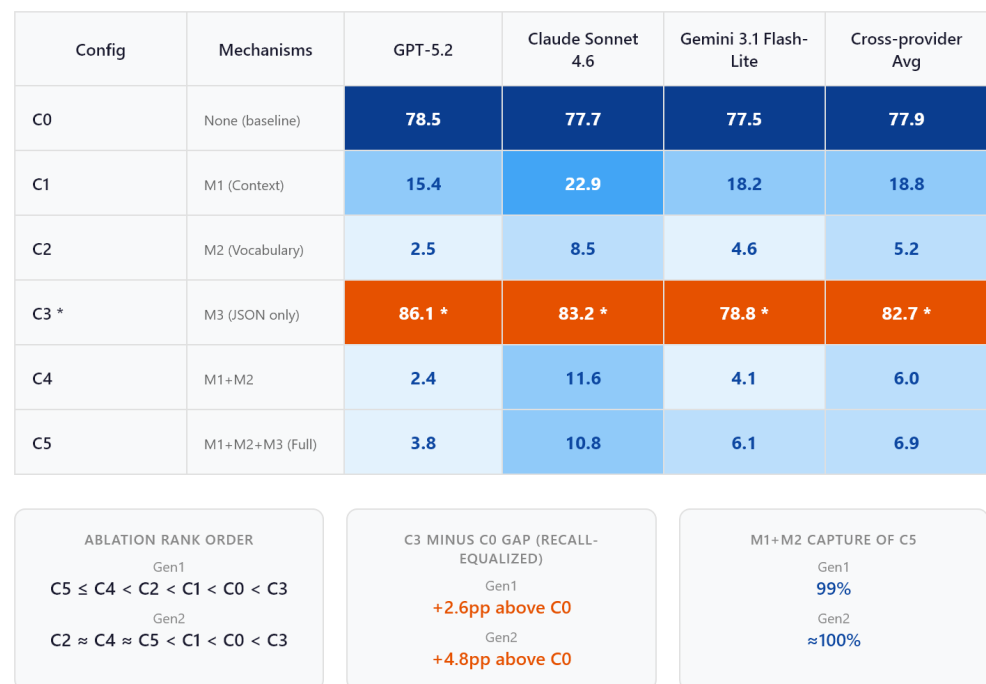


Figure 2. Generation 2 ablation heatmap: *HR* (%) across C0–C5 ($n = 144$ per cell). Rank ordering preserved ($C5 = C4 = C2 < C1 < C0 < C3$). Under the production detector C3 exceeds C0 for all providers; recall-equalization narrows this gap to +4.8pp (Gen2) vs. +2.6pp (Gen1) and leaves it significant only for GPT-5.2 and Claude Sonnet 4.6 (Figure 3). The asterisk (*) marks the C3 (JSON-only) cells, whose apparent excess over C0 is the detection-format bias examined in Figure 3.

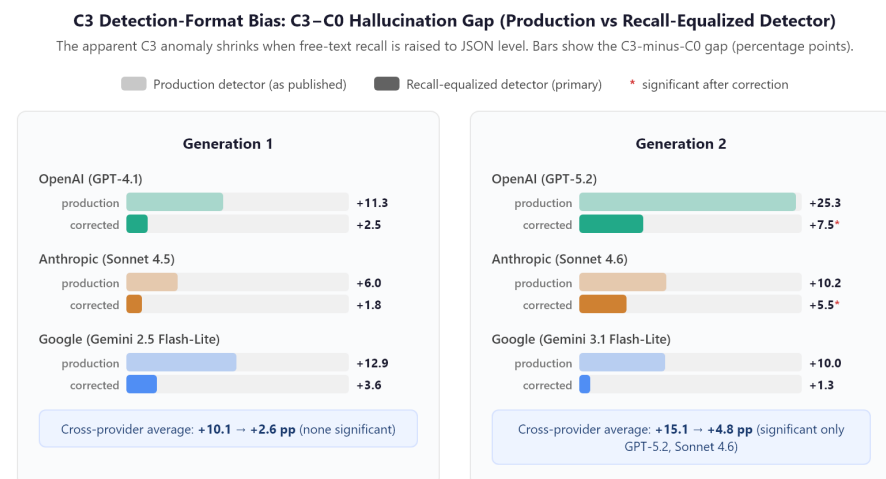
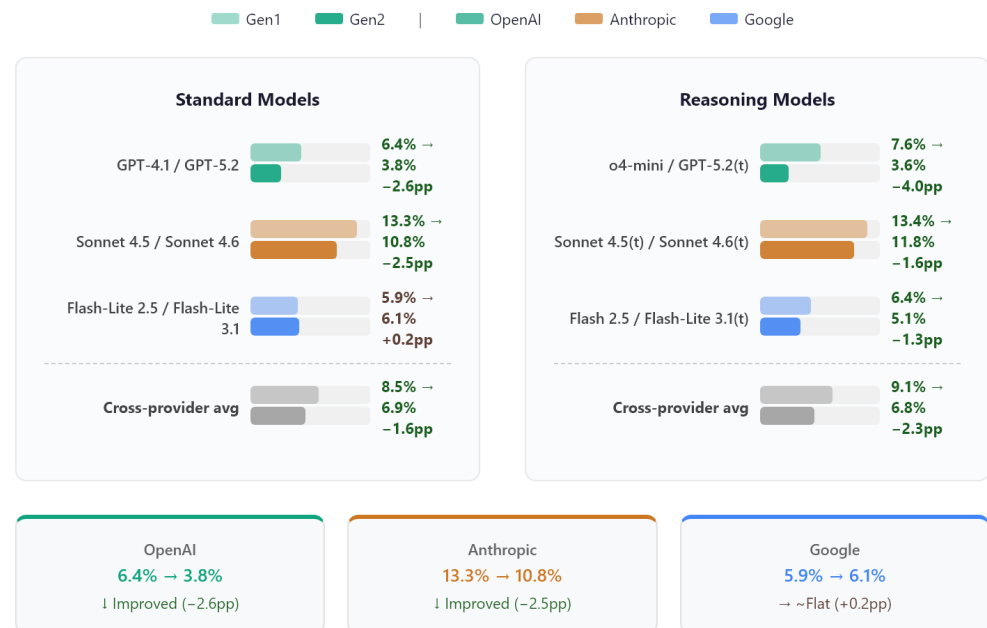


Figure 3. C3 detection-format bias: C3-minus-C0 hallucination gap under the production detector (which extracts tool names from JSON fields) versus the recall-equalized detector (free-text recall raised to JSON level). Equalizing recall shrinks the cross-provider gap from +10.1pp to +2.6pp (Gen1) and from +15.1pp to +4.8pp (Gen2). After correction the gap is statistically significant only for GPT-5.2 (+7.5pp) and Claude Sonnet 4.6 (+5.5pp). Bar colors denote providers (green = OpenAI, orange = Anthropic, blue = Google); within each provider, the lighter bar is the production detector and the darker bar the recall-equalized detector; red asterisks mark gaps that remain statistically significant after correction.



Under the recall-equalized (corrected) detector, OpenAI improves and Anthropic improves (13.3% → 10.8%, not a regression), Google stays roughly flat (5.9% → 6.1%), and the cross-provider standard-model average improves slightly (8.5% → 6.9%). These are two timepoints, not a temporal trend.

Figure 4. Cross-generational C5 hallucination rates under the recall-equalized detector: Gen1 (faded) vs. Gen2 (solid). OpenAI improves (−2.6/−4.0pp) and Anthropic improves (−2.5/−1.6pp); Google is essentially flat (+0.2/−1.3pp). The cross-provider average is comparable across generations (8.5% to 6.9%). Under the production detector Anthropic instead appears to regress; see Table 11. Bar colors denote providers (green = OpenAI, orange = Anthropic, blue = Google; gray = cross-provider average), with faded bars for Generation 1 and solid bars for Generation 2.

4. Discussion

4.1. Key Findings and Mechanism Analysis

The hallucination mitigation transfers across providers and generations: *HR* is reduced by 83%–96% across all twelve configurations (recall-equalized), and the grounded ablation ordering {C5, C4, C2} < C1 < C0 is preserved across all six standard models. The apparent C3 > C0 gap, by contrast, is largely a detection-format artifact (Section 2.10): under recall-equalized detection it survives as a small, significant effect only for two current-generation models (GPT-5.2, Claude Sonnet 4.6). At the same time, residual hallucination under the full architecture differs considerably by model, by up to a factor of 3.5 (e.g., recall-equalized 3.8% for GPT-5.2 vs. 13.3% for Claude Sonnet 4.5). This variation suggests that identical architectural constraints may still interact differently with each model’s vocabulary adherence behavior.

A practical takeaway is the comparability of the architecture’s effectiveness across the two generations tested. Under the recall-equalized detector the cross-provider C5 average is similar at the two timepoints (8.5% in Gen1 vs. 6.9% in Gen2 for standard models), with two of three providers improving and one essentially flat, which supports the idea that the three-mechanism design yields a small residual hallucination level under the tested settings. We do not yet have enough evidence to say whether this residual is a hard limit, or whether it can be pushed down further by better models and/or additional techniques such as constrained decoding or retrieval reranking. In multi-provider enterprise settings, this suggests that the architecture generally does not need to be redesigned for a new model generation, although provider-specific residual rates can still shift and per-provider directions diverge; two timepoints cannot establish a temporal trend.

4.1.1. Verbosity, Per-Mention Quality, and Hallucination

While Anthropic models are consistently more verbose and also show higher residual inventory non-compliance, the relationship between tools-per-query and hallucination rate across all 12 model configurations is weak and not statistically significant (Pearson $r = 0.31$, $p = 0.31$, $n = 12$). The main reason is that provider identity confounds the relationship: Anthropic models cluster at high verbosity and high *HR*, while the other providers do not show a consistent trend.

Two cross-generational observations illustrate why verbosity alone is not a good explanation. OpenAI increases tools/query substantially from Gen1 to Gen2 (8.8 $\rightarrow$ 14.2 tools/query) while reducing hallucination under C5 (recall-equalized 6.4% $\rightarrow$ 3.8%). Conversely, Gemini 3.1 Flash-Lite produces fewer tools/query than Flash-Lite 2.5 (8.0 vs. 12.7), yet its aggregate C5 rate is marginally higher (recall-equalized 6.1% vs. 5.9%).

A more informative view is the per-tool-mention hallucination rate. Under C5 and the recall-equalized detector, the per-mention hallucination rates closely track the overall *HR* regardless of output length (e.g., 4.0% for GPT-5.2 versus 13.2% for Claude Sonnet 4.5). This indicates that the remaining hallucination is mainly driven by the model's vocabulary adherence quality at the per-mention level, rather than by the total number of mentions generated. At the per-mention level Anthropic improves across generations (13.2% to 11.7%), consistent with the recall-equalized cross-generational result.

4.1.2. Existence Hallucination vs. Recommendation Quality

Our primary metric (*HR*) evaluates whether the recommended tools exist in the Online-CADCOM database, rather than whether they are the most suitable tools for the user's query. Existence hallucination and recommendation quality are separate concerns. Our contribution is therefore existence hallucination reduction: it is necessary for appropriateness, but not sufficient. Existence verification is objective and automatable, while appropriateness evaluation is more subjective.

That said, we observe that under C5, models tend to organize recommendations around tools that actually belong to the queried domain rather than suggesting generic alternatives. This is an automated, inventory-side relevance proxy and not a user-centered appropriateness judgment. We did not collect user ratings, task-completion outcomes, or expert appropriateness scores, so claims about whether recommended tools are the most suitable for a given query are outside our evidence base. A systematic comparison against the platform's MCDA-based rankings and a user-centered appropriateness study are left to future work.

4.2. Practical Implications and Deployment

As a post hoc analysis, we applied a stoplist of 62 ultra-generic tokens to remove extraction artifacts (this filtering is used only for this decomposition analysis and does not change the *HR* values reported in Results). We then manually classified the 197 most frequent out-of-inventory tool names (covering 76.5% of 20,430 tool-like mentions across 6912 queries) into four categories using web verification. Evidence was based on a vendor product page, a GitHub/SourceForge repository, or a Wikipedia entry, and the evidence URLs are provided in the Supplementary Material. The classification was performed independently by two raters: the first author, and an independent second rater (a university researcher with an engineering background, not an author of this study) who classified all 197 names blind to the first author's labels, using the same category definitions and his own web searches. Inter-rater agreement was substantial (Cohen's $\kappa = 0.645$, 95% CI [0.55, 0.74]; raw agreement 77.7%, 153/197), and the two raters agreed unanimously (197/197) that none of the names are a fabricated (H1) tool name. The 44 disagreements, all on the boundaries between the H2, near-miss, and non-specific categories, were resolved by a

consensus discussion (29 resolved to the first rater's label, 12 to the second rater's, and 3 to a jointly agreed third category); the percentages below use the consensus labels, which shift the decomposition by at most 2.1pp relative to the first rater's labels. Each label is grounded in objective evidence (a vendor or repository product page, a database lookup, or a Wikipedia entry, with evidence URLs in the Supplementary Material). We also ran a sensitivity analysis on the 14 ambiguous names that were resolved conservatively, which showed the decomposition remains stable within $\pm 4.5\text{pp}$ (maximum shift). Of the 15,634 classified mentions:

- 57.5% (8994) are H2 (external-real) tools (e.g., LTspice, Altium Designer, PSpice).
- 24.8% (3875) are near-miss references to inventory tools.
- 17.7% (2765) are non-specific mentions (generic labels). The recall-equalized analysis (Section 2.10) shows these labels do not account for the residual C3 gap, which consists of real out-of-inventory tools.
- 0.0% are H1 (fabricated) within the top-197 names.

Operationally, the H2-only C5 rate is 2.9% (vs. headline $HR \sim 7.4\%$), corresponding to a 91.1% reduction from C0; the C5-level H2 figures are identical under the first-rater and consensus labels, because none of the relabeled names occurs in C5 responses. Per-provider C5 H2-only rates are 1.3% (OpenAI), 2.8% (Google), and 4.1% (Anthropic). Even under the worst-case reclassification scenario, where all ambiguous names are counted as H2, the upper bound reaches 5.3%, which still represents an 84% reduction from C0. At the query level, $P_{\text{any,H2}} = 17.5\%$ compared to $P_{\text{any}} = 49.5\%$, indicating that roughly two-thirds of contaminated C5 responses involve only near-miss or non-specific mentions.

4.2.1. Reasoning Models and Hallucination

Reasoning provides no consistent benefit under C5 in either generation, but the two results carry different evidential weight. The Generation 1 contrast is cross-family (different checkpoints per condition); here the direction happens to favor standard models (Gen1 Google, $p < 0.001$; Gen1 cross-provider aggregate, $p = 0.008$; cross-provider C5 6.8% standard vs. 7.8% thinking), but because model identity is confounded with reasoning depth this is only suggestive. The Generation 2 contrast is a clean within-model toggle and is the load-bearing evidence: per-provider $\Delta C5$ ranges from -0.7 to $+1.6\text{pp}$, only Anthropic reaches significance ($+1.6\text{pp}$, $p < 0.005$), and the cross-provider aggregate is negligible (7.5% vs. 7.8%; all per-provider $|r_{rb}| \leq 0.41$). Together these extend the pattern reported by Yao et al. [9] to the constrained setting, with the toggle-based Gen2 result providing the cleaner test; the no-benefit conclusion is unchanged under the recall-equalized detector.

From a practical standpoint, thinking models add 514–1717 additional thinking tokens and substantial latency overhead (Sections 3.5 and 3.6.5) without producing any measurable reduction in out-of-inventory recommendations.

4.2.2. Mechanism Contribution Analysis

Interpretation caveat: this ablation isolates configuration-level effects rather than strictly causal mechanisms, since M1 context includes tool names. C4 (Context + Vocabulary) achieves 11.0% (Gen1) and 8.1% (Gen2), capturing 92.8%–98.8% of the full C5 reduction. In Gen2, C4 performs nearly identically to C5 (8.1% vs. 7.5%).

The C3 configuration: a detection-format bias and a small current-generation effect.

Under the production detector, C3 (JSON only, no grounding) appears to increase hallucination over C0 by $+10.1\text{pp}$ (Gen1) and $+15.1\text{pp}$ (Gen2). As established in Sections 2.10 and 3.6.2, this gap is largely a detection-format artifact: re-scoring all standard responses with the recall-equalized detector collapses it to a cross-provider average of $+2.6\text{pp}$ (Gen1) and $+4.8\text{pp}$ (Gen2) (Figure 3), and after correction the residual is significant

only for two current-generation models, GPT-5.2 (+7.5pp) and Claude Sonnet 4.6 (+5.5pp), with all Gen1 models and Gen2 Gemini small and non-significant. The effect is thus a small, current-generation, model-specific phenomenon rather than a universal one. This small residual is consistent with prior findings that constraining models to rigid output formats such as JSON can reduce their performance [30].

We initially hypothesized that mandatory `toolName`/`toolId` fields create slot-filling pressure filled with non-specific labels. The corrected analysis does not support this: a detector variant that additionally counts non-specific labels yields essentially the same residual gap as the tools-only variant for every model, so the residual over-generation under C3 consists of real out-of-inventory tools, not slot-filling junk. We therefore drop the slot-filling hypothesis. The remaining current-generation residual is consistent with the structured-output and grounding interaction reported by Ayala and Bechard [15] and indicates that ungrounded JSON enforcement still warrants caution for the strongest current models, though the effect is far smaller than the production detector suggested.

Practical implications.

- **Minimum effective configuration:** M1 + M2 (C4) is the minimum effective configuration, as quantified in the mechanism-contribution analysis above.
- **Standard models suffice:** Reasoning modes provide no consistent advantage under C5 while adding token and latency overhead.
- **JSON enforcement requires caution:** The apparent ungrounded-C3 increase is largely a detection-format artifact (Section 2.10), and the corrected residual is significant only for two current-generation models. When grounded, M3 provides no consistent benefit and can slightly hurt (it significantly increases inventory non-compliance for Gen2 OpenAI), so add M3 only when machine-parseable JSON output is required.
- **Per-provider validation:** Because residual inventory non-compliance differs by model (Section 4), post-generation name verification can further reduce out-of-inventory outputs.
- **Vocabulary scaling:** The 82-tool inventory fits entirely in the prompt, so M2 here is a complete constraint. For substantially larger inventories, M2 would require a top-*K* retrieval step, which introduces a retrieval recall ceiling and converts M2 into an approximate constraint; whether the reduction reported here holds in that regime is untested.

Suggested deployment heuristic (M3: JSON enforcement).

- If $C4 \approx C5$ (no significant difference), M3 can be omitted unless JSON output is required for parsing.
- If $C4 > C5$ significantly: M3 is not recommended.
- If $C5 > C4$ significantly: the full C5 architecture is recommended.

In all cases, M1 + M2 represents the minimum effective configuration. Total cost: \$180.98 across 6912 calls (\$0.0008–\$0.1078 per query); details in Supplementary Materials.

Qualitative example.

For example, under C0, GPT-5.2 recommends Altium Designer, OrCAD, and Cadence Allegro, which are real tools but absent from the inventory (100% out-of-inventory). Under C5, the same query instead yields KiCad, PCB Calculator, and Impedance Calculator, all verified and resulting in 0% HR. Additional examples are provided in the Supplementary Materials.

Contributions. This work makes four main contributions. First, the three-mechanism anti-hallucination architecture is effective, and that effectiveness transfers across providers: across twelve LLM configurations from three major commercial providers (OpenAI, Anthropic, Google) and 6912 controlled API calls, the full architecture (C5) reduces hallucination consistently across provider ecosystems, and the C0-to-C5 reduction holds and strengthens to 64–75 pp per provider, under a recall-equalized detector. Second, this ef-

fectiveness is comparable across two model generations on the cross-provider average (recall-equalized C5: 8.5% Gen1, 6.9% Gen2; two of three providers improve and one is flat, none materially regress), though per-provider rates shift, so we describe the architecture as comparable in aggregate rather than uniform, and we note that two timepoints cannot establish a temporal trend. Third, reasoning-mode models provide no significant benefit over their standard counterparts under architectural constraints, across all providers and both generations and including at C3. Fourth, we contribute a methodological finding on detection-format bias: a detector that extracts tool names from JSON fields catches out-of-inventory tools in JSON-only output (C3) but misses most of the same real tools in free-text baselines (C0), which inflates the apparent C3-minus-C0 gap (from +10.1pp to a recall-equalized +2.6pp in Gen1, and from +15.1pp to +4.8pp in Gen2); after this correction, ungrounded JSON enforcement carries a small effect that is statistically significant only for two current-generation models (GPT-5.2 +7.5pp, Claude Sonnet 4.6 +5.5pp) and absent otherwise, a current-generation rather than universal phenomenon that extends observations by Ayala and Bechard [15] on structured-output and hallucination interactions. Scope: all four contributions concern existence hallucination (inventory compliance), which is objective and automatable; user-centered evaluation of recommendation appropriateness, helpfulness, or user satisfaction is outside the scope of this study and is identified as future work.

4.3. Threats to Validity

Internal validity. All test prompts are generated programmatically and are identical across all models, which reduces prompt-design bias but does not fully reflect real user query distributions. The C0 baseline represents a condition with no mitigation effort, rather than the best achievable performance without the three mechanisms. The C0-to-C5 reduction should therefore be interpreted as an upper bound. Three repetitions per cell (144 queries) provide sufficient power for large effects but may miss smaller differences.

Temperature confound. Provider temperatures differ: OpenAI and Anthropic use the API default (1.0) and Google uses 0.7, a deliberate choice reflecting typical production defaults. Lower temperature may contribute to Google's lower Gen1 rates. To bound this, we reran Google standard at temperature 1.0 for the C0 and C5 endpoints in both generations; the C5 effect is negligible (Gen1 $\Delta -0.4\text{pp}$, 95% CI $[-1.7, +0.8]$; Gen2 $\Delta +0.1\text{pp}$, 95% CI $[-2.0, +2.2]$; both CIs include zero), bounding the temperature contribution to at most about 2pp against the roughly 10pp cross-provider spread. Within-provider comparisons (standard vs. thinking, C0 vs. C5, ablation ordering) and cross-generational comparisons are unaffected because temperature is constant within each provider. The apparent $C3 > C0$ gap appears for all providers including Google and is largely a detection-format effect (Section 2.10), not a temperature artifact. Cross-provider rankings should be read with this small residual confound in mind; the most extreme gap we report (GPT-5.2 3.3% vs. Claude Sonnet 4.5 11.3%) is between two temperature-1.0 providers and so cannot be attributed to temperature. The common-temperature control covers the C0 and C5 endpoints only, so a temperature interaction specific to the C3 configuration has not been tested.

Partial factorial limitation. The reasoning evaluation uses a partial factorial design that crosses standard vs. reasoning with only the C0 and C5 endpoints, rather than all six ablation configurations; we additionally added the C3 endpoint for Gen2 to test reasoning at the JSON-only constraint. Two further caveats apply. First, the Generation 1 reasoning comparison is cross-family (e.g., GPT-4.1 vs. o4-mini), so it confounds reasoning depth with model identity and should be read as a between-model contrast rather than a clean reasoning toggle; the within-model Gen2 toggle is the cleaner test. Second, restricting to C0, C3, and C5 limits our ability to characterize reasoning effects on the intermediate

configurations C1, C2, and C4. However, the negligible standard-vs-thinking differences observed at C5 (0.0–1.7pp across all providers) and the absence of C3 amplification under reasoning suggest that intermediate configurations would show similarly negligible reasoning effects, as the grounding mechanisms appear to dominate over reasoning depth at all tested constraint levels.

Construct validity. Hallucination detection uses a three-strategy extraction pipeline with multi-level grounding classification (see Section 2.9). Automated validation on 50 stratified samples estimates detection precision at 100% (zero false positives across 485 extracted mentions) and recall at approximately 94.5% overall (>99% for JSON-enforced responses, lower for free-form C0, which accounts for 27 of the 28 undetected mentions). Because false negatives are concentrated in unstructured C0 responses (where regex-based Strategy 3 is the primary detector), the measured C0 baseline hallucination rate is a conservative lower bound; any detection bias therefore works against our reported C0-to-C5 reduction, making the improvement a conservative estimate. The per-mention *HR* metric does not capture query-level user experience; we additionally report P_{any} (query-level hallucination probability) to address this gap. Here the metric measures hallucination for existence only and ignores the relevancy of the suggested grounded tools for the input search query, which is further discussed in Section 4.2; user-centered appropriateness of the grounded recommendations was not evaluated and is left to future work. The manual H1/H2 decomposition in Section 4.2 was independently verified by a second rater blind to the first author's labels; inter-rater agreement was substantial (Cohen's $\kappa = 0.645$), the absence of fabricated (H1) names was confirmed unanimously, and the remaining disagreements were resolved by consensus (Section 4.2). The residual hallucination rate reported for C5 may also reflect limitations of our detection rather than the true behavior of the model. The M3 JSON schema includes a field called `toolId` and a field called `toolName`. For this analysis we have only tested if the tool name is present in the tool inventory, but not whether the model outputs the correct or consistent `toolId`. Verifying whether the model returns a correct `toolId` would require a separate matching step against the database primary keys, and is left to future work.

Detection sensitivity. To assess whether detector parameter choices affect the robustness of hallucination measurement, we conducted two sensitivity analyses across all 6912 responses. First, we identified three potentially ambiguous tool names in the D1–D4 evaluation domains (“EAGLE,” “jcalc,” “Maddox”) and nine additional ambiguous names in non-evaluation categories. Across all responses, every occurrence of these names was classified as grounded (zero hallucinated occurrences), and excluding them from analysis changes C5 hallucination rates by <1 percentage point per provider. Second, we swept the fuzzy containment ratio threshold (0.45–0.70, production value 0.55) and Levenshtein edit distance threshold (0–3, production value 2) across all 24 combinations. The conclusion that the full architecture reduces hallucination ($C5 < C0$) holds for all 24 threshold combinations in both model generations. We additionally tested whether the apparent $C3 > C0$ gap is a detection-format artifact by re-scoring all standard responses with the recall-equalized detector of Section 2.10; the C3-minus-C0 gap narrows from +10.1/+15.1pp to +2.6/+4.8pp (Gen1/Gen2) and survives as a significant effect only for GPT-5.2 and Claude Sonnet 4.6, while the C0-to-C5 reduction is unchanged and in fact larger under the corrected detector. The maximum reclassification rate across threshold combinations is 2.0% of the 5580 unique tool names. These results indicate that the reported findings are robust to reasonable variations in detection parameters.

Inventory-relative measurement. Our *HR* metric captures all tool mentions absent from the 82-tool verified inventory, including commercially available tools not represented in our database (H2). A frequency-weighted audit of the 197 most frequent out-of-inventory

names (76.5% of all mentions) found that 57.5% are H2 (external-real tools such as LTspice, Altium Designer, and PSpice), 24.8% are near-miss references to inventory tools, 17.7% are non-specific mentions, and 0% are H1 (fabricated). This means *HR* slightly overstates the “confabulation” rate while accurately measuring inventory non-compliance, which is the operationally relevant metric for a curated recommendation platform. A full ground-truth classification of all 2870 unique consolidated names would require exhaustive cross-referencing against comprehensive commercial software databases, infeasible at scale but addressed for the high-frequency names that dominate the distribution. Importantly, the architecture reduces both H2 and near-miss mentions (82.8% and 55.2% C0-to-C5 reduction in mention counts, respectively), indicating that the grounding mechanisms constrain recommendations to the verified inventory regardless of whether alternatives are real or fabricated.

External validity. Results are derived from one engineering tool recommendation research platform (Online-CADCOM) with 82 tools across four domains. Crucially, all four domains (PCB design, PCB design calculators, switched-mode power supplies and converters, and transformers) belong to a single electronic-design and power-electronics super-domain and share substantial vocabulary and tool overlap. Our findings should therefore be read as characterizing this homogeneous family of related domains; we make no claim that they extend to dissimilar recommendation domains (e.g., medical, legal, or consumer products) or to other database sizes, which require further validation. Three providers are tested (representing major commercial LLM API providers), but open-source models (Meta Llama, Mistral) are excluded. In Generation 1, the three providers implement reasoning capabilities differently (separate model, toggle, variant), making cross-provider reasoning comparisons approximate; Generation 2 partially resolves this as all three converge on a single-model paradigm with configurable thinking toggles. The two-generation design shows that the cross-provider C5 average is similar at the two timepoints measured, but this is an average over divergent per-provider trajectories (some providers improve, others are flat), and “generation” is not a controlled variable: the three providers’ Gen1-to-Gen2 transitions differ in model family, training data, and release timing. Two timepoints are therefore not sufficient for claims of “temporal stability,” generational trend extrapolation, or that the architecture is stable per provider; providers may also silently update model weights between measurement points, and longer cross-generational tracking with repeated sampling is needed to establish true temporal patterns. Additionally, M2’s closed-vocabulary constraint currently enumerates all 82 tool names directly in the prompt (around 200 tokens). Scaling to substantially larger inventories presents a clear practical challenge: listing 1000 tool names would require roughly 2500 prompt tokens, while 10,000 names would exceed the context limits of most models. A practical solution is a retrieval step (e.g., embedding-similarity or category-based filtering) to select a top-*K* candidate subset per query before constructing the M2 vocabulary list. However, this introduces a retrieval recall ceiling (if the optimal tool is not in the top-*K* set, it cannot be recommended), converting M2 from a complete constraint to an approximate one. The 82-tool setting isolates M2’s mechanism effectiveness independent of retrieval quality; whether the measured hallucination reduction degrades gracefully as inventory size grows (and M2 becomes retrieval-dependent) is an open empirical question. While the mechanisms are architectural and prompt-level, structured-output support differs across providers (e.g., OpenAI’s `response_format`, Anthropic’s tool-use blocks, Google’s `responseMimeType`) and requires minor per-provider parameterization; “provider-portable” refers to the architectural design, not a zero-adaptation deployment. The effectiveness is also not uniform per provider: the residual C5 rate varies by up to $3.5\times$ across models and the per-generation direction differs by provider, so portability should be understood as a similar cross-provider average rather

than equivalent per-provider behavior. Likewise, the retrieval-backed M2 variant needed for larger inventories would itself be provider- and embedding-dependent, so its recall ceiling cannot be assumed to transfer from the 82-tool setting measured here.

5. Conclusions

We presented a cross-provider, cross-generational study of hallucination mitigation in LLM-based tool recommendation, evaluating a three-mechanism architecture across twelve model configurations from three commercial providers and two model generations (6912 API calls). The central result is that the full architecture (C5) reduces mention-level inventory non-compliance consistently across providers and generations: under the recall-equalized detector adopted as the primary metric, rates fall from roughly 69–80% baselines to 4–13%, and the C0-to-C5 reduction is unchanged (and larger) under that detector. The cross-provider C5 average is similar at the two timepoints measured (8.5% Gen1, 6.9% Gen2), although per-provider trajectories diverge, so generation is not a controlled variable and we do not claim uniform cross-generational stability. At the query level, pooled $P_{\text{any}} \approx 49.5\%$: roughly half of responses still contain at least one out-of-inventory mention, so the user-facing impact exceeds the mention-level rate.

Three secondary findings emerge. First, the apparent C3 (ungrounded JSON) penalty is largely a detection-format artifact: under recall-equalized detection the C3-minus-C0 gap narrows to +2.6pp (Gen1) and +4.8pp (Gen2) and remains significant only for two current-generation models, so ungrounded JSON enforcement warrants modest caution rather than being universally counterproductive. Second, reasoning-mode variants provide no statistically significant benefit over their standard counterparts under architectural constraints, across both generations and all three providers, including at C3. Third, the minimum effective configuration (context plus vocabulary, C4) captures 92.8–98.8% of the full reduction, with residual rates varying by up to $3.5\times$ across models (as low as 3.8% for GPT-5.2).

These results derive from one engineering tool platform (Online-CADCOM) with 82 tools across four related electronic-design and power-electronics domains, where the full vocabulary fits within prompt context. Generalization to other domains, substantially larger inventories (where retrieval-based vocabulary selection would replace full enumeration), open-source models, and user-centered recommendation quality requires further validation, and future work should test partial grounding (M1+M3, M2+M3), formalize recall-equalized detection for free-text and structured outputs, and extend cross-generational tracking. For practitioners, the architecture enables reliable LLM-powered tool recommendation within curated engineering inventories and is portable across the three providers tested with only minor per-provider parameterization; because per-provider behavior and per-generation direction vary, deployments should be re-validated after model or provider changes rather than assuming unchanged effectiveness. More broadly, because the inventory is closed and the four domains are closely related, these results should not be read as evidence about open-domain or broad-catalog tool recommendation, where the candidate space is open-ended and a fixed in-prompt vocabulary is unavailable; that setting, and broader recommendation tasks generally, remain open for future work.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/ai7070273/s1>: Table S1: API parameters for reproducibility; Table S2: Query accounting across all experimental phases; Table S3: Generation 1 baseline hallucination rates (HR, %) under C0 across all models and domains.

Author Contributions: Conceptualization, L.M. and G.M.; methodology, L.M. and G.M.; software, L.M.; validation, L.M.; formal analysis, L.M.; investigation, L.M.; resources, L.M. and G.M.; data

curation, L.M.; writing of the original draft, L.M.; review and editing, L.M. and G.M.; visualization, L.M.; supervision, G.M.; project administration, G.M.; funding acquisition, G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by the European Regional Development Fund within the Operational Program “Bulgarian national recovery and resilience plan,” under the Project BG-RRP-2.004-0005 “Improving the research capacity and quality to achieve international recognition and resilience of TU-Sofia.” The APC was funded by the same project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All experimental data, source code, and analysis scripts supporting the findings of this study are publicly available at <https://github.com/nauka-lm/llm-tool-recommendation-replication> (accessed on 13 July 2026). The repository includes the 82-tool verification inventory, the H1/H2 classification dataset with evidence URLs, prompt templates for all six configurations (C0–C5), the evaluation framework source code (prompt builder, hallucination detector, metrics calculator), analysis scripts for all reported statistics and figures, and reproduction instructions.

Acknowledgments: The authors used Claude Sonnet 4.5 (Anthropic) to assist with LaTeX formatting and language editing. All the scientific content, interpretations, and conclusions are the original work of the authors. This article is a revised and expanded version of a paper entitled “Database-Grounded LLM Recommendations for Engineering Tool Selection: An Anti-Hallucination Architecture” [10], which was presented at the 1st International Conference on Industrial & Systems Engineering (ICISE-2026), Tirana, Albania, 16 April 2026.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Menlo Ventures. 2025: The State of Generative AI in the Enterprise. 2025. Available online: <https://menlovc.com/perspective/2025-the-state-of-generative-ai-in-the-enterprise/> (accessed on 27 February 2026).
2. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Madotto, A.; Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **2023**, *55*, 1–38. [CrossRef]
3. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–55. [CrossRef]
4. Menxhiqi, L.; Marinova, G. Dynamic expert module for tool selection in Online-CADCOM platform. In Proceedings of the 8th International Balkan Conference on Communications and Networking (BalkanCom), Piraeus, Greece, 17–20 June 2025.
5. Xu, H.; Zhu, Z.; Pan, L.; Wang, Z.; Zhu, S.; Ma, D.; Cao, R.; Chen, L.; Yu, K. Reducing Tool Hallucination via Reliability Alignment. In Proceedings of the 42nd International Conference on Machine Learning (ICML), Vancouver, BC, Canada, 13–19 July 2025.
6. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; Volume 33, pp. 9459–9474.
7. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 24824–24837.
8. Geng, S.; Josifoski, M.; Peyrard, M.; West, R. Grammar-constrained decoding for structured NLP tasks without finetuning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; pp. 10932–10952. [CrossRef]
9. Yao, Z.; Liu, Y.; Chen, Y.; Chen, J.; Fang, J.; Hou, L.; Li, J.; Chua, T.S. Are reasoning models more prone to hallucination? *arXiv* **2025**, arXiv:2505.23646.
10. Menxhiqi, L.; Marinova, G. Database-Grounded LLM Recommendations for Engineering Tool Selection: An Anti-Hallucination Architecture. In Proceedings of the 1st International Conference on Industrial & Systems Engineering (ICISE), Tirana, Albania, 16–17 April 2026.
11. Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. Siren’s song in the AI ocean: A survey on hallucination in large language models. *Comput. Linguist.* **2025**, *51*, 1373–1418. [CrossRef]

12. Shuster, K.; Poff, S.; Chen, M.; Kiela, D.; Weston, J. Retrieval augmentation reduces hallucination in conversation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 3784–3803. [\[CrossRef\]](#)
13. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv* **2024**, arXiv:2312.10997.
14. Peng, B.; Galley, M.; He, P.; Cheng, H.; Xie, Y.; Hu, Y.; Huang, Q.; Liden, L.; Yu, Z.; Chen, W.; et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv* **2023**, arXiv:2302.12813.
15. Ayala, O.; Bechard, P. Reducing hallucination in structured outputs via retrieval-augmented generation. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), Mexico City, Mexico, 16–21 June 2024; pp. 228–238. [\[CrossRef\]](#)
16. OpenAI. GPT-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
17. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 27730–27744.
18. Wu, H.; He, Z.; Zhang, X.; Yao, X.; Zheng, S.; Zheng, H.; Yu, B. ChatEDA: A large language model powered autonomous agent for EDA. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 3184–3197. [\[CrossRef\]](#)
19. Liu, M.; Ene, T.D.; Kirby, R.; Cheng, C.; Pinckney, N.; Liang, R.; Alben, J.; Anand, H.; Banerjee, S.; Bayraktaroglu, I.; et al. ChipNeMo: Domain-adapted LLMs for chip design. *arXiv* **2024**, arXiv:2311.00176.
20. Luo, L.; Li, Y.F.; Haffari, G.; Pan, S. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
21. Cimino, G.; Deufemia, V. SIGFRID: Unsupervised, Platform-Agnostic Interference Detection in IoT Automation Rules. *ACM Trans. Internet Things* **2025**, *6*, 1–33. [\[CrossRef\]](#)
22. Marinova, G.; Guliashki, V.; Chikov, O. MCDA approaches for automatic tool selection in a cloud-based Online-CADCOM platform. In Proceedings of the 17th International Conference on Telecommunications (ConTEL), Graz, Austria, 11–13 July 2023; pp. 1–4. [\[CrossRef\]](#)
23. Bommasani, R.; Liang, P.; Lee, T. Holistic evaluation of language models. *Ann. N. Y. Acad. Sci.* **2023**, *1525*, 140–146. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.P.; et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 46595–46623.
25. Maekawa, S.; Hassell, J.; Pezeshkpour, P.; Mitchell, T.; Hruschka, E. Towards Reliable Benchmarking: A Contamination Free, Controllable Evaluation Framework for Multi-step LLM Function Calling. In Proceedings of the The Fourteenth International Conference on Learning Representations (ICLR), Rio de Janeiro, Brazil, 23–27 April 2026.
26. Menxhiqi, L.; Marinova, G. Knowledge base assisting PCB design tool selection and combination in Online-CADCOM platform. In Proceedings of the International Conference on Information Technologies and Information Society (ITIS), Ljubljana, Slovenia, 9–10 November 2023; pp. 174–181.
27. Mala, C.S.; Gezici, G.; Giannotti, F. Hybrid Retrieval for Hallucination Mitigation in Large Language Models: A Comparative Analysis. *arXiv* **2025**, arXiv:2504.05324.
28. Geng, S.; Cooper, H.; Moskal, M.; Jenkins, S.; Berman, J.; Ranchin, N.; West, R.; Horvitz, E.; Nori, H. JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models. *arXiv* **2025**, arXiv:2501.10868.
29. Cliff, N. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychol. Bull.* **1993**, *114*, 494–509. [\[CrossRef\]](#)
30. Tam, Z.R.; Wu, C.K.; Tsai, Y.L.; Lin, C.Y.; Lee, H.Y.; Chen, Y.N. Let Me Speak Freely? A Study on the Impact of Format Restrictions on Large Language Model Performance. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, Miami, FL, USA, 12–16 November 2024; pp. 1218–1236. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.