THE NO-WITNESS RULE: SELECTIVE PROVENANCE CHECKS FOR TOOL-CALL ARGUMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using agents can supply identifiers that never appeared in their observations, and well-formed calls can execute with those unsupported values. Candidate-constrained decoding addresses this by restricting which values can be emitted, but an empty candidate set is a poor intervention signal in our controlled comparisons: permissive typing never triggers recovery, while strict typing reduces task success to 8.5% from 33.0%. We introduce NOWIT, a selective provenance policy for referential tool-call arguments. The agent drafts a call freely; NOWIT checks each eligible argument against a monotone witness set built from user turns, tool returns and documentation. It executes passing drafts unchanged and replaces a flagged draft with a constrained read-only lookup or clarification. Recovery is invoked on 8.43% of call steps in the primary run. On BFCL v4 with a frozen 8B model, unwitnessed referential emission falls from 8.65% to 2.53% (paired difference -6.12 pp, $[-8.44, -4.09]$), and unwitnessed state-mutating emissions fall from 67 to 22. The reduction repeats across three seeds, at a second site, and at 14B, where the rate falls from 13.03% to 2.39%. Task-score differences satisfy the pre-stated $\pm$7-point equivalence margin across seeds. On τ^2-BENCH retail, refusals on unwitnessed emissions fall from 38 to 0. Together, the controlled trigger comparison and replicated reductions support checking the proposed value as a practical intervention rule for tool-using agents.

1 INTRODUCTION

An agent retrieving a booking, updating a ticket or reading a file must supply a reference to an existing entity. A plausible identifier is insufficient: the call needs an appropriate value from the task or environment. Language models can generate such identifiers without having observed them, while a tool schema checks their format without establishing their source. Once the call executes, an unsupported argument can become an unsupported action.

Prior work addresses this problem through clarification and uncertainty. Agents can be prompted to ask when instructions are underspecified (Wang et al., 2025); structured uncertainty can trigger clarification (Suri et al., 2026), and post-training can teach abstention (Zhou et al., 2026). These approaches encourage recovery, but the decision remains with the model: schemas that demand a field can still elicit invented values despite an available escape (Usman, 2026). Our prompt control names the failure and both remedies, yet changes the unwitnessed rate by 0.13 percentage points (Appendix C.2). At site B, the baseline rate is 9.00% for 8B and 13.03% for the more task-accurate 14B model. These results motivate checking provenance at execution time.

Candidate-constrained decoding intervenes directly by restricting a reference position to available identifiers. Existing formulations also specify a fallback when the candidate set is empty, such as relaxing the grammar or refusing (Zhang et al., 2026a; Miyamoto et al., 2026). For multi-turn tool use, however, non-emptiness and support for the proposed value are different properties. A permissive candidate extractor can fill a slot with ordinary conversation spans even when none supports the draft. Tightening the extractor can instead reject a call whose proposed arguments are already witnessed. We isolate this choice by holding drafting, recovery and enforcement fixed: STRICT invokes recovery

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

on 68.21% of call steps and scores 8.5%, compared with 33.0% for PERMISSIVE, which never intervenes.

Candidate coverage supplies a second reason to separate these decisions. On a diagnostic census, 77 of 763 emissions have no witnessed episode-reference value despite a non-empty candidate set. Another 97 omit a reference value that is witnessed but missed by the extractor. This reveals a gap between candidate membership and exposure. A synthetic task with a known unique target then isolates the consequence: excluding that target prevents a successful lookup, while the agent's stopping behaviour depends on the protocol (Section 4.4).

NoWit places the intervention test on the drafted value (Figure 1). It maintains a monotone witness set W_t from user turns, executed tool returns, documentation and schema enumerations. The model drafts freely, and calls whose eligible referential arguments pass the witness check execute unchanged. A flagged draft is withheld and re-decoded into a read-only lookup or clarification. This separates what the agent has observed, when recovery is invoked, and which recovery actions are available. Proposition 1 establishes that an abstention branch remains derivable under the mask.

Our experiments test the trigger first, then the complete policy. With the recovery path fixed, the typed empty-set controls expose the task costs of unnecessary interventions. NoWit reduces unwitnessed emission from 8.65% to 2.53% in the primary 8B pair while invoking recovery on 8.43% of call steps. The direction repeats across three seeds, a second site and 14B. On retail τ^2-bench, which validates identifiers during execution, refusals on unwitnessed emissions fall from 38 to 0. Task scores, rollout costs and source sensitivity are reported alongside these provenance outcomes (Section 4).

This paper makes three contributions.

1. A controlled comparison of intervention triggers. With drafting, recovery and enforcement fixed, the tested empty-set gates range from no intervention to frequent intervention with substantial task loss (Section 4.2).

2. A separation of candidate coverage from stopping. Reference-coverage analysis identifies values omitted by a non-empty candidate set; synthetic trials show that answer exclusion and failure to stop are distinct outcomes (Section 4.4).

3. NoWit, a value-level provenance policy. It reduces unwitnessed emissions and writes across model sizes, sites and seeds, with task effects inside the pre-stated ± 7-point equivalence margin and explicit accounting of recovery cost (Sections 4.3 and 4.5).

2 Related Work

Restricting generation to a formal language at decode time is mature engineering (Dong et al., 2025; Li et al., 2026b; Guidance AI contributors, 2026; Willard & Louf, 2023; Xu & Bouyarmane, 2026), with foundations in PICARD's incremental rejection of inadmissible tokens (Scholak et al., 2021), Synchromesh's completion-engine constraints on syntax, scope and types (Poesia et al., 2022), and input-dependent grammars for structured NLP (Geng et al., 2023). We use that machinery and study the intervention policy layered over it.

The formulation nearest to ours indexes the grammar by the environment, so a reference position admits exactly the names available at that point, and proves that no out-of-scope identifier can be emitted (Zhang et al., 2026a). Two things separate us. Its candidate set is populated from a program's own declarations rather than from a multi-turn trajectory whose witnesses accumulate through user turns and tool returns, and when its slot query returns no candidates the grammar is relaxed to a coarser fragment, so the model writes more freely at precisely the moment nothing licenses what it writes. NoWit inverts that branch. Giving the empty set distinguished treatment is not new either, since in column-level access control for text-to-SQL it drives a terminal refusal (Miyamoto et al., 2026), we instead compare candidate-emptiness and value-level gates under shared recovery (Section 4.2).

A growing literature measures what masking takes away, and it is the reason Proposition 1 is necessary rather than decorative. Structured-output constraints can suppress tool invocation outright (Li et al., 2026a), a grammar component can repair form while degrading the decision (Lee, 2026), the tradeoff is sharpest at small scale (Ray, 2026), and part of the loss belongs to the prompt rather than the

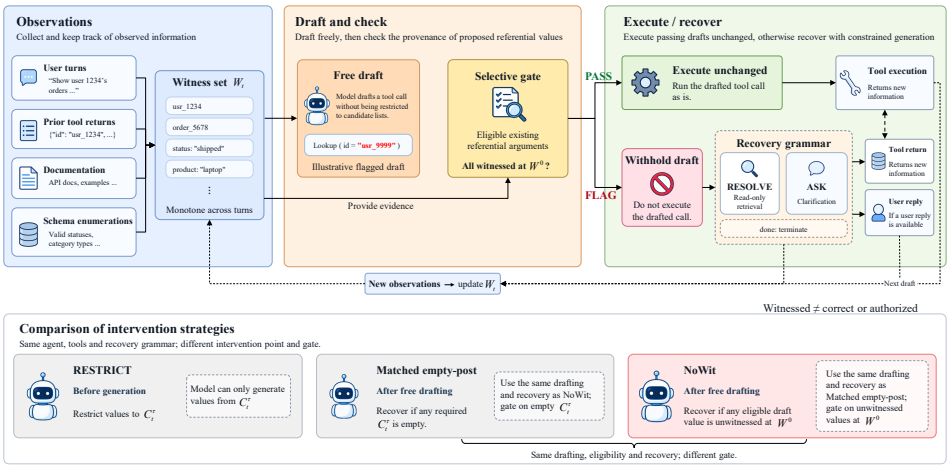

Figure 1: **NoWit separates observations, intervention and recovery.** The identifiers are illustrative. Restrict constrains candidates before generation; NoWit checks the proposed value. The matched control changes the gate while retaining the same recovery path. Read-only calls add observations for subsequent drafts.

decoder (Lee et al., 2026; Nguyen et al., 2026; Reddy et al., 2026; Geng et al., 2025). These results motivate retaining an abstention branch and measuring complete task outcomes alongside constraint satisfaction.

The behaviour we target has also been named directly, and the remedies offered for it are advisory. Agents facing unclear instructions can be prompted to ask, with a benchmark built for it (Wang et al., 2025); when a schema demands a field, models invent one, and an offered escape goes essentially unused (Usman, 2026). Uncertainty can be structured to trigger clarification (Suri et al., 2026), alignment can teach a model when to abstain (Zhou et al., 2026), and grounding can be made action-conditioned with the support set induced automatically (Jiang et al., 2026). Each of these offers a remedy the model may decline, or trains a disposition to take it, and none makes the alternative unreachable; our prompt arm is the measurement of why that matters here, since naming the rule and both remedies moves the fabrication rate by 0.13 percentage points and the task score by nothing (Appendix C.2).

The closest statement of the problem credits support only from successful tool observations or protocol-authorised user fields, and reports that stop-and-re-verify policies reduce unsupported writes without pricing what they cost in autonomy (Zhang et al., 2026b). Our witness predicate uses literal exposure, with error-return eligibility tested explicitly (Appendix K.2). We compare intervention gates and report their execution costs (Section 4.3). A parallel line reasons about provenance at the argument rather than the call, and stops at detection (Appendix E.1).

3 The No-Witness Rule

The intervention policy separates three choices: the evidence that licenses a value, the condition that withholds a draft, and the recovery actions offered afterward. This separation supports both the selective policy and the controlled trigger comparison.

3.1 The Witness Set

Definition 1 (Witness set)**.** At turn t, W_t contains typed literal spans credited from user turns, prior tool-return text, supplied documentation and schema enumerations. It is monotone across turns. A value is witnessed at W^0 if it occurs verbatim, as an exact span or a case-sensitive word-boundary match in a source utterance; at W^1 if it is in the closure of W^0 under five fixed rules; and at W^2 if it is type-consistent. The implementation enforces $W^0 \subseteq W^1 \subseteq W^2$.

W^0 is the operating predicate; W^1 and W^2 are sensitivity levels. The five closure rules cover Unicode, whitespace and case normalisation; number and date reformatting; path projection; schema-enumerated synonyms; and composition of witnessed components (Appendix B.3). Monotonicity preserves credit after context truncation, while phrase matching accepts multi-word observations that a tokenised candidate list may miss. The predicate measures literal exposure, including text returned by failed calls in the frozen BFCL implementation. We examine that source choice in Appendix K.2; the detector and audit rules are specified in Appendix K.

3.2 Two Places to Intervene

For a referential slot of type τ, let C_t^τ be the finite candidate list extracted from W_t. Candidate membership and witness membership can differ because extraction can omit a span that the witness predicate accepts. Both policies retain RESOLVE, a currently callable read-only tool, and ASK, a rendered clarification, as recovery options.

RESTRICT acts before generation. It restricts each referential value to C_t^τ; a tool leaves the call grammar if a required slot has no candidates. The abstention branches remain reachable.

NOWIT acts on a free draft. It checks each eligible existing referential argument against W_t at W^0. If all pass, the draft executes unchanged. Otherwise, the draft is withheld and the model re-decodes under the recovery grammar, which offers read-only retrieval and clarification. Passing preserves that particular call; earlier interventions can still change the subsequent trajectory.

The matched empty-post control retains NOWIT's free drafting, check timing, eligibility filter, recovery grammar and code path. It replaces only the gate, invoking recovery when some required slot has an empty C_t^τ. We vary candidate typing across PERMISSIVE, SHAPED and STRICT. Intervention counts record recovery re-decodings actually performed.

The value-level gate responds to a proposed argument's exposure. The empty-set gate responds to the extractor's available alternatives, including slots other than a flagged argument. Their distinction is directly measurable: a draft can be witnessed while a required candidate list is empty, or unwitnessed while every list is non-empty. The controlled runs quantify the resulting interventions and task outcomes (Section 4.2).

3.3 Guarantees and Costs

The grammar preserves an alternative to executing a tool call.

Proposition 1 (Abstention-reachability)**.** *At every step where the rule fires, at least one non-*`call` *continuation is derivable under the mask. No witness set can leave a tool call as the only derivable action.*

The root alternation is `root` ::= `call` | RESOLVE | ASK | `done`. The ASK and `done` branches are unconditional, so changing W_t cannot remove all alternatives to `call`. Candidate-dependent tools enter the call branch only while their required referential lists are non-empty. This establishes reachability under the implemented whole-action grammar; synthetic experiments measure whether the model actually takes the available escape (Section 4.4). The serving implementation is detailed in Appendix F.1.

RESOLVE uses tools that preserve the benchmark's scored state. Of 170 public methods, 59 are classified as read-only and 0 modify public state, checked for assignments, subscript updates and in-place mutations. This aligns retrieval with the scorer's public-state boundary.

A witness-set update can invalidate a compiled grammar. RESTRICT constrains every step, whereas NOWIT invokes the recovery path on 8.43% of call steps in the primary run. Standalone compile time grows from 63.7 ms at 20 candidates to 758.6 ms at 500, motivating type-matched candidate subsets. We report the additional tool calls and capped tasks separately in Section 4.3.

4 Experiments

We first vary the empty-set trigger with recovery and enforcement fixed, identifying how candidate typing changes intervention frequency and task cost (Section 4.2). The two-environment comparison

Table 1: **Trigger and recovery ablations.** The first block keeps post-hoc free drafting, PERMISSIVE remedy candidates and eligibility fixed; only the trigger changes. W counts unwitnessed writes. Remedy percentages use call-step denominators, and capped counts are official forced terminations. The lower panel retains the historical task-score decomposition, with within-arm Wilson 95% intervals; these are not paired effect intervals. Its emission endpoints were not archived locally and are not imputed. Each row uses all 200 Miss Param tasks.

Policy / trigger	Task	U/N	U (%)	W	Remedy (%)	Capped
Matched post-hoc enforcement and remedy candidates						
Empty: PERMISSIVE	33.0	97/994	9.76	84	0.00	4
Empty: SHAPED	21.5	80/1353	5.91	73	36.44	34
Empty: STRICT	8.5	15/1988	0.75	12	68.21	100
NOWIT: value-level	36.0	26/1,028	2.53	22	8.43	11

Task-score decomposition	Passed / tasks	Task (%)	95% interval
Unconstrained	70/200	35.00	[28.73,41.84]
Phased, no remedy	71/200	35.50	[29.20,42.35]
Detect, then refuse	67/200	33.50	[27.32,40.30]
NOWIT (NoWit)	72/200	36.00	[29.67,42.86]

then measures NoWit's reductions in unwitnessed emissions, writes and tool refusals (Section 4.3). Synthetic trials separate answer coverage from voluntary stopping (Section 4.4). Finally, replication and sensitivity analyses test the stability of the reduction and quantify task-score variation (Section 4.5).

4.1 Experimental Setup and Statistical Protocol

We evaluate frozen Qwen3-8B and Qwen3-14B on BFCL v4 multi-turn tool use and on retail τ^2-BENCH, scoring programmatic accuracy and mean reward respectively. The primary cohort is the 200-task BFCL Miss Param split at site A; site B supplies an independent 8B prompt comparison and the 14B replication, and four BFCL categories supply 800 paired tasks. Census, intervention and seed cohorts are separate runs and are recorded as such (Table S1).

Benchmark arms use thinking mode, pinned revisions, vLLM and XGrammar (vLLM contributors, 2025; Dong et al., 2025), and the synthetic 8B follow-up uses non-thinking formatting. Controls are unrestricted generation, pre-emptive Restrict, the released AWN prompt (Wang et al., 2025), the matched typed empty-post policies, and the historical recovery ablations. Prompts, source eligibility and package pins are in Appendices C.8, G and K.

The primary rate U/N counts operationally unwitnessed referential emissions among all audited referential emissions, and we report writes, task score, remedies and termination alongside it, because a policy can move any one of them alone. Labels come from the automatic EXISTING/NEW/GENERATIVE partition and the W^0 exposure predicate. Rate intervals use 10,000 paired task bootstrap draws, retaining zero-emission tasks and recomputing pooled ratios; seed variation uses a Student t interval over three task-score differences; equivalence uses the pre-stated ± 7-point margin and a 90% interval, since absence of significance is not equivalence.

4.2 Isolating the Intervention Trigger

The typed empty-post sweep holds free drafting, eligibility, recovery and code path fixed, changing only the candidate type function used by the empty-set gate. PERMISSIVE includes ordinary user spans, keeps every required list non-empty in these runs and invokes no remedies. It scores 33.0% while passing every draft through the shared pipeline.

Stricter typing increases interventions and reduces task success. SHAPED invokes remedies on 36.44% of call steps and STRICT on 68.21%, and their task differences from PERMISSIVE are -11.5 pp $[-18.5, -4.5]$ and -24.5 pp $[-31.0, -18.0]$, measured after every recovery attempt has run, so this is the cost of the policy rather than the cost of an unrecovered interception. STRICT does drive the unwitnessed rate below NoWit's, to 0.75%, and pays 8.5% task success for it (Figure 3a). Across

Table 2: **Main results across models and environments.** BFCL task scores are accuracy (%); retail scores are mean reward. U/N is the count of operationally unwitnessed referential emissions, and W counts unwitnessed writes. Intervals are paired episode 95% intervals for rate differences from the baseline in the same block. Baseline contrasts are zero by definition (ref.). Retail intervals retain all 114 task IDs, including recorded infrastructure errors. A/B denote independent serving sites. Bold identifies the proposed policy, not the best value. †: zero follows from aligned construction and audit.

Policy	Task	U/N	U (%)	W	ΔU pp [95% CI]
BFCL Miss Param / Qwen3-8B / site A					
Unconstrained	35.0	84/971	8.65	67	0.00 (ref.)
RESTRICT pre-emptive	5.0	0/2638	0.00†	0	-8.65 [-11.20,-6.45]
NOWIT (NOWIT)	36.0	26/1028	2.53	22	-6.12 [-8.44,-4.09]
BFCL Miss Param / Qwen3-8B / site B					
Unconstrained	34.00	85/944	9.00	73	0.00 (ref.)
AwN prompt	28.50	47/719	6.54	40	-2.47 [-5.40,+0.11]
NOWIT (NOWIT)	36.50	25/1022	2.45	22	-6.56 [-9.25,-4.18]
BFCL Miss Param / Qwen3-14B / site B					
Unconstrained	35.00	138/1059	13.03	115	0.00 (ref.)
NOWIT (NOWIT)	37.00	27/1132	2.39	23	-10.65 [-15.59,-6.73]
Retail / Qwen3-8B / paired intervention cohort					
Unconstrained	0.04	40/306	13.07	9	0.00 (ref.)
NOWIT (NOWIT)	0.05	0/234	0.00†	0	-13.07 [-17.79,-8.88]

the tested sweep, increasing empty-set interventions trades lower unwitnessed rates for substantial task loss.

Pre-emptive restriction provides a separate enforcement comparison. RESTRICT reaches a zero unwitnessed rate by aligned construction, because its candidate list is a subset of what the audit accepts, and it scores 5.0% with 110 of 200 tasks hitting the step cap. Figure 2b separates remedies fired on wholly witnessed drafts from those fired on drafts containing a flagged value, which is where the two gates diverge.

The historical recovery panel completes the picture, with 71/200 successes under detection and no remedy, 67/200 under refusal, against 72/200 for NOWIT. These task-only ablations compare the available recovery variants (Appendix J.1). A same-detector ordinary-repair arm with matched read actions and retry budget remains the comparator needed to isolate constrained recovery's incremental benefit.

4.3 MAIN RESULTS ON TWO ENVIRONMENTS

Testing the value instead of the set removes most of the behaviour while touching one call step in twelve. The primary rate falls from 8.65% to 2.53%, a paired difference of -6.12 pp $[-8.44, -4.09]$ (Table 2), with the gate firing on 8.43% of call steps, and unwitnessed state-mutating calls fall from 67 to 22. The direction repeats at site B and at 14B, where the larger baseline rate falls further, from 13.03% to 2.39%.

The reduction accompanies additional activity. NOWIT invokes 141 remedies over 1,673 call steps, its trajectories carry 9.85% more tool calls than the baseline's, 1,673 against 1523, and forced terminations rise from 2 to 11 (Figure 3c). Section 4.5 reports the task-score intervals and the pre-stated equivalence margin.

The released prompt baseline leaves more residual unwitnessed emission. AWN (Wang et al., 2025) is the released ask-when-unclear method for this failure, run with its own prompt at its pinned commit. Its paired rate-difference interval includes zero, and it leaves 40 unwitnessed writes at 6.54% exposure. BFCL's user turns are scripted, so clarification there ends a turn while NOWIT can also select a read-only tool, which means the comparison prices enforcement together with access to recovery, and τ^2-BENCH below is the board where asking is answered.

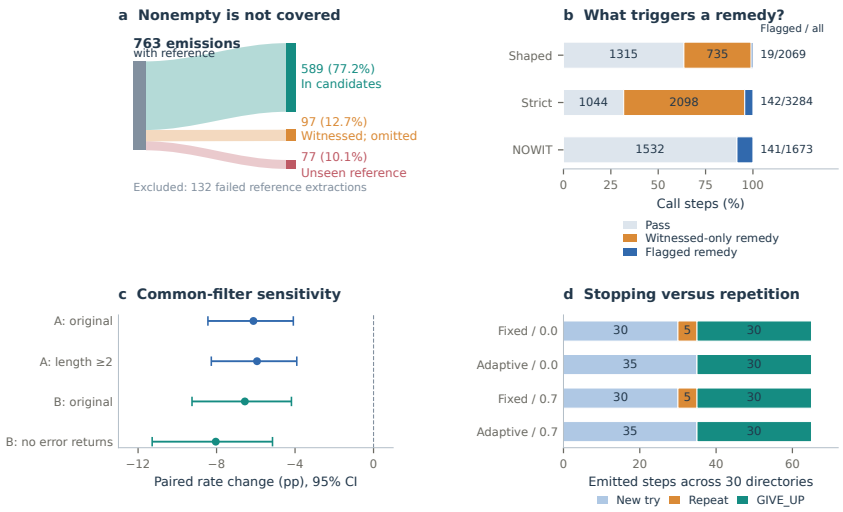

Figure 2: **Coverage, gate selectivity and stopping are three different things.** (a) Reference coverage on the diagnostic census, excluding failed reference extraction; (b) full-run call-step partitions, with segment counts and right-side flagged/all counts; (c) paired sensitivity to common exposure filters; (d) every emitted lookup or stop across 30 synthetic directories per arm and temperature.

On the environment that adjudicates each emission, the refusals disappear. BFCL's grader compares end state, so an unwitnessed write that happens to coincide with the reference costs nothing there, whereas retail τ^2-BENCH validates referents at the emission, and refusals on unwitnessed emissions fall from 38 to 0. Mean reward remains near the floor, 0.04 versus 0.05. The refusal outcome records failed requests observed by the environment; the provenance outcome records whether their arguments were witnessed. On BFCL, the corresponding reduction in unwitnessed state-mutating emissions is 67 to 22 (Appendix L.1).

4.4 Candidate Coverage Against Stopping Behaviour

Non-empty candidates leave measurable gaps in episode-reference coverage. On the diagnostic census, 77 of 763 emissions have no witnessed value from the episode's reference solution despite a non-empty candidate list, and a further 97 omit a reference value that was witnessed (Figure 2a). The two groups distinguish missing witnessed reference values from extraction omissions. Episode-reference coverage is a diagnostic proxy; valid off-reference actions can exist.

A known unique target lets the synthetic task isolate answer exclusion from stopping. In the original Qwen3-0.6B protocol, answer-excluding candidates reach the 12-step cap in every trial. In the fully logged 8B protocol, all 30 wrong-set trials per temperature stop, for both fixed and adaptive candidate sets, and pruning removes 5 repeated identifiers per temperature without ever making the missing answer available (Figure 2d). The protocols differ in model, prompts and directory construction. Together they show that answer exclusion prevents successful lookup while leaving stopping behaviour protocol-dependent. Figure S3 shows every trajectory, including the free and answer-containing controls.

4.5 Robustness to Seeds, Sites and the Exposure Definition

Task-score effects vary across seeds and are close to zero in the pooled comparison. Across four BFCL categories the task difference is +1.38 pp, 95% $[-1.75, +4.50]$ with McNemar $p = 0.44$. The three Miss Param seed effects are +1.00, -4.00 and -2.00 pp, with mean -1.67 pp and 95% t interval $[-7.92, +4.58]$, and the 90% interval $[-5.91, +2.58]$ clears the pre-stated ± 7-point margin. This test quantifies equivalence at that specified tolerance (Figure 3d).

Every seed pair reduces the rate, and one task carries much of what remains. For seed two the paired interval excludes zero while the historical independent-arm interval includes it, and a single task contributes 49% of the residual numerator. The task atlas and concentration curves keep that

Table 3: **Task coverage and seed robustness.** Top: all four BFCL multi-turn categories, with the same baseline–NoWit pairing. Bottom: three server-seed pairs on Miss Param; each interval resamples tasks within that seed. $B/$NoWit denotes baseline and treatment in the same run pair. Seed-level task uncertainty is broader than any one episode-bootstrap interval.

BFCL category	Tasks	Baseline	NoWit	Δ task (pp)
Base	200	47.50	51.00	+3.50
Missing function	200	46.00	47.50	+1.50
Missing parameter	200	35.00	36.00	+1.00
Long context	200	29.50	29.00	-0.50
All categories	800	39.50	40.88	+1.38

Server seed	Task B/NoWit	Δ task	U (%) B/NoWit	ΔU pp [95% CI]
Default	35.0/36.0	+1.00	8.65/2.53	-6.12 [-8.44,-4.09]
One	36.50/32.50	-4.00	8.73/2.50	-6.24 [-8.66,-4.09]
Two	33.50/31.50	-2.00	8.28/3.85	-4.43 [-7.61,-0.22]

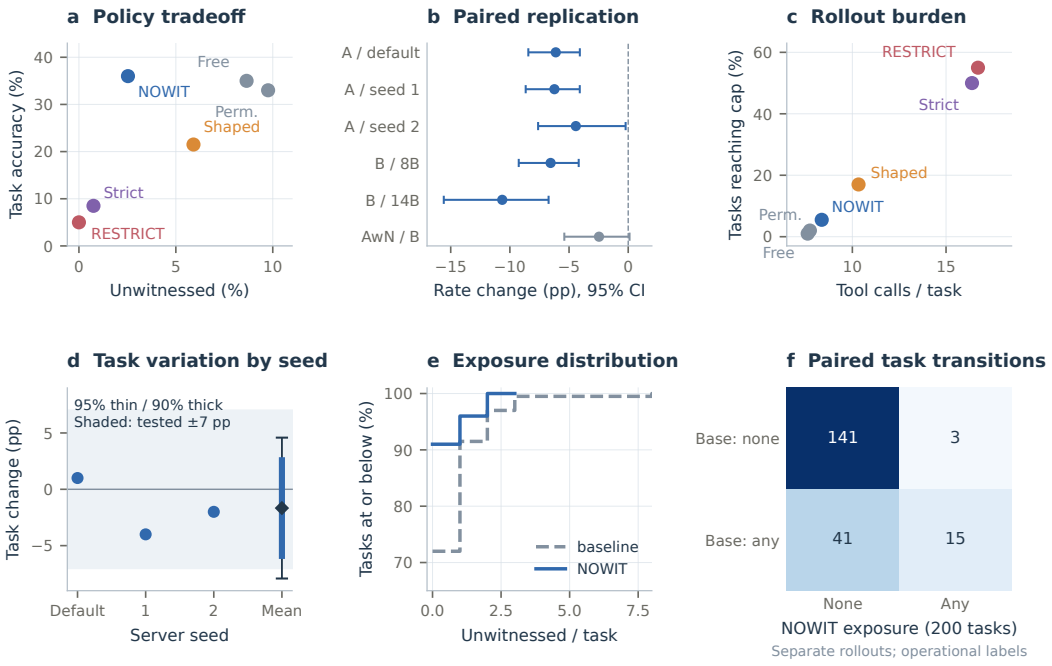

Figure 3: **Exposure reduction, task performance and rollout cost, read jointly.** (a) Six site-A policies; (b) paired rate intervals across seeds, sites and model sizes; (c) calls and capped-task frequency; (d) seed-level task effects with 95%/90% intervals and the tested margin; (e) primary per-task exposure CDF; (f) all 200 paired task outcomes for any unwitnessed emission. Full distributions appear in Figure S4.

heterogeneity visible rather than averaging it away (Figure S4 and appendix J). Figure 4 resolves this aggregate effect into task-state transitions, changes in exposure opportunities and tool-specific residuals. Most task IDs have unchanged unwitnessed counts, while a smaller subset accounts for the reduction; the residual concentration varies across cohorts.

The reduction survives the tested exposure definitions: symmetric short-value filtering and exclusion of explicit error returns (Table 4 and fig. 2c). Removing JSON error returns withdraws credit from 14 site-B baseline emissions and 0 NoWit emissions, and the any-violation task difference remains -17.5 pp $[-23.5, -11.5]$, while a broader error-text filter gives the same emission endpoints (Figure S5 and appendix K.2). These are re-scorings of fixed trajectories, so they test the definition rather than a re-run policy.

Table 4: **The reduction survives common measurement filters.** Entries show unwitnessed/total emissions and their percentage on the same frozen baseline–NoWit trajectories. A/B are separate 200-task cohorts. Filters are applied symmetrically; intervals use paired task resampling. These are offline relabelings, not new policy rollouts.

Exposure definition / site	Baseline U/N (%)	NoWit U/N (%)	ΔU pp [95% CI]
Original exposure / A	84/971 (8.65)	26/1028 (2.53)	-6.12 [-8.44,-4.09]
Values of length ≥ 2 / A	59/946 (6.24)	3/1005 (0.30)	-5.94 [-8.27,-3.91]
Original exposure / B	85/944 (9.00)	25/1022 (2.45)	-6.56 [-9.25,-4.18]
Exclude error returns / B	99/944 (10.49)	25/1022 (2.45)	-8.04 [-11.28,-5.15]

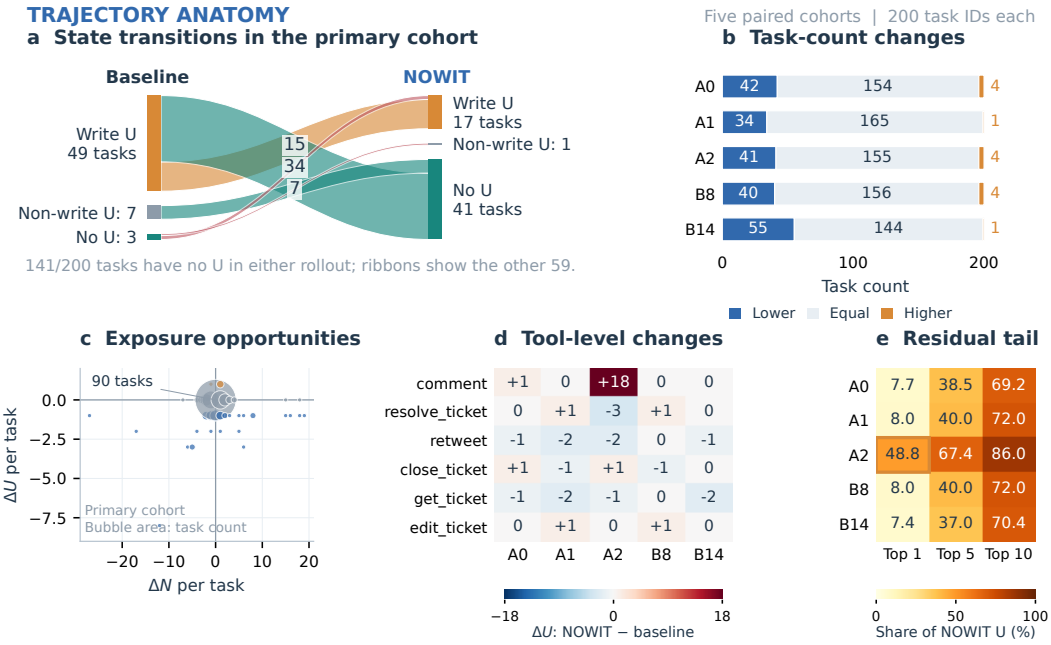

Figure 4: **Task-level anatomy of exposure reduction.** (a) Primary-cohort transitions among no U, non-write U only and any unwitnessed write; no-U pairs are omitted. (b) Lower/equal/higher unwitnessed counts per task. (c) Changes in audited referential emissions N and unwitnessed emissions U. (d) ΔU for the largest NoWit residual tool sources. (e) Shares of each NoWit cohort's U in its highest-burden tasks. A0/A1/A2 denote site-A seeds; B8/B14 denote site-B model sizes. Pairs link task IDs across separate rollouts.

5 Conclusion

NoWit checks the provenance of a proposed referential value before its call executes. Separating this gate from candidate construction and recovery exposes the task cost of the tested empty-set policies. The value-level policy reduces unwitnessed emission from 8.65% to 2.53% at 8B and 13.03% to 2.39% at 14B, while removing the observed 38 retail refusals on unwitnessed emissions. Candidate coverage, stopping, task performance and execution cost provide complementary measures of the policy's effect. Human validation of the exposure instrument and a matched ordinary-repair comparison are the next steps toward measuring the incremental benefit of constrained recovery (Appendices F and K).

References

Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ziyi Xu, Yilong Zhao, Ruihang Lai, and Tianqi Chen. XGrammar: Flexible and efficient structured generation engine for large language models. In *Proceedings of Machine Learning and Systems (MLSys)*, 2025.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of EMNLP*, pp. 10932–10952, 2023. doi: 10.18653/v1/2023.emnlp-main.674. URL https://aclanthology.org/2023.emnlp-main.674/.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. JSONSchemaBench: A rigorous benchmark of structured outputs for language models, 2025. URL https://arxiv.org/abs/2501.10868.

Guidance AI contributors. llguidance: Low-level guidance — constrained decoding for large language models. https://github.com/guidance-ai/llguidance, 2026.

Lai Jiang, Cheng Qian, Zhenhailong Wang, Pan Lu, Heng Ji, and Hao Peng. ACCORD: Action-conditioned contextual grounding for language agents, 2026. URL https://arxiv.org/abs/2606.16432.

Ivan Yee Lee, Loris D'Antoni, and Taylor Berg-Kirkpatrick. The format tax, 2026. URL https://arxiv.org/abs/2604.03616.

Janghoon Lee. Repair, not improvement: Decomposing constrained decoding in tool-call abstention, 2026. URL https://arxiv.org/abs/2608.13959.

Fangzheng Li, Aimin Zhang, and Chen Lv. Constraint tax in open-weight LLMs: An empirical study of tool calling suppression under structured output constraints, 2026a. URL https://arxiv.org/abs/2606.25605.

Linzhang Li, Yixin Dong, Guanjie Wang, Ziyi Xu, Alexander Jiang, and Tianqi Chen. XGrammar-2: Dynamic and efficient structured generation engine for agentic LLMs, 2026b. URL https://arxiv.org/abs/2601.04426.

Ryoto Miyamoto, Xin Fan, and Hayato Yamana. Policy-conditioned constrained decoding for column-level access control in text-to-SQL, 2026. URL https://arxiv.org/abs/2607.12341.

Ngoc Trinh Hung Nguyen, Alonso Silva, Laith Zumot, Liubov Tupikina, Armen Aghasaryan, and Mehwish Alam. Thinking before constraining: A unified decoding framework for large language models, 2026. URL https://arxiv.org/abs/2601.07525.

Gabriel Poesia, Alex Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2201.11227.

Jaideep Ray. The constraint tax: Measuring validity-correctness tradeoffs in structured outputs for small language models, 2026. URL https://arxiv.org/abs/2605.26128.

Avinash Reddy, Thayne T. Walker, James S. Ide, and Amrit Singh Bedi. The hidden cost of structured generation in LLMs: Draft-conditioned constrained decoding, 2026. URL https://arxiv.org/abs/2603.03305.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of EMNLP*, pp. 9895–9901, 2021. doi: 10.18653/v1/2021.emnlp-main.779. URL https://aclanthology.org/2021.emnlp-main.779/.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.2028. URL https://aclanthology.org/2026.findings-acl.2028/.

Rana Muhammad Usman. PhantomFill: When the form demands an answer, language models invent one, 2026. URL https://arxiv.org/abs/2607.20492.

vLLM contributors. vLLM: Easy, fast, and cheap LLM serving. https://github.com/vllm-project/vllm, 2025.

Wenxuan Wang, Shi Juluan, Zixuan Ling, Yuk-Kit Chan, Chaozheng Wang, Cheryl Lee, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R. Lyu. Learning to ask: When LLM agents meet unclear instruction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 21773–21784, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1104. URL https://aclanthology.org/2025.emnlp-main.1104/.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models, 2023. URL https://arxiv.org/abs/2307.09702.

Xingzi Xu and Karim Bouyarmane. Trie automata for constrained decoding over large finite sets, 2026. URL https://arxiv.org/abs/2608.12574.

Shuoming Zhang, Ruiyuan Xu, Haofeng Li, Qiuchu Yu, Yangyu Zhang, Chunwei Xia, Xiaobing Feng, Chenxi Wang, Huimin Cui, and Jiacheng Zhao. Decode-time grammars: Constrained LLM generation over a refinement order of grammar fragments, 2026a. URL https://arxiv.org/abs/2607.18357.

Zibo Zhang, Han Zhou, Longyan Tan, and Chenyu Tang. Tracing what agents change when required information is missing. Workshop on FAGEN at ICML 2026 (Poster), 2026b.

Yijin Zhou, Linqian Zeng, Xiaoya Lu, Wenyuan Xie, Dongrui Liu, Junchi Yan, and Jing Shao. Exploring agentic tool-calling decisions via uncertainty-aligned reinforcement learning, 2026. URL https://arxiv.org/abs/2606.06976.