Who Hallucinates Tools, How Often, and What Fixes It?

S Akash [1] Shine Gupta [2]

Abstract

Tool-name hallucination is a known failure mode, but who does it, how often, and what fixes it? We define a per-call Tool Fabrication Rate (TFR) and measure it on BFCL multi-turn. Commercial frontiers are silent. Across 6,000+ opaque-baseline calls (five Anthropic 4.x versions at 0/2,592, $\leq 0.14\%$, and the OpenAI gpt-4.x/4o/gpt-5 tiers at 0/2,117 and 0/1,311) we log zero fabrications. Open-weight families are not: Qwen3, Qwen2.5, and Llama fabricate at up to **10.4**%, and 70% of those calls name execution-ready state-changing actions such as fabricated place_order and cancel_booking calls, so under a permissive executor the failure is an unintended action, not a failed lookup. We propose Registry-Visible Reprompting (RVR), a training-free middleware that, on a registry miss, returns one re-prompt wrapping a structured tool_not_found envelope; it removes all twenty pooled Qwen3 fabrications (Fisher's exact, $p = 3.5\times10^{-5}$ on 20/1,417 vs. 0/945). A powered ablation locates the fix: a wrong registry list recovers as well as the real one (Fisher $p = 1.0$), so recovery is driven by the rejection's format, not the registry's content. We release the harness, traces, and per-call labels.

1. Introduction

An agent calls search_flights. The registry contains find_flights. The call fails. Practitioners hit this mismatch in production with Claude, Gemini, and Grok (Czapla, 2026), and what stings is that the model knew the task and picked something plausible. It just got the name wrong. Who actually does this, how often, and what fixes it? We answer with a per-call diagnostic, a cross-vendor audit, and a training-free fix that clears every fabrication we logged.

Existing benchmarks measure related quantities at the wrong grain, per-task or per-query and folded into composite scores (Li et al., 2023; Huang et al., 2024; Xu et al., 2024; Section 2). Read together, the failure looks architecture-agnostic and uniform. We don't think it is. At the per-call grain the split is sharp: commercial frontiers (Anthropic 4.x, OpenAI gpt-4.1/4o) sit at 0% TFR, while open-weight Qwen3 fabricates at a non-zero, non-monotone rate. Registry-Visible Reprompting (RVR), a single structured re-prompt on a registry miss, removes all twenty pooled Qwen3 fabrications in the matched per-tier subset.

We define the per-call Tool Fabrication Rate (TFR) and run it over two populations on BFCL multi-turn: the Anthropic 4.x family at five versions spanning May 2025 to April 2026, and Qwen3-Instruct at six sizes (0.6B–32B) in 4-bit MLX on consumer hardware. Both see three BFCL splits plus a four-distractor probe with the real target removed and a synthetic look-alike in its slot. BFCL is strictly typed, so this isn't production, but it is the closest we could get to Czapla's setting on a benchmark. The Qwen3 size sweep (Table 1, plotted in Figure 1) we report descriptively, not as a fitted scaling law.

Why a per-call existence rate matters: a fabricated name is not just a failed call, it is an attack surface. Spracklen et al. (2025) audited package hallucinations in code-generating LLMs and found open-source models invent non-existent dependencies at $\sim 21.7\%$ against $\sim 5.2\%$ for commercial ones, an open-vs-closed gap directionally consistent with (though an order of magnitude larger than, and in a different domain from) the Qwen-vs-Anthropic split we report, and one that an adversary can turn into "slopsquatting" by registering the hallucinated name and waiting for the model to reach for it. Tool registries are the same shape of target. We make no exploitation claim, and measure existence hallucination at the per-call grain because that is the unit at which such a risk would accrue.

[1]Indian Institute of Technology Patna, India [2]Turing (turing.com). Correspondence to: S Akash <2201ee54_s akash@iitp.ac.in>, Shine Gupta <guptashine5002@gmail .com>.

Proceedings of the *43rd* International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

The two families diverge. Across 2,592 Anthropic calls under C_0, zero TFR events (Clopper–Pearson 95% two-sided upper bound $\leq 0.14\%$). The five commercial OpenAI gpt-4.1 and gpt-4o tiers also log 0% TFR (0/2,117 pooled, $\leq 0.18\%$). The split designed to coax fabrication, multi_turn_miss_func, didn't bite: the agent chained five smaller in-registry calls instead of inventing a one-shot. With every Anthropic cell at $0/N$, no temporal slope is fittable; the five versions function as a stability check, not a trend.

Qwen3 does not. Pooled per-call rates climb from 0% at 0.6B, peak at **1.64**% at 14B, and drop back to 0% at 32B. We report this shape descriptively. A generative tool-name prior that strengthens with scale before a larger model's registry-grounding control catches up would produce something like it, a U-shaped capability window (Wei et al., 2022; McKenzie et al., 2023). But six points isn't a curve, and we say so. The 0% endpoints at 0.6B and 32B bracket the 1.64% peak at the same 4-bit MLX precision, which is weakly suggestive that the arc is not a pure compression artifact; and a full-precision (bf16) control confirms the floor: fabrication persists at bf16 (2/316; Section 5.6), so the arc is not a pure quantization artefact. We do not claim the endpoints settle the question.

RVR is training-free: intercept the proposed call, check the name, and on a miss re-prompt with a structured tool_not_found envelope listing what is available. Across the four Qwen3 tiers with non-zero TFR in the matched subset, RVR clears all twenty pooled fabrications: Fisher's exact one-sided on 20/1,417 vs. 0/945 gives $p = 3.5 \times 10^{-5}$. A secondary ablation hints at where the fix lives: dropping the registry list ($C_{0.7}$) and even filling the envelope with wrong names ($C_{0.8}$) both stay at zero. We do not over-read this. All three arms are zero-event, so we detect no content effect at this sample size: the $C_1 - C_{0.7}$ difference is bounded by a Newcombe 95% CI of $[-1.5, +1.4]$ pp, consistent with content buying up to $\sim$1 pp. A format-over-content reading (§6) is plausible and echoes Min et al. (Min et al., 2022), and we power it on a high-event cell (Qwen2.5, $C_0 = 6.10\%$): a wrong list recovers as well as the right one (Fisher $p = 1.0$; §6). On the Anthropic zero-event regime RVR holds at zero, but strict-pass slips (Sonnet $60 \rightarrow 53/60$; Haiku $60 \rightarrow 56/60$); a Newcombe-hybrid TOST at $N = 60$ is underpowered for a 5-pp non-inferiority margin, so we can't rule the cost in or out. We argue for tier-conditional deployment.

The Anthropic C_0 bound ($\leq 0.14\%$ at $N = 2{,}592$) sits an order of magnitude below the Qwen3 point estimates in the 0.95–1.64% band. The bound-vs-estimate asymmetry is the finding. The intra-family non-monotonic shape on Qwen3 is itself descriptive evidence aggregate metrics couldn't have surfaced. What we give a deployer is concrete: (i) a per-call diagnostic that exposes existence failures production guardrails silently absorb, (ii) a training-free fix (RVR) that clears every logged fabrication, and (iii) a hedged but actionable ablation: at this N the envelope alone suffices, so a deployer can ship the structured rejection without echoing internal tool names and lose nothing detectable. We release four artifacts: the per-call TFR diagnostic, an eleven-model audit over eleven months, RVR with per-tier validation, and an open-source harness with 153 unit tests. Total API spend $82; the Qwen3 sweep ran at zero marginal cost on the M5.

2. Related Work

Tool-using agents and benchmarks. Tool-using LLM agents are evaluated on a small, fairly stable set of suites. The Berkeley Function-Calling Leaderboard (Patil et al., 2024; 2025) started as single-turn API selection and has grown into a stateful multi-turn suite. τ-bench (Yao et al., 2024) pushes in a different direction, scoring dialog-driven retail and airline agents that must reconcile a user goal against a tool catalog. Both report end-to-end task success, which fuses tool selection with reasoning, parsing, and policy errors into one number. Trace-level harnesses (Trivedi et al., 2024; Yao et al., 2023) expose more, yet still report pass-rate as the headline. This is the wrong granularity for the failure we care about; TFR isolates a portable slice of it across vendors and benchmarks.

Empirical analysis of agent failures. Practitioners have informally reported that frontier models call tools that were never declared. Czapla (2026) reproduces the behavior across Claude, Gemini, and Grok, noting that "Claude 4.5 hallucinated access to a tool I hadn't given it yet …I've reproduced similar behavior with Gemini and Grok." On the academic side, Zhu et al. (2026) mine 998 bug reports from CrewAI and LangChain and find "API misuse," "API incompatibility," and "Documentation Desync" dominating the root-cause distribution, with the Self-Action lifecycle stage hit hardest. Neither line puts a number on the failure under controlled conditions. We do, formalizing the unauthorized-tool subclass as TFR and reporting it across five models.

Prior tool-hallucination metrics. Tool-call name errors have been measured under several aggregations, and the choice of denominator matters. Li et al. (2023) introduce "API Hallucination" in API-Bank

as a ground-truth-vs-prediction name-mismatch error category, with per-task error rates up to 61.4% on then-current models. The closest prior measurement to ours is Huang et al. (2024), whose MetaTool subtask 3 deliberately removes the correct tool from the registry and tracks the resulting fabrication through a per-query Correct Selection Rate. That isolates the registry-membership phenomenon, but stays at the per-query level on single-turn instructions. Xu et al. (2024) name the phenomenon directly ("tool type hallucination," where the model fabricates a tool absent from the available set), but their fix is a training-time reliability alignment that expands the action space (defer / clarify / switch), and they report it inside a composite reliability score rather than as a standalone per-call rate. We share their target failure but differ on both axes: a per-call rate, and a training-free inference-time intervention. The "does the called tool exist" question is not itself new: ToolBeHonest (Zhang et al., 2024) catalogues a non-existent tool error category qualitatively, and BFCL v4's hallucination metric (Patil et al., 2025) measures a different failure, invoking a tool when none is appropriate. What is new is the per-call operationalization across capability tiers and vendors, and the content-controlled ablation (Section 6) it makes possible: TFR extends MetaTool's premise from per-query to per-call on multi-turn agentic traces, and from controlled removal to the natural distribution of registry-membership violations. Per-call is what lets the intervention in Section 3.2 be evaluated against a content-controlled $C_{0.7}$ baseline, a contrast aggregate metrics cannot expose. Appendix Table 7 positions TFR against these prior measurements by what each one counts; the denominators and elicitation setups differ, so the numbers are not directly comparable, but the granularity gap is the point.

Constrained decoding and self-correction. A separate line constrains agents at decode time. Grammar-constrained and JSON-mode decoding (Willard & Louf, 2023; OpenAI, 2024a) prevent malformed calls. Provider-side function-calling APIs reject unknown names but surface the error as an opaque exception (OpenAI, 2024b; Anthropic, 2024). Self-correction methods such as Reflexion (Shinn et al., 2023) and self-debug (Chen et al., 2023) push the diagnosis back onto the agent's own trace. The closest neighbor in method shape is CRITIC (Gou et al., 2024), an external-feedback verify-then-correct loop that needs no training and treats the model as a black box. RVR is a registry-membership-specialized instance of that pattern. What distinguishes it is the content-controlled $C_{0.5}/C_{0.7}/C_1$ ablation, which separates whether the gain comes from retrying, from a structured-error code, or from the registry list itself. Production frameworks have started to adopt structured tool-not-found feedback; the LangChain community (limerickgds, 2025) explicitly proposes returning a structured ToolMessage on a failed call, but stops short of echoing the available registry. That gap is exactly what our $C_{0.7}/C_1$ contrast measures.

Closely related interventions. Three concurrent works deserve direct comparison. Kumar & Cohen (2026) introduce Localized In-Context Learning (L-ICL) for symbolic planners, locating the first constraint violation and injecting a minimal corrective example, lifting valid-plan rate from 59% to 89% on 8×8 gridworld. RVR is similarly minimal but operates on tool-using agents and pulls its signal from the runtime registry rather than curated examples. Zhang et al. (2026) train Qwen3-8B with Fission-GRPO, an RL recipe that resamples failed trajectories with diagnostic feedback, gaining +5.7pp error recovery on BFCL-v4 multi-turn. That is a training-time fix on the same benchmark family we target at inference time, and the two are orthogonal. Engländer et al. (2026) describe a related failure: agents discover useful signals but fail to act on them ("agents use the environment to fetch expected information, but not to revise their strategy"). RVR addresses the dual problem of acting on unobserved tools, evaluated as a paired content-controlled contrast rather than observationally.

Adjacent work. Guo et al. (2024) targets tool-server stability rather than name existence, and production tool-skill packaging (Anthropic, 2025) is orthogonal.

Concurrent work on tool hallucination (2026). Three 2026 papers bracket our setting. Yin et al. (2025) introduce SimpleToolHalluBench, whose "no tool available" and "only distractor tools" regimes mirror our target-removed probe, and report that strengthening reasoning (RL/SFT/prompting) monotonically amplifies tool hallucination. Qi (2026) find a non-monotonic effect on a different axis, chain-of-thought budget at fixed scale on Qwen2.5-1.5B function calling. Our axis is orthogonal to both: model scale at fixed (disabled) thinking, where the non-monotonicity appears across the Qwen3 size ladder rather than along a reasoning knob. Healy et al. (2026) detect tool-selection hallucination from a model's internal representations in a single forward pass; we work purely behaviorally and, rather than detect, prevent with a message-level re-prompt that needs no logit or activation access.

3. Method

3.1. Tool Fabrication Rate (TFR)

Let a tool registry $\mathcal{R}$ be a finite set of tool names with JSON schemas, exposed to the agent at episode start. The agent's message stream is parsed into a sequence of tool calls c, each carrying a name c.name, and we define

$$\mathrm{TFR}(m, b, x) = \frac{|\{c : c.\mathrm{name} \notin \mathcal{R}\}|}{|\{c : c \text{ is a parsed tool call}\}|},$$

where m, b, x index model, benchmark, and condition. The denominator is per call, not per task. Per-task aggregation conflates the rate with trace length: on long tool-using episodes, a single bad agent emitting twenty bad calls swings the headline number by an order of magnitude relative to one. Per-call observations inside a task are not assumed i.i.d.; task-level cluster-bootstrap confidence intervals are reported instead. System failures (timeout, refusal, parse-fail) are excluded from numerator and denominator alike.

3.2. Registry-Visible Reprompting (RVR)

RVR is a single-turn intervention placed between the agent and the tool executor (Appendix Figure 3). On a membership-violating call, it returns a re-prompt that names the offending call and lists the available tools:

```
def rvr(agent_msg, registry):
    proposed = parse_tool_call(agent_msg)
    if proposed.name not in registry:
        return Action.RE_PROMPT(
            f"Tool '{proposed.name}' not in registry. "
            f"Available: {sorted(registry.keys())}.")
    return Action.EXECUTE(proposed)
```

Echoing $\mathcal{R}$ back into the conversation places valid names in the model's recent context, which is what the next decoding step attends to. The intervention uses one re-prompt, a production-realistic budget. It is training-free and assumes no gradient access, so it composes with RL-based methods such as Fission-GRPO (Zhang et al., 2026). For closed-API regimes without logit access, it is the message-level analog of constrained decoding.

3.3. Conditions and the content-vs-format question

Five conditions are run on a hallucinated call. C_0 is the opaque baseline, returning the framework's default error string. $C_{0.5}$ is a naive retry with the literal text "previous call failed, try again". $C_{0.7}$ is a structured error, JSON {"error":"tool_not_found"}, with no registry attached. C_1 is RVR proper (Section 3.2): the structured envelope plus the sorted real registry. $C_{0.8}$ holds the C_1 envelope prose verbatim but swaps the listed tools for a count-matched decoy set guaranteed absent from the registry. The ladder is built so that adjacent steps isolate one factor each: $C_{0.7} : C_{0.5}$ isolates structure, $C_1 : C_{0.7}$ isolates the presence of a list, and $C_1 : C_{0.8}$ isolates whether the listed names must be correct. At scale we report the conservative $C_1 : C_0$.

We do not try to localize the effect to a specific cognitive story. An earlier draft pre-registered three mechanisms (induction-head copying, registry-as-cue, message-level rejection); the data cannot separate copying from cueing, so we drop that apparatus rather than over-read it. What the design is aimed at is a single binary: does recovery require the registry content, or only the rejection format? If a wrong list ($C_{0.8}$) re-introduces fabrications where the real list (C_1) does not, the model is reading the offered names; if neither arm fabricates, the comparison bounds the content effect rather than settling it (Section 6), and distinguishing the two requires fabrication events under the ablation. The distractor probe (Section 3.4) is retained to characterize where on the scale curve fabrications concentrate, not as a mechanism discriminator.

3.4. Distractor-Type Probe

One distractor is injected into $\mathcal{R}$ on tasks where C_0 already passes, with its type varied to characterize which kind of look-alike concentrates fabrications (Appendix Figure 4). near_name draws a string at Levenshtein distance 1 or 2 from a real target. synonym is high embedding-cosine but Levenshtein-far: semantic but not orthographic overlap. matched_random is lexically random, matched to the real target on description length and arity, controlling for the "registry-size +1" confound. unrelated is lexically and semantically disjoint, the floor case. To capture the Czapla-style runtime collision, the real target is also removed and the distractor placed in its slot: the agent expects a tool that exists, but the registry offers only a similarly-named substitute.

4. Experimental Setup

Models. We evaluate two populations. The Anthropic 4.x family (Opus 4.7, Sonnet 4/4.5/4.6, Haiku 4.5) is queried through the provider's tool-use API at temperature=0, with rolling aliases resolved at run-start to dated snapshots and pinned in the manifest. The Qwen3-Instruct family (0.6B, 1.7B, 4B, 8B, 14B, 32B) runs locally as mlx-community 4-bit MLX builds on an Apple M5 with 32 GB unified memory, macOS 15. The manifest lists SDK and dataset pins. A supplementary cross-family release covers six additional open

models: Llama-3.1-8B, Mistral-7B-v0.3, Gemma-2-9B, Qwen2.5-7B, Phi-3.5-mini, and DeepSeek-R1-Distill-Qwen-7B. Both populations decode greedy.

Benchmark. All experiments run on BFCL v4 multi-turn, using three splits: multi_turn_base (aligned registry), multi_turn_miss_func (one required tool removed), and multi_turn_long_context. The distractor probe runs on $N = 15$ multi_turn_base tasks where the opaque baseline already passes. For each task we remove the real target tool and inject one synthetic distractor in its slot. Appendix Table 8 reconciles which model saw which split: the "five Anthropic versions across eleven months" breadth lives on base only, the three-split regime sweep is two production models plus one local model, and the per-cell counts sum to the pooled 0/2,592 Anthropic figure.

Statistics. TFR is per parsed tool call. Per-cell rates get Clopper–Pearson upper bounds; non-zero cells get a BCa cluster bootstrap (10,000 resamples, clustered by task). Pooled C_0 vs. C_1 uses Fisher's exact one-sided. Distractor-type effects use Friedman with Nemenyi post-hoc, and non-inferiority uses TOST at a 5-pp margin. We apply Holm–Bonferroni across the three primary tests.

Reproducibility. A repro_manifest.json pins the BFCL v4 commit, SDK versions, dated model aliases, decoding parameters, and random seeds (master seed 42). The supplementary archive at github.com/Akasxh/tool-fabrication-rate holds the harness, 153 unit tests, the JSONL trace logs, and the scripts that regenerate every numerical claim (Appendix B).

5. Results

One finding, two regimes.

5.1. Regime tests: TFR is at or near zero across BFCL multi-turn splits

Per-call TFR for Sonnet 4.6 and Haiku 4.5 across the three BFCL multi-turn splits under C_0 is in Appendix Table 5. Every cell in the regime grid is $\mathbf{0/N}$ (C0 calls only). Among the six C0 cells in Appendix Table 5, the widest Clopper–Pearson 95% two-sided per-cell upper bound is $\leq 4.45\%$ (the 0/81 long-context Sonnet cell); pooled over the 661 C0 regime calls, $\leq 0.56\%$ (the miss_func split, designed to elicit fabrication, also stays $0/N$; both Anthropic models chain smaller in-registry calls instead, trace in Section 5.5).

5.2. Anthropic stability check across 5 model versions

The probe injects one synthetic distractor tool into a registry from which the real target tool has been removed: the agent expects a tool to exist, and the registry offers only a similarly-named distractor. Four distractor types (near-name, synonym, matched-random, unrelated) are tested on $N = 15$ multi_turn_base tasks where C_0 already passes. We ran five Anthropic 4.x versions over an 11-month window across the Opus, Sonnet, and Haiku families. On this probe every cell is $\mathbf{0/N}$ (per-version breakdown in Appendix Table 6; 95% two-sided pooled upper bound $\leq 0.19\%$). Combine the regime cells with the probe cells and the C_0 Anthropic audit pools to **0/2,592** (probe 1,931 + regime 661; per-call aggregator, deduplicated across overlapping run-ids), or $\leq 0.14\%$.

On this benchmark, Anthropic models do not invent tool names: five versions, eleven months, the same null. Two readings fit and we cannot separate them: post-training hardened against fabrication, or BFCL's redundant tool basket lets the agent route around a missing tool rather than guess. Both likely contribute. The gap with prior work shrinks once you recall that older models had no tool-use post-training and that prior benchmarks aggregated per task, not per call.

5.3. Qwen3 scaling curve: TFR is non-monotonic across 0.6B–32B

The 14B→32B drop, 1.64% to 0%, is our central intra-family result (Table 1; plotted in Figure 1). Pooled per-call TFR rises from 0% (0/75, $\leq 4.80\%$) at 0.6B, peaks at 14B ($6/366 = 1.64\%$), and collapses back to 0% (0/224, $\leq 1.63\%$) at 32B. That 32B upper bound is wide enough to overlap the 14B point estimate, so "collapse" here should be read as "not-detected at this N," not as emergent capability. The per-arm peak also shifts with scale (bold cells, Table 1): near_name at 1.7B, matched_random at 4–8B, synonym at 14B. A Friedman exact permutation test on this drift is underpowered ($\chi^2_{(3)} = 3.0$, $p = 0.46$; $n = 4$ sizes), so we read it as phenomenology, not inference.

A same-family check strengthens the picture. Qwen2.5-7B-Instruct on the same probe gives $28/459 = 6.10\%$, higher than any Qwen3 size at any arm. This is not an apples-to-apples generation comparison (different post-training and tokenizer), but it says the failure mode is not a Qwen3-specific artefact of one release. The lineage carries it.

The zero-event frontier is broad; the non-zero band is open-weight. Commercial models track the Anthropic null rather than the Qwen3 band (Table 4, ap-

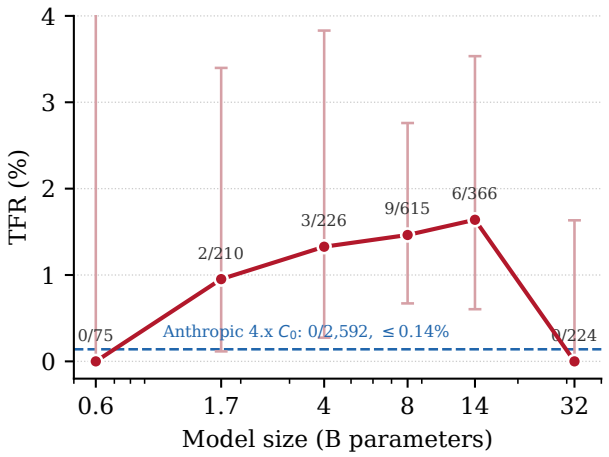

Figure 1. Qwen3-family tool-fabrication rate (TFR) under the opaque baseline C_0, four-distractor probe with the target tool removed, 0.6B–32B (per-call pooled; the same data as Table 1). Points are pooled per-size TFR; error bars are Clopper–Pearson 95% intervals. The dashed line marks the pooled Anthropic 4.x C_0 upper bound (0/2,592, $\leq 0.14\%$; Section 5.2), shown for reference. The shape is non-monotone (near-zero at 0.6B, a $\sim$1–2% band at 1.7–14B, and 0% at 32B), but with only six sizes and wide intervals (e.g. the 32B bound overlaps the 14B estimate) we read it descriptively: a single-family, single-quantization shape, not a fitted scaling law and not evidence of an emergent capability. Quantization is held at nominal 4-bit across sizes but not at fixed reconstruction error per checkpoint; we therefore do not attribute the shape to capability alone (see Section 7 and the bf16 control noted there).

pendix). On the same probe the five OpenAI gpt-4.1 and gpt-4o tiers also log 0% TFR, pooling to 0/2,117 at $\leq 0.18\%$, alongside the five Anthropic 4.x versions at 0/2,592 ($\leq 0.14\%$; Section 5.2). Every open-weight model with a non-zero rate is in an open lineage: the Qwen3 band (0.95–1.64%), Qwen2.5-7B (6.10%), and Llama-3.1-8B (1.25%). This is an open-vs-closed split, not a verified model property: the serving paths differ (structured API vs. text-parsed 4-bit MLX), so we report a serving-and-model bundle and draw no scaling law or cross-vendor ranking (Table 4; Section 7). As a confirmatory extension, the gpt-5 generation (gpt-5, gpt-5-mini, gpt-5-nano, gpt-5.5) adds 0/1,311 ($\leq$ 0.28%), all 0% TFR; we keep it out of the locked 0/2,117 OpenAI headline pool and report it only as a robustness check (Appendix B).

We treat the shape itself as the result: one would expect 14B to fabricate less than 1.7B, and it does not. The traces suggest why: at 1.7B the model rarely emits well-formed calls, leaving little surface to fabricate on; at 14B it formats fluently but does not check the registry; at 32B it does both. Six points do not fit a scaling law, but the shape is consistent: at each scale

Table 1. Qwen3 scaling curve under C_0 (target removed). Cells are events / parsed calls; bold marks the per-row peak.

Size	near	syn.	match.	unrel.	Pooled
0.6B	0/18	0/20	0/18	0/19	0/75
1.7B	2/54	0/54	0/49	0/53	2/210
4B	0/57	0/54	3/56	0/59	3/226
8B	1/219	3/134	3/129	2/133	9/615
14B	0/63	6/167	0/67	0/69	6/366
32B	0/55	0/57	0/55	0/57	0/224

the dominant failure is whichever distractor that scale is best primed to mimic.

5.4. What gets fabricated: execution-ready state-changing actions

A per-call rate understates the exposure if the invented calls are harmless stubs. They are not. We classified each of the 53 open-weight fabrication events from the C_0 target-removed pools (Qwen3 20, Qwen2.5 28, Llama 5) by the effect of the named tool: state-changing if it mutates filesystem, account, or transaction state, else read-only. The per-event labels are released with the code. Thirty-seven (70%) name state-changing actions rather than read-only queries, and all 37 carry a concrete, execution-ready argument object: no empty or malformed calls. Eleven name irreversible business transactions: place_order (4), cancel_order (3), cancel_booking (3), and create_ticket (1). Their payloads are fully specified, e.g. a fabricated place_order with quantity=150, price=25.5, order_type="buy" (a 150-share purchase), and a fabricated cancel_booking carrying a (synthetic BFCL-fixture) booking_id and card_id, each ready to execute.

In our harness these calls are caught because the tool is absent from the registry. The exposure surfaces under a permissive executor that fuzzy-matches a requested tool to the nearest registered one or auto-registers tools on first use, routing a fabricated place_order to whatever real order tool exists. We demonstrate no such exploit. We report that the fabrications are well-formed and argument-complete, and that they skew state-changing, so the failure mode is an unintended action, not a failed lookup. Worst-case per-arm rates reach $14/135 = 10.4\%$ $[5.8, 16.8]$ (Qwen2.5, synonym distractor), well above the 6.10% pooled figure. Commercial models never reach this surface: zero fabricated calls of any kind across more than 6,000 probes (Sections 5.2, 5.3).

Table 2. RVR per-tier validation on Qwen3 sizes with non-zero C_0 events. Cells are events / parsed calls. The C_0 column is the full target-removed distractor-probe pool from Table 1; C_1 is the registry-visible re-prompt arm.

Tier	C_0	C_1	Δ events
Qwen3-1.7B	2/210	0/200	2/2
Qwen3-4B	3/226	0/228	3/3
Qwen3-8B	9/615	0/258	9/9
Qwen3-14B	6/366	0/259	6/6
Pooled	20/1,417	0/945	20/20

5.5. RVR per-tier validation: 20 → 0 events across the Qwen3 sweep

We applied the registry-visible re-prompt C_1 on the four Qwen3 sizes with TFR > 0 in C_0. Across the 945 matched C_1 calls, RVR holds at 0 events against the **20** fabrications logged in the corresponding C_0 target-removed probe pool for these four tiers (Table 2). The C_0 side here is the full target-removed probe pool per tier (2/210, 3/226, 9/615, 6/366; Table 1), pooling to 20/1,417; every C_1 tier is zero-event.

The Anthropic C_1 probe (Sonnet 4.6 + Haiku 4.5) preserves 0/539 hallucinations. The strict-pass subset drops: Sonnet 60/60 → 53/60 (−11.7 pp), Haiku 60/60 → 56/60 (−6.7 pp). A Newcombe-hybrid TOST at the 5 pp margin gives $p_{\mathrm{TOST}} = 0.91$ (Sonnet) and 0.63 (Haiku). At $N = 60$ the test is underpowered ($1-\beta < 0.30$), so we can call neither non-inferiority nor inferiority. RVR stays zero-event on the Anthropic frontier; the strict-pass cost is undetermined at this N (Section 7).

The four-condition ladder on Qwen3-8B walks down a clean gradient (Appendix Figure 6). Opaque baseline 1.46% (9/615), naive retry 0.39% (1/258), structured envelope $C_{0.7}$ 0/448 ($\leq$ 0.82%), full RVR C_1 0/258. The structured envelope alone already takes TFR to zero; stacking the registry list on top buys no further detectable reduction (Newcombe-hybrid 95% CI on $C_1 - C_{0.7}$: $[-1.5, +1.4]$ pp), as we analyze in Section 6. The practical upshot: a deployer who wants prevention without echoing internal tool names can ship $C_{0.7}$ and lose nothing.

Fisher's exact on the pooled TFR > 0 subset. The C_0/C_1 denominators differ per tier, so the trials are not paired at the call level; we treat the four tiers as independent strata. We report Fisher's exact one-sided test on the pooled 20/1,417 vs. 0/945 contingency: $\mathbf{p = 3.5 \times 10^{-5}}$. Per-tier Fisher's exact one-sided p values are $\{0.26, 0.12, 0.042, 0.040\}$ for 1.7B/4B/8B/14B respectively. Each tier is individually underpowered, but the sign is consistent. The pooled contrast survives Holm–Bonferroni at $p_{\mathrm{Holm}} = 1.0 \times 10^{-4}$.

Trace inspection. On multi_turn_miss_func_24 the Anthropic agent chains five in-registry calls (pwd→ls→cd→mkdir→mv) where a one-shot mv_recursive() would have been fabricated. Across 30 near-name probe traces, agents reach for other in-registry tools rather than the removed target's or the distractor's name. RVR adds zero provider round-trips on hits, one on misses; the rendered registry is ∼100–200 tokens at ≤25 tools (under 5% of context).

5.6. Cross-family and precision controls

Two objections target the open-weight result: that the Qwen3 band is a 4-bit MLX quantization artefact, and that it reflects a single open family. We address both on the same C_0 target-removed probe (visualized in Appendix Figure 8); all intervals are Clopper–Pearson 95% two-sided.

Precision ladder. We re-ran Qwen3-8B at three precisions. Fabrication does not vanish at full precision: bf16 logs 2/316 = 0.63% $[0.08, 2.27]$, alongside 8-bit 2/435 = 0.46% $[0.06, 1.65]$ and 4-bit 9/623 = 1.44% $[0.66, 2.72]$.[1] Point estimates trend downward as precision rises, but all three intervals overlap, so we claim no precision effect. The load-bearing fact is the floor: tool fabrication persists at full precision and is therefore not a quantization artefact.

Second open family. Llama-3.1-8B-Instruct (4-bit), a lineage independent of Qwen, fabricates at 5/400 = 1.25% $[0.41, 2.89]$ on the same probe, inside the Qwen-family band. With Qwen2.5-7B-Instruct's 28/459 = 6.10% (Section 5.3), the open-weight gap appears in two distinct lineages, not one. RVR generalizes across the boundary too, though not to a clean zero. The C_1 registry-visible re-prompt cuts Llama-3.1-8B from 1.25% to 2/487 = 0.41% $[0.05, 1.48]$ (threefold) and Qwen2.5-7B from 6.10% to 1/320 = 0.31% $[0.01, 1.73]$ (roughly twentyfold). In each case the residual emissions (mkdir and cancel_order on Llama, mkdir on Qwen2.5) were caught by RVR's tool_not_found rejection layer rather than reaching the environment: both stages fire (prompt removes most, rejection net catches the rest), defense-in-depth, not a single-shot guarantee. These nulls are weaker than the matched Qwen3 sweep (0/945), as expected for single $N \approx$

[1] This 4-bit cell (9/623) is the precision-control acquisition; the scaling-sweep 8B cell (9/615, Table 1) is a separate run with the same nine fabrication events over marginally fewer parsed calls.

300–500 probes rather than a per-tier-matched design, but consistent across both families. Four further open families (Mistral-7B-v0.3, Gemma-2-9B, Phi-3.5-mini, DeepSeek-R1-Distill-7B) emitted no parseable tool calls under our harness and yield no TFR denominator (Section 7).

6. What recovers: content or format

Does recovery use the content of the echoed registry, or only the format of a structured rejection? The condition ladder speaks to this, but only bounds the effect, because every arm is zero-event.

The ablation is suggestive, not decisive. On Qwen3-8B the structured error $C_{0.7}$, a tool_not_found envelope with no registry attached, also drives fabrications to zero, matching full RVR (Section 5.5). If the model needed to read the registry to recover, removing it might cost something; within this experiment's resolution we see no such cost.

The decoy arm sharpens, but does not close, the question. $C_{0.7}$ removes the list; it does not test a list present but wrong. A registry-as-cue account predicts a decoy list (formatted, count-matched, populated with non-existent tools) fails to recover, with no valid name to copy; a format-only account predicts it recovers anyway, since "your last call was rejected; choose from this set" need not be truthful to suppress the call. The $C_{0.8}$ condition runs this contrast, holding the C_1 envelope prose verbatim and swapping only the listed names for guaranteed-absent decoys.

The decoy list logs 0/410 (CP 95% upper bound $\leq$ 0.90%; near_name 0/97, synonym 0/106, matched_random 0/107, unrelated 0/100), matching the no-list envelope ($C_{0.7}$, 0/448) and the real list (C_1, 0/258 on 8B). All three are zero-event, so we detect no content effect at this sample size: absence of evidence, not evidence of equivalence. The $C_1 - C_{0.7}$ and $C_1 - C_{0.8}$ differences are bounded by a Newcombe 95% CI of $[-1.5, +1.4]$ pp, consistent with the registry content buying up to $\sim$1 pp of recovery, not nothing. The data are thus compatible with format-over-content, the model responding to the message-level rejection rather than the valid names offered, but we do not treat zero-vs-zero as proof, and a registry-as-cue account is not ruled out within the $\pm$1.5 pp band.

We settle it on a high-event cell. Qwen2.5-7B opens at $C_0 = 6.10\%$ (28/459), and the recovery ladder (Appendix Figure 7) walks it down to 0.31% at C_1. All three envelope arms collapse the rate far below C_0 (Fisher, $p < 10^{-3}$) and none differs from another (pairwise Fisher $p > 0.37$); the wrong-list arm matches the real-list arm exactly (Fisher $p = 1.0$). With events to recover, the content effect left open by the $\pm$1.5 pp Qwen3-8B bound does not appear: a powered null where a wrong registry (or none) recovers as well as the right one. A bare retry ($C_{0.5}$) helps but less, leaving a residual $\sim 5\times$ the structured arms ($p = 0.09$ vs. C_1).

What this does not claim. Format-sufficiency is about recovery, not task success: on TFR alone RVR matches any structured rejection (a hard reject also yields 0), but RVR is recovery-preserving, returning the live registry so the agent re-selects a valid tool rather than aborting. The value of the true registry thus shows up in task-completion rate, not TFR. And the result holds only on tiers with C_0 fabrications to suppress (1.7–14B); 0.6B and 32B carry no signal to recover.

7. Discussion and Limitations

A per-tier operator recommendation is condensed in Table 3 (appendix); the rest of this section states what the audit does and does not license.

We cannot disentangle the Anthropic–Qwen3 gap: parameter count, training recipe, and provider guardrails co-vary, and commercial-vs-open bundles serving path with weights (Section 5.3). Quantization is no confound: the precision ladder (Section 5.6) shows fabrication persists at full precision. The Anthropic null is also indistinguishable from BFCL v4 contamination (v4 predates every model tested; we ran no membership-inference control). RVR's 20 $\rightarrow$ 0 is quantization-invariant (C_0 and C_1 share weights); on Anthropic it is a call-path no-op yet strict-pass drops (Section 5.5), and our TOST is underpowered at $N = 60$, so we recommend $C_{0.7}$/RVR only on tiers with audited TFR>0, paired with executor-side allow-listing. Scope: the probe tests registry-shape, not runtime-shape; TFR ignores latency, abstention, and pass-rate; and we claim no scaling law. With only two non-Qwen3 lineages (Llama, Qwen2.5), we claim no general cross-family transfer. A second benchmark bounds the reach: on τ-bench (simulated user) the commercial null holds (0/2,244 over gpt-4.1, Sonnet 4.6, Haiku 4.5, $\leq$0.16%), but our probe drew no open-weight fabrication either: Qwen3-8B and 14B made 191 valid in-registry calls at zero ($\leq$1.91%) while completing none of the 24 probed τ tasks. The null may then reflect too little engagement, not robustness, so we read τ as scoping the open-weight gap to BFCL-style target-removed registries, not as a negative result.

References

Anthropic. Tool use with Claude. Anthropic API Documentation, 2024. URL https://docs.anthropic.com/en/docs/agents-and-tools/tool-use/overview.

Anthropic. Equipping agents for the real world with Agent Skills. Anthropic engineering blog, October 2025. URL https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills.

Chen, X., Lin, M., Schärli, N., and Zhou, D. Teaching large language models to self-debug, 2023. URL https://arxiv.org/abs/2304.05128.

Czapla, P. The unauthorized tool call problem. Answer.AI blog, February 2026. URL https://www.answer.ai/posts/2026-01-20-toolcalling.html. Accessed 2026-04-29.

Engländer, L., Althammer, S., Üstün, A., Gallé, M., and Sherborne, T. Agents explore but agents ignore: LLMs lack environmental curiosity, 2026. URL https://arxiv.org/abs/2604.17609.

Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., and Chen, W. CRITIC: Large language models can self-correct with tool-interactive critiquing. In ICLR, 2024. URL https://arxiv.org/abs/2305.11738.

Guo, Z., Cheng, S., Wang, H., Liang, S., Qin, Y., Li, P., Liu, Z., Sun, M., and Liu, Y. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models, 2024. URL https://arxiv.org/abs/2403.07714.

Healy, K., Srinivasan, B., Madathil, V., and Wu, J. Internal representations as indicators of hallucinations in agent tool selection, 2026. URL https://arxiv.org/abs/2601.05214.

Huang, Y., Shi, J., Li, Y., Fan, C., Wu, S., Zhang, Q., Liu, Y., Zhou, P., Wan, Y., Gong, N. Z., and Sun, L. MetaTool benchmark for large language models: Deciding whether to use tools and which to use. In ICLR, 2024. URL https://arxiv.org/abs/2310.03128.

Kumar, A. and Cohen, W. W. Localizing and correcting errors for LLM-based planners, 2026. URL https://arxiv.org/abs/2602.00276.

Li, M., Zhao, Y., Yu, B., Song, F., Li, H., Yu, H., Li, Z., Huang, F., and Li, Y. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3102–3116, 2023. URL https://aclanthology.org/2023.emnlp-main.187/.

limerickgds. Allow graceful handling of 'tool not found' errors in LangGraph ToolNode to enable agent self-correction. LangChain Forum thread #2427, December 2025. URL https://forum.langchain.com/t/allow-graceful-handling-of-tool-not-found-errors-in-langgraph-toolnode-to-enable-agent-self-correction/2427. Production-engineer-proposed structured-error pattern (our $C_{0.7}$); explicitly stops short of listing available tools.

McKenzie, I. R., Lyzhov, A., Pieler, M., Parrish, A., Mueller, A., Prabhu, A., McLean, E., Kirtland, A., Ross, A., Liu, A., et al. Inverse scaling: When bigger isn't better. Transactions on Machine Learning Research, 2023. URL https://arxiv.org/abs/2306.09479.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022. URL https://arxiv.org/abs/2202.12837.

OpenAI. Structured outputs. OpenAI Platform documentation, 2024a. URL https://platform.openai.com/docs/guides/structured-outputs.

OpenAI. Function calling. OpenAI Platform documentation, 2024b. URL https://platform.openai.com/docs/guides/function-calling.

Patil, S. G., Mao, H., Yan, F., Ji, C. C.-J., Stoica, I., and Gonzalez, J. E. Berkeley function calling leaderboard, 2024. URL https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html.

Patil, S. G., Mao, H., Ji, C. C.-J., Yan, F., Suresh, V., Stoica, I., and Gonzalez, J. E. Berkeley function calling leaderboard (BFCL) V4, 2025. URL https://gorilla.cs.berkeley.edu/leaderboard.html. Accessed 2026-04-29.

Qi, X. Brief is better: Non-monotonic chain-of-thought budget effects in function-calling language agents, 2026. URL https://arxiv.org/abs/2604.02155.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Spracklen, J., Wijewickrama, R., Sakib, A. H. M. N., Maiti, A., Viswanath, B., and Jadliwala, M. We have a package for you! a comprehensive analysis of package hallucinations by code generating LLMs. In Proceedings of the 34th USENIX Security Symposium (USENIX Security), 2025. URL https://arxiv.org/abs/2406.10279.

Trivedi, H., Khot, T., Hartmann, M., Manku, R., Dong, V., Li, E., Gupta, S., Sabharwal, A., and Balasubramanian, N. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), 2024. URL https://arxiv.org/abs/2407.18901.

Wei, J., Kim, N., Tay, Y., and Le, Q. V. Inverse scaling can become U-shaped, 2022. URL https://arxiv.org/abs/2211.02011.

Willard, B. T. and Louf, R. Efficient guided generation for large language models. arXiv preprint arXiv:2307.09702, 2023. URL https://arxiv.org/abs/2307.09702.

Xu, H., Zhu, Z., Pan, L., Wang, Z., Zhu, S., Ma, D., Cao, R., Chen, L., and Yu, K. Reducing tool hallucination via reliability alignment, 2024. URL https://arxiv.org/abs/2412.04141.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Yin, C., Sha, Z., Cui, S., Meng, C., and Li, Z. The reasoning trap: How enhancing LLM reasoning amplifies tool hallucination, 2025. URL https://arxiv.org/abs/2510.22977.

Zhang, Y., Chen, J., Wang, J., Liu, Y., Yang, C., Shi, C., Zhu, X., Lin, Z., Wan, H., Yang, Y., Sakai, T., Feng, T., and Yamana, H. ToolBeHonest: A multi-level hallucination diagnostic benchmark for tool-augmented large language models, 2024. URL https://arxiv.org/abs/2406.20015.

Zhang, Z., Zhao, F., Wang, R., Wang, Z., Liang, B., Wang, J., Hu, Y., Cao, S., and Wong, K.-F. Robust tool use via fission-GRPO: Learning to recover from execution errors, 2026. URL https://arxiv.org/abs/2601.15625.

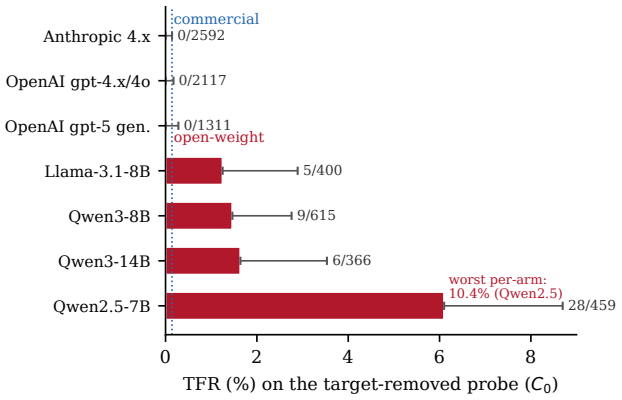

Figure 2. Commercial versus open-weight TFR on the target-removed C_0 probe. Five Anthropic 4.x versions (0/2,592), the OpenAI gpt-4.x/4o tiers (0/2,117), and the gpt-5 generation (0/1,311) log zero fabrications; open-weight lineages fabricate at 1.25 to 6.10%, with the worst per-arm cell at 10.4% (Qwen2.5, synonym distractor). Error bars are Clopper–Pearson 95% upper bounds.

Zhu, X., Wu, J., Zhang, X., Li, T., Mu, Y., Zhai, J., Shen, C., Fang, C., and Liu, Y. An empirical study of bugs in modern LLM agent frameworks, 2026. URL https://arxiv.org/abs/2602.21806.

LLM/Agent Usage Disclosure

Experimental subjects. The paper evaluates eleven LLMs as subjects of measurement: five Anthropic 4.x versions (Opus 4.7, Sonnet 4/4.5/4.6, Haiku 4.5) via Anthropic's tool-use API, and six Qwen3-Instruct sizes (0.6B–32B) as 4-bit MLX builds on a 32 GB Apple Silicon machine. All are treated as black-box targets; version snapshots, dated aliases, and decoding configurations are pinned in the supplementary manifest.

Drafting, code, and statistics. The paper text and the harness (adapters, intervention conditions, statistical scripts, trace logger, cost meter) were drafted with LLM-based assistance under author supervision. All claims, numerical results, methodological decisions, and statistical tests (Fisher's exact, Newcombe-hybrid TOST, Holm–Bonferroni, BCa cluster-bootstrap, Friedman+Nemenyi) are reproducible from the released harness and its 153 unit tests; no p-value appears without a code path that regenerates it. Contributions, experimental design, scope claims, and limitations were authored under direct human supervision.

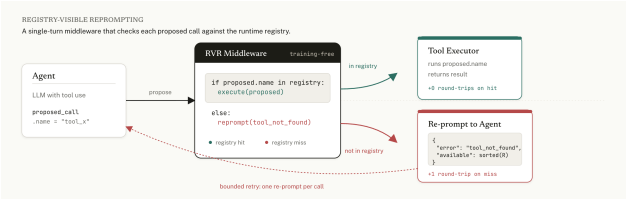

Figure 3. RVR middleware. An $O(1)$ membership check runs in-process on each proposed call. Hits execute. Misses trigger one re-prompt with a structured tool_not_found envelope (and the sorted registry under C_1). The no-violation path adds zero provider round-trips; the violation path adds one, bounded to a single retry.

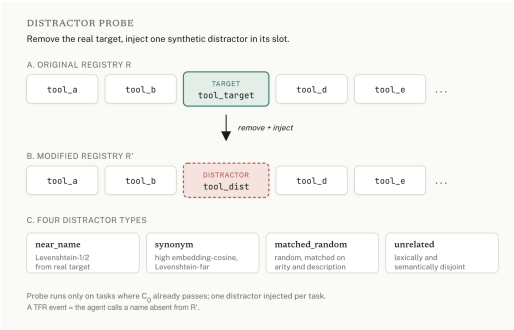

Figure 4. The distractor probe. Starting from a registry where the opaque baseline already passes, the real target tool is removed and one synthetic distractor is injected in its slot. The four arms vary the similarity profile of the distractor.

A. Supplementary Figures and Tables

B. Pre-Registration and Reproducibility

The pre-registered protocol is locked in git commit c391017, dated before any data was collected. Three deviations from that protocol are worth naming. Probe N was reduced from 100 to 15 per cell to fund breadth across five Anthropic versions and six Qwen3 sizes. The OpenAI gpt-4.1 and gpt-4o tiers were added to the vendor scope after the pre-registration (five commercial models, 0/2,117 pooled C0 calls, all 0% TFR). The gpt-5 generation (gpt-5, gpt-5-mini, gpt-5-nano, gpt-5.5) has since completed and extends the commercial null: 0/1,311 pooled, all 0% TFR ($\leq 0.28\%$). We keep these out of the locked 0/2,117 headline pool and report them as a confirmatory extension. $C_{0.7}$ was run at scale on Qwen3-8B only, not on Anthropic. Holm–Bonferroni correction across the three primary tests (Fisher's exact, Friedman, TOST): only Fisher's clears $\alpha = 0.05$ at $p_{\text{Holm}} = 1.0 \times 10^{-4}$.

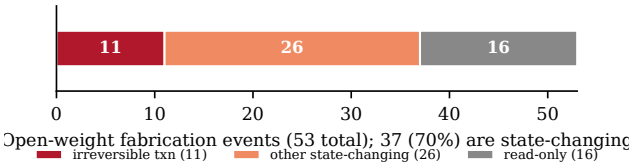

Figure 5. What gets fabricated. Of the 53 open-weight C_0 fabrication events, 37 (70%) name state-changing actions, 11 of them irreversible business transactions (place_order, cancel_order, cancel_booking, create_ticket); the remaining 16 are read-only queries.

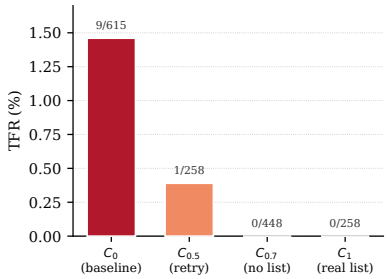

Figure 6. Qwen3-8B 4-condition ablation. The structured-error envelope ($C_{0.7}$) already takes TFR to zero; adding the registry list (C_1) yields no further detectable reduction (both arms zero-event; see §6 for why this bounds rather than proves a format-over-content account).

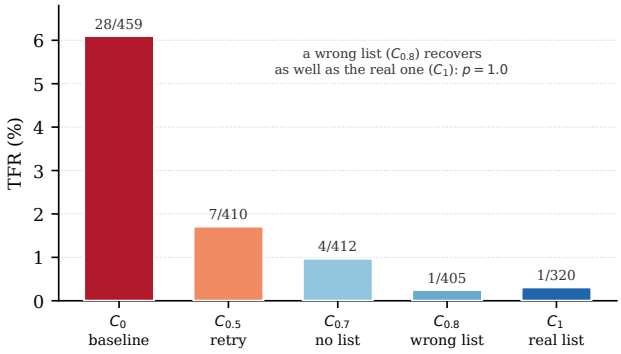

Figure 7. Qwen2.5 ablation ladder. A bare retry ($C_{0.5}$) only halves the 6.10% baseline, while the structured tool_not_found envelope cuts it about 20×; a wrong registry list ($C_{0.8}$) recovers as well as the real one (C_1; Fisher $p = 1.0$), so recovery is format-driven.

The supplementary repository at github.com/Akasxh/tool-fabrication-rate contains the full harness (153 unit tests), the JSONL trace logs that produce every number in the paper, the aggregator and statistical scripts, the reproducibility manifest with dated model aliases, the full deviation log, the JSONL trace schema, the headline-number derivation rule, and a registry-size pilot. Running python scripts/aggregate_all.py on the released traces reproduces every numerical claim. Distractor-injection RNG uses a SHA-256-derived stable hash, so reproduction does

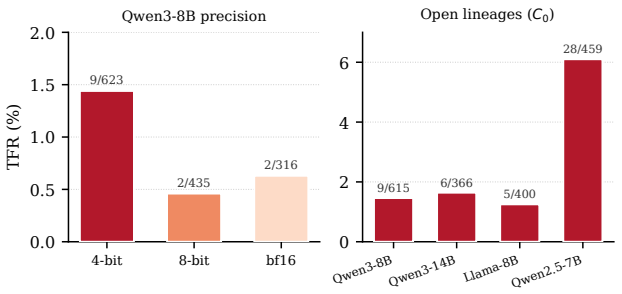

Figure 8. Controls. Left: the precision ladder (Qwen3-8B at 4-bit/8-bit/bf16) shows fabrication persists at full precision, so it is not a quantization artefact. Right: the C_0 fabrication rate across open lineages.

not depend on PYTHONHASHSEED; MLX 4-bit local inference is not bitwise-deterministic across runs but per-task tool-call decisions are stable to within ± 1 token.

C. Broader Impacts and Responsible Disclosure

This work is diagnostic and defensive: it measures a failure mode and ships a training-free mitigation. Every fabrication reported here was elicited in a typed, offline harness with no real executor and no side effects. We demonstrate no exploit and trigger no real-world action. The underlying failure class (agents emitting calls to non-existent tools) is already documented in prior tool-hallucination work (Li et al., 2023; Huang et al., 2024), and the open-weight models we name are publicly released, so we judge the net effect of releasing a per-call measurement plus a cheap fix to be defensive rather than uplifting. The one concrete risk we surface (a permissive executor that fuzzy-matches or auto-registers a fabricated state-changing call) is mitigated by the executor-side allow-listing we recommend alongside RVR; RVR reduces the fabrication load reaching that allow-list rather than replacing it. Code and the per-event severity labels are released at github.com/Akasxh/tool-fabrication-rate to support reproduction and defensive tooling.

Table 3. Deployer decision table. Recommendation by model tier for an operator running tool-calling agents on BFCL-style registries. "Audited TFR" is the per-call tool-fabrication rate measured under C_0 (no intervention); intervals are Clopper–Pearson 95% two-sided. The intervention column lists RVR (C_1, registry-visible re-prompt) and its prompt-only variant $C_{0.7}$ (structured-error envelope, no internal tool names echoed). RVR reduces measured fabrication and adds an executor-side rejection layer; it is one layer of defense-in-depth, not a guarantee (Section 7). We recommend the intervention only on tiers with audited TFR > 0, always paired with executor allow-listing.

Tier	Audited TFR (evidence)	Middleware recommendation	Rationale / caveat
Commercial frontier (Anthropic, OpenAI)	≈0, audited. Anthropic 0/2,592 (≤ 0.14%); OpenAI 0/2,117 (≤0.18%) (§5.2, §5.3)	No fabrication middleware required. Keep executor allow-listing as standard hygiene.	Five Anthropic versions over 11 months and five OpenAI tiers are zero-event on the target-removed probe. On Anthropic, RVR is a call-path no-op yet strict-pass can drop at $N = 60$ (§5.5): adding it buys no measured safety here and may cost utility. Null is not "proven safe": it is bounded by the audit, and is indistinguishable from BFCL-v4 contamination (§7). Re-audit on model updates.
Audited open-weight (TFR > 0: Qwen3 1.7–14B, Llama-3.1-8B, Qwen2.5-7B)	0.4–6.1%, measured. Qwen3 pooled 20/1,417; Llama-3.1-8B 5/400 = 1.25%; Qwen2.5-7B 28/459 = 6.10% (§5.3, §5.6)	Deploy $C_{0.7}$ or RVR (C_1) plus executor allow-listing. Prefer $C_{0.7}$ if internal tool names must stay private.	On the matched Qwen3 sweep RVR drives 20 → 0 (Fisher one-sided $p = 3.5 \times 10^{-5}$; §5.5); $C_{0.7}$ alone already reaches zero on Qwen3-8B at no detectable cost (Fig. 6). Cross-family it reduces not eliminates: Llama 1.25% → 0.41%, with both residuals caught by the rejection layer, not the prompt (§5.6). Hence the executor allow-list is load-bearing, not optional. Not a quantization artefact: persists at bf16 (§5.6).
Unaudited open-weight (any model without a per-call TFR audit)	Unknown; assume TFR > 0. Gap appears in two distinct open lineages (Qwen, Llama), so treat open weights as fabrication-prone by default (§5.6)	Audit first (run the C_0 target-removed probe), then deploy $C_{0.7}$/RVR plus executor allow-listing regardless of the audit.	We observe non-monotonic TFR across 0.6–32B (§5.3): a larger model is not safer by default, so capability is no substitute for an audit. With wide intervals at small N, a zero in the audit bounds but does not prove safety. The executor allow-list is the guarantee; RVR/$C_{0.7}$ reduces the load reaching it.

Table 4. Tool-fabrication rate (TFR) on the target-removed distractor probe (C_0): the commercial frontier vs. the open-weight band. Cells are events / parsed calls; UB is the two-sided Clopper–Pearson 95% upper bound. This is an open-vs-closed serving-and-model bundle, not a verified model property (commercial tiers are API-served, open-weight tiers are text-parsed 4-bit MLX builds), so we draw no scaling law or cross-vendor ranking from it. The gpt-5 generation is a confirmatory extension, kept out of the locked 0/2,117 OpenAI headline pool.

Tier	Events / N	TFR (UB)
Commercial (closed, API-served)		
Anthropic 4.x (5 ver., pooled)	0/2,592	0% ($\leq$0.14%)
OpenAI gpt-4.1/4o (5 tiers)	0/2,117	0% ($\leq$0.18%)
OpenAI gpt-5 gen. (4)	0/1,311	0% ($\leq$0.28%)
Open-weight (4-bit MLX, text-parsed)		
Qwen3 (peak, 14B)	6/366	1.64%
Qwen2.5-7B	28/459	6.10%
Llama-3.1-8B	5/400	1.25%

Table 5. Regime-test TFR by model and BFCL split under C_0. Cells are events / parsed calls.

Model	base	miss_func	long_ctx
Sonnet 4.6	0/116	0/128	0/81
Haiku 4.5	0/109	0/120	0/107

Table 6. Anthropic 4.x distractor probe with target removed (C_0). Pooled C_0 over regime + probe is 0/2,592, $\leq$0.14%; the C_1 subset adds 0/659 at $\leq$0.56%.

Model	N	TFR
Opus 4.7	293	0
Sonnet 4.6	561	0
Sonnet 4.5	285	0
Sonnet 4	253	0
Haiku 4.5	539	0
Pooled probe	1,931	0

Table 7. Prior tool-hallucination measurements versus TFR. Granularity is the denominator each metric divides by; existence-clean? asks whether the metric isolates registry-membership violations (a called name is absent from the offered set) from relevance, argument, and timing errors. Not apples-to-apples: the headline numbers come from different denominators, different model cohorts, and different elicitation setups (single-turn instructions vs. multi-turn agentic traces; target-removed probes vs. natural distributions). They are not directly comparable and we do not claim a ranking; the table positions what each metric counts, not which model is best. TFR is the only per-call rate and the only one that scores existence on multi-turn traces with a content-controlled ablation.

Work	Granularity	Setting	Headline number	Existence-clean?
API-Bank (Li et al., 2023)	per-task	single-turn API calls	name-mismatch up to 61.4%	partial
MetaTool (Huang et al., 2024)	per-query	single-turn, target removed	Correct Selection Rate	yes
ToolBeHonest (Zhang et al., 2024)	per-task	multi-level diagnostic	multi-level (no single rate)	partial
Reasoning-Trap (Yin et al., 2025)	per-task	no-tool / distractor-only	monotonic in reasoning	yes
Reliability align. (Xu et al., 2024)	composite	training-time, expanded actions	inside reliability score	partial
Ours (TFR)	per-call	multi-turn agentic traces	**0**%–**1.64**%	yes

Table 8. Model × BFCL-split coverage. Cells are no-intervention (C0) calls; a glyph marks the experiment design that produced them. ●: the five-distractor probe grid (target removed, per-call existence scored). ○: the regime cell from the main grid. The "five Anthropic versions across eleven months" claim holds on base only; the three-split regime sweep is two production models (Haiku 4.5, Sonnet 4.6) plus one local model (Qwen3-8B, base). Anthropic hallucinated = 0 in every cell; the per-call TFR signal is the Qwen3 size sweep on base. Cells sum to the pooled Anthropic C_0 total 2,592.

	BFCL split (C0 calls)		
Model (version)	base	long-ctx	miss-func
Anthropic 4.x			
Sonnet 4.0 (05/2025)	● 253	–	–
Sonnet 4.5 (09/2025)	● 285	–	–
Sonnet 4.6	● 677	○ 81	○ 128
Opus 4.7	● 293	–	–
Haiku 4.5	● 648	○ 107	○ 120
Qwen3 size sweep (4-bit)			
0.6B	● 75	–	–
1.7B	● 210	–	–
4B	● 226	–	–
8B	● 615	○ 108 (base regime)	
14B	● 366	–	–
32B	● 224	–	–