

Do Tool Schemas Fully Specify Argument Binding?

Jai Kumar Sharma
Virginia Tech
jaisharma@vt.edu

Peeyush Tapadiya
Accenture
peeyushtapadiya@accenture.com

Abstract

Tool schemas constrain the structure of a call but need not determine how request evidence becomes argument values. We formalize this layer as a field-level binding contract over presence, evidence, ordered operations, parameters, and validation, and state an identifiability boundary: no schema-only estimator can guarantee recovery of policy distinctions absent from its input. We audit all 1,311 filtered requests in pinned Berkeley Function-Calling Leaderboard (BFCL) V4 Live Simple and Multiple with five open checkpoints and the official abstract-syntax-tree (AST) checker. A partial structural diagnostic and AST acceptance are non-nested: 1,470 of 6,555 calls pass only the diagnostic and 50 pass only AST. A stratified AI-only audit finds coarse concern detection far more consistent across two isolated passes than fine attribution. A pre-specified test of a deterministic schema-derived operation-label encoding gives 67.70% AST acceptance versus 68.27% under the same guidance without field tags (-0.56 points; 95% task-cluster CI -1.17 to $+0.05$; 186 fixes, 223 breaks). Value-only turnover is nearly balanced; the net imbalance sits in key-set changes. The point estimate reverses sign on one post-hoc out-of-panel checkpoint; thus no improvement is established on the fixed panel, and independently authored contracts remain untested.

1 Introduction

One BFCL interface exposes an optional string filter, described only as “a filter string to apply to the results, such as filtering by location name or type.” Asked for locations named *alpha*, *beta*, and *gamma*, a model emits *alpha*, *beta*, *gamma*; the benchmark’s accepted answers are *name IN ('alpha', 'beta', 'gamma')* and *name:alpha OR name:beta OR name:gamma*. All three representations satisfy the declared string type, so none of the declared structural constraints distinguishes

them, and the exposed description states no grammar that would. Deciding between them is a question about how request evidence becomes a value, not about type.

At the first action boundary of a tool-using agent, the model must choose a function and bind such evidence before any tool can execute. Existing evaluations already separate function selection, structure, and argument correctness (Kokane et al., 2025; Patil et al., 2025; Garimella et al., 2026), and recent methods recognize that argument values can require specification-driven modification rather than extraction (Cheng et al., 2026; Ghoshal et al., 2026). We propose no new correctness stage and do not claim that well-formed calls with wrong values are new. We ask instead the question in this paper’s title: *does the ordinary schema fully specify how each field’s value should be produced?* It need not: where a policy-relevant distinction is absent from the exposed metadata, no schema-only estimator can guarantee its recovery.

We do not test an authored binding contract. We ask how a correct-function partial structural diagnostic and BFCL AST acceptance overlap on a complete fixed panel (RQ1); how stable an explicitly AI-only attribution of their residual is (RQ2); and whether a deterministic schema-derived operation encoding improves first-call AST acceptance over the same guidance without field tags (RQ3). Three findings follow: the two criteria are non-nested in both directions over 6,555 model–task calls. Coarse concern detection is far more consistent across isolated AI passes than fine attribution. And the tags produce 409 paired outcome transitions without aggregate improvement. These are respectively a benchmark result, an AI-sensitivity result, and a proxy-encoding result: not execution correctness, not human prevalence, and not a test of an authored contract. Together they motivate, but do not validate, independently authored binding contracts.

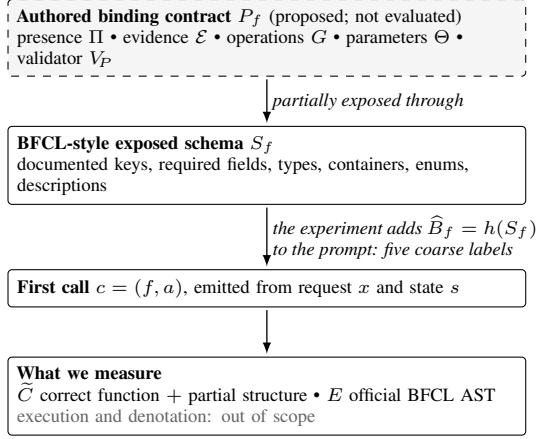


Figure 1: The authored contract is a proposal: it is partially exposed through the schema, and we never instantiate it. What the experiment adds to the prompt is $\hat{B}_f = h(S_f)$, a coarse re-encoding of cues the schema already contains, so it cannot supply a distinction the schema omits.

2 Schemas and binding policies

Figure 1 summarizes the layer this paper is about. Let x be a request, s the available interaction state, $\mathcal{F}$ a candidate registry, and $c = (f, a)$ a first call with argument tree a . The ordinary schema S_f specifies documented keys, required fields, types, containers, enums, and possibly formats. A standards-complete schema validator would define a structural predicate $V_S(c)$. Our empirical pipeline does *not* implement that full predicate. Let K, R, T , and M denote documented keys, required-field presence, primitive-type matching, and exact-enum matching, respectively. We report

$$\tilde{V}(c, S_f) = \mathbb{K}[K(c) \wedge R(c) \wedge T(c) \wedge M(c)], \quad (1)$$

after the output has been parsed into the supported universal call shape. For comparison with a benchmark target function f^* , we use $\tilde{C}(c) = \mathbb{K}[f = f^*] \tilde{V}(c, S_f)$. The tilde is substantive: neither $\tilde{V}$ nor $\tilde{C}$ is full JSON-Schema validity.

What the schema need not fix is how a value should be produced. Write this layer, for field path p of f , as a *binding contract*

$$P_f(p) = \langle \Pi, \mathcal{E}, G, \Theta, V_P \rangle, \quad (2)$$

with a presence policy Π (required, omit-if-absent, emit-default, conditionally required), admissible evidence sources $\mathcal{E}$, an *ordered* operation program G , the parameters Θ that make each operation determinate (alias maps, target formats, units, grammars, rules), and a postcondition V_P . The contract

is a fixed property of the interface, mapping request and state (x, s) to permitted presence and value outcomes: different outputs on different requests instantiate one policy. We introduce P_f as a conceptual decomposition, a vocabulary for saying precisely what an ordinary schema can leave open, rather than an executable semantics: we do not give the tuple a denotation here. We also do not evaluate an authored instance of it; RQ3 tests only a schema-derived projection of its operation component.

Nor is the ordinary schema silent about all of it: required arrays constrain part of Π , enum constrains part of Θ and V_P , and richer standards encode conditional presence and some serialization (Wright et al., 2022b,a; OpenAPI Initiative, 2024). Two keywords matter for the presence analysis in Section 6.1: under the 2020-12 validation vocabulary, `default` is annotation metadata that neither requires nor forbids emitting the field, and `format` is by default an annotation rather than an assertion (Wright et al., 2022a). The claim defended here is narrower: the BFCL-style exposed schemas we study do not generally *identify* a complete request-evidence-to-value policy.

The component this paper can put at field granularity is the operation vocabulary. Projecting P_f onto the unordered set of operation labels used in G , which discards order, the parameters Θ , the evidence sources $\mathcal{E}$, and presence Π , leaves an *authored operation signature*

$$B_f(p) \subseteq \mathcal{O}, \quad \mathcal{O} = \{\text{COPY, SELECT, NORMALIZE, COMPOSE, DERIVE}\}, \quad (3)$$

which declares only which request-to-value operations the interface permits. B_f is authored metadata published alongside the schema: public routing information, not a prediction of a model’s private reasoning. A field may permit several classes at once, and the set is unordered. The classes mean: `COPY` reproduces explicit request evidence without semantic transformation; `SELECT` chooses from an explicitly licensed closed set; `NORMALIZE` applies an explicitly licensed format, case, representation, date, time, or unit transformation; `COMPOSE` constructs a list, object, query expression, joined value, or nested structure from multiple licensed pieces; and `DERIVE` computes a value only through an explicitly documented deterministic rule.

The projection is lossy by construction, and we do not claim the five labels are exhaustive or mutually exclusive. Order alone can matter: $\text{SELECT}(\text{alias}) \rightarrow \text{NORMALIZE}(\text{format})$ and its reverse need not agree, so an unordered set cannot by itself determine a value. Presence is a *separate component* of P_f , not a sixth operation, and we avoid calling it orthogonal because Section 6 shows that prompting the vocabulary changes presence behavior: the components are analytically distinct but empirically coupled.

The evaluated object is instead

$$\hat{B}_f = h(S_f), \quad (4)$$

where h is a frozen deterministic heuristic over ordinary schema metadata. Estimating B_f is not the same problem as recovering P_f , and the distinction bounds what any schema-only reader can guarantee.

Observation 1 (Schema-only identifiability boundary). *Let $\sigma(I)$ be the metadata visible to an estimator for interface I , and let $P(I)$ and $B(I)$ collect the field-level contracts and signatures associated with I . Let g be any deterministic estimator of the full contract. If $\sigma(I_1) = \sigma(I_2)$ but $P(I_1) \neq P(I_2)$, then g returns one output for both and fails to recover at least one contract. For an estimator of the coarse signature the same argument requires $B(I_1) \neq B(I_2)$: policies that differ only in parameters Θ share a signature, and a signature estimator can be correct for both.*

We state this immediate observation only to fix an interface boundary: no schema-only estimator, whatever its quality, can guarantee recovery of policy distinctions absent from its input. It predicts nothing about the direction or magnitude of any particular heuristic, including h , which RQ3 measures empirically.

Our data illustrate the projection loss rather than an exact metadata collision. Two Live Simple interfaces declare a string location field whose description opens with the same “City, State (abbr)” template; their complete descriptions differ, and h assigns both $\{\text{NORMALIZE}\}$. One is satisfied only by a US state abbreviation and the other only by a country name (Section 6.1): a difference of target format and region, which lives in Θ , exactly the component the unordered signature discards. The weather description states a rule for choosing between the two forms (a state abbreviation when the city has a state, a country otherwise);

the hotel description lists both forms without saying which applies. What each interface requires here is fixed by BFCL’s accepted answers, which Section 4 shows are sometimes more permissive than the documented interface. The coarse signature retains neither the regional choice nor the evaluator-specific target, so the loss is in the projection rather than in the schema. This sets what RQ3 can establish: not whether h recovers B_f , but whether reorganizing cues the schema already contains changes behavior, and in which direction.

BFCL supplies a separate predicate $E(c, Y)$ through its AST checker and set of accepted answers Y . The checker canonicalizes selected surface forms and handles listed alternatives. Neither E nor $\tilde{C}$ equals downstream denotational correctness under tool execution. Our RQ1 overlap therefore compares two operational criteria, not “syntax” followed by semantic truth.

3 Study design

3.1 Data, models, and generation

We use all 258 BFCL V4 live_simple cases and all 1,053 live_multiple cases at pinned commit f7cf7359b7ac. Every selected case requires one target call. Simple provides one candidate function; Multiple provides 2–37. BFCL describes the queries and tool documentation as user-contributed, but the public release is filtered, deduplicated, standardized, and redacted: a broad external benchmark, not contamination-free deployment traffic.

The fixed panel contains Qwen3-0.6B, Qwen3-1.7B, Qwen3-4B-Instruct-2507 (Yang et al., 2025), Qwen2.5-14B-Instruct (Yang et al., 2024), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). Three checkpoints share a model family, and the 14B model is a larger local scale point rather than a frontier-model study. Each model produces one greedy output per condition and case, with its pinned chat template, seed 101, a 256-new-token cap, and thinking disabled where exposed. Every main-panel model-condition cell contains all 1,311 tasks; no task rows were dropped (checkpoint revisions and runtime: Appendix B).

Earlier controlled probes motivated the hypothesis but were highly templated, so we demote them to Appendix C and exclude them from the headline claims.

3.2 Five conditions

Every condition receives the same candidate registry and universal one-call JSON shape. The five conditions are:

1. **Schema control:** the original schema prompt, with untouched schemas; this is a byte-identical drift/replay control.
2. **Guidance:** untouched schemas plus definitions and cautions for the closed five-operation vocabulary in Eq. 3.
3. **Placeholder:** the same guidance, with `x-binding-operations:["UNSPECIFIED"]` repeated at every field. This is a repeated-metadata format control; the token itself is not semantically neutral.
4. **Schema tags:** the same guidance, with the frozen target-blind $\hat{B}_f$ attached to every field of every candidate.
5. **Shuffled tags:** the same guidance and an exact within-registry permutation of the heuristic tag arrays. It preserves the tag multiset, metadata keys, and prompt character count, but is a semantic perturbation, not a neutral placebo.

The vocabulary contains no operation aliases beyond the five classes, and DERIVE carries the deterministic-rule definition above. The heuristic uses only type, enum, and field-description cues; it never loads requests, targets, outputs, or the identity of the correct candidate.

The condition set and the primary comparison (schema tags minus guidance) were informed by a completed pilot study but fixed before any output from the present evaluation was inspected. Guidance rather than schema control is the baseline because it is the condition that differs from the tags only in the field-level metadata; schema control differs in the operation instructions as well. We therefore call it pre-specified rather than preregistered, and because the pilot reused these tasks and models we do not describe the present evaluation as a replication. Tags versus placeholder, tags versus shuffled, guidance versus schema control, and all decompositions are exploratory.

3.3 Metrics, inference, and evidence gate

Official BFCL AST acceptance E is primary. We report $\tilde{C}$ separately and never nest AST beneath it.

Comparisons are paired within each model–task unit. The primary percentile interval uses 10,000 task-cluster bootstrap draws (seed 20260728), retaining all five fixed-model rows whenever a task is sampled. It represents task resampling for this fixed, family-correlated panel; it does not treat models as sampled, does not fully capture dependence between related interfaces, and does not estimate stochastic-decoding variance.

Before outcome analysis, an automated stage reconstructed every prompt, checked the pinned model and configuration records, and replayed all 32,775 scored rows against an isolated pinned BFCL scorer: deterministic internal consistency, not cryptographic provenance. A second implementation recomputed the reported totals and intervals from the row ledgers independently.

Three populations recur below: the *main run*, the final five-condition evaluation of Tables 1 and 3; the *audit run*, an earlier execution of the schema-control condition whose AST rejections RQ2 samples; and the *follow-up*, the later 121-task extension of Appendix I, which reuses the main-run guidance rows for its tasks. Audit labels stay tied to the audit run and are not transferred to the main run.

4 RQ1: how do the two criteria overlap?

Before attributing any discrepancy to binding policy, we first measure how the correct-function partial structural diagnostic and BFCL AST acceptance disagree. Table 1 gives complete schema-control accounting over 6,555 model–task units. The correct-function partial structural diagnostic $\tilde{C}$ passes for 5,996 calls (91.47%), while AST accepts 4,576 (69.81%). Crucially, the criteria are non-nested: 1,470 calls pass only $\tilde{C}$, and 50 pass only AST. The 1,470-call cell identifies a post-diagnostic benchmark disagreement, not a semantic-error rate; the reverse cell confirms that our partial diagnostic can also reject calls accepted by BFCL.

The two panels report the same 6,555 units under different rules, and their residual counts differ by three. Panel (a) crosses the two criteria directly. Panel (b) assigns each unit to the first gate it fails, in this order: (1) AST accepted; (2) protocol or shape failure, where the whole output is not exactly one JSON object with the keys `tool` and `arguments` and nothing else (the strict protocol), or no single call can be recovered from

	AST accepted		AST rejected		Total
	<i>n</i>	%	<i>n</i>	%	<i>n</i>
$\tilde{C}$ passes	4,526	69.05	1,470	22.43	5,996
$\tilde{C}$ fails	50	0.76	509	7.77	559
Total	4,576	69.81	1,979	30.19	6,555

(a) Direct criterion overlap: non-nested.

Ordered outcome	<i>n</i>	%
AST accepted	4,576	69.81
Protocol / supported-shape failure	19	0.29
Wrong function	277	4.23
Right function, $\tilde{C}$ fails	216	3.30
Post-diagnostic AST rejection	1,467	22.38

(b) Ordered diagnostic localization.

Table 1: Schema-control results across 1,311 tasks and five checkpoints. Percentages use 6,555 as denominator. Panel (a) directly crosses the correct-function structural diagnostic $\tilde{C}$ with official AST. Panel (b) applies ordered localization gates and is therefore not obtained by relabeling the four overlap cells.

the first JSON value in the output (the supported shape); (3) wrong function; (4) right function but $\tilde{V}$ fails; (5) post-diagnostic AST rejection. Both $\tilde{C}$ and the AST checker run on the recovered call, so an output that wraps a valid call in extra text or a provider-style envelope can pass $\tilde{C}$ yet fail the strict protocol gate; three of the 19 gate-(2) units are such calls, which is why Panel (a)’s structural-pass/AST-reject cell holds 1,470 and Panel (b) 1,467. Outputs from which no call can be recovered fail $\tilde{C}$, receive an AST rejection, and fall in gate (2). RQ2 audits a separately frozen population of 1,463 such rejections from the audit run (Section 5); we do not force the 1,470, 1,467, and 1,463 counts to agree.

The gap is not driven by one weak checkpoint: $\tilde{C}$ exceeds AST acceptance for all five, from 13.27 points on Qwen2.5-14B to 36.69 on Qwen3-0.6B (Table 7), and the reverse cell is nonempty for each (4–16 calls; Appendix H). With five related checkpoints we do not read a model-strength trend into that spread.

Two implementation details run against the headline direction. Our diagnostic treats an object with declared properties as closed, whereas JSON Schema is open by default; admitting undocumented keys would enlarge the disagreement cell from 1,470 to 1,503 and shrink the reverse cell from 50 to 46, so the closed reading understates the gap. And the reverse cell is not residual noise: all 50 cases select the correct function, and they separate into two kinds. In 46 of them BFCL accepts more than the documented interface sanctions: enum values outside the documented enum (30) and nulls where the schema declares a primitive (16). The remaining four are false negatives of our closed-object reading rather than benchmark permissiveness, since the interface they come from directs callers to replace its placeholder properties with actual names. The 46 are what make the dis-

agreement between the documented interface and the benchmark’s accepted-answer semantics two-sided (Appendix H).

The gap also holds outside the panel: a post-hoc Llama-3.1-8B checkpoint from a third family shows a 21.28-point gap against the panel’s 21.66 (Appendix K).

The forward disagreement is heterogeneous: one audited call alters a host identifier the request supplies verbatim; another emits a comma-joined filter where BFCL requires a query-language form. Both are AST rejections on different evidence, which motivates RQ2 rather than counting every residual as one kind of failure.

5 RQ2: how stable is AI-only attribution?

In the audit run, from which the audit packet was built, 1,463 calls passed protocol, selected the reference function, passed $\tilde{V}$, and failed AST, dominated by string/scalar disagreements (1,095, 74.8%) and optional-value ones (290, 19.8%). These are checker pathways, not semantic causes: an answer-listed optional field can impose a presence choice the schema does not require.

To probe that distinction, we drew a stratified probability sample of 160 model–task units from those 1,463 rejections and collapsed byte-equivalent evidence only after sampling, yielding 149 coding cases. Phase A showed the conversation, candidate schemas, and prediction while withholding the BFCL answer and checker report, and asked for one of three coarse judgments: the prediction appears to be a model error, appears acceptable, or the interface or request is ambiguous. Each pass’s Phase A was frozen before Phase B revealed the withheld fields and asked for one of seven fine categories. The passes ran under procedural isolation, and all Phase-B disagreements were separately AI-adjudicated (Appendix L). Be-

Coding stage	Agreement	κ
Phase A: coarse (3 classes)	138/149 (92.62%)	0.879
Phase B: fine (7 classes)	74/149 (49.66%)	0.368

Table 2: The RQ2 result: two procedurally isolated passes agree on the coarse judgment far more often than on the fine category. Both passes and the adjudicator share an underlying model family, so this measures protocol stability, not inter-annotator reliability.

cause both passes and the adjudicator share an underlying model family, we do not treat their agreement as inter-annotator reliability.

Phase-A coarse agreement was 138/149 (92.62%; unweighted three-class Cohen’s $\kappa = 0.879$, confusion matrix in Appendix L); fine seven-category Phase-B agreement fell to 74/149 (49.66%, $\kappa = 0.368$). The drop is itself the finding: deciding whether a disagreement is due to request evidence, presence policy, interface specification, equivalence, or the evaluator is far less stable than the coarse three-way judgment.

Table 2 states that contrast; the category shares it partitions are in Appendix G, where their intervals account for the stratified unit sample but add no clustering for repeated tasks or duplicate evidence patterns. This is an audit of the earlier run: its 1,463-unit population was frozen before the main run’s condition set was generated and differs by four rows from the 1,467 of Table 1b. We neither transfer case labels to the main run nor interpret the four-row difference.

The defensible inference is about stability rather than prevalence. Under this protocol, coarse concern detection is substantially more consistent across the two isolated passes than fine attribution, and the low Phase-B agreement does not by itself distinguish among taxonomy overlap, limitations of the coding passes, and genuine evidential under-determination. The weighted shares estimate how this AI-only protocol partitions the frozen 1,463-rejection population; they do not estimate human semantic prevalence or ground truth.

In those shares the optional/default category is second-largest at 26.63%, which motivates the separation drawn in Section 2: deciding whether a field exists is not the same construct as deciding which operation produces its value. *If* the adjudicated labels were correct and the categories exclusive, those cases would lie outside an operation-only vocabulary’s scope; both premises are uncertain, so this is a sensitivity calculation, not a limit

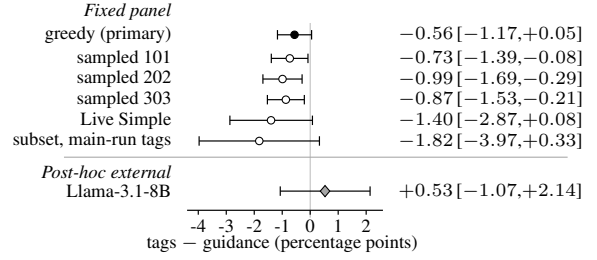


Figure 2: Primary and sensitivity contrasts with 95% task-cluster intervals. Filled circle: the pre-specified greedy estimate. Open circles: exploratory re-estimates on the same fixed panel. Diamond: the post-hoc out-of-panel checkpoint. Rows differ in decoding policy, task population, or checkpoint and are not exchangeable replications.

on what $\mathcal{O}$ could repair. In particular, the 2.61% interface-underspecified share is one protocol’s assignment under an unstable fine taxonomy, not an estimate of how prevalent missing binding policy is. The low Phase-B agreement argues against reporting any single “binding failure” percentage without adjudicated or human evidence. It also sets up RQ3 as a targeted test of one representation rather than an attempted repair of the entire residual.

6 RQ3: do schema-derived hints help?

The pre-specified primary comparison is negative in direction and does not establish improvement. Schema tags attain 4,438/6,555 (67.70%) AST acceptance versus 4,475/6,555 (68.27%) for the same guidance without field tags: -0.56 percentage points, with a 95% task-cluster interval of $[-1.17, +0.05]$. The paired ledger contains 186 fixes, 223 breaks, 4,252 unchanged accepted calls, and 1,894 unchanged rejected calls; Table 3 in Appendix D gives all five condition cells. The prompts are not length-matched: the tags add 441 characters on average (113 input tokens), so the contrast measures content, length, and position jointly.

The exploratory controls caution against a single mechanism story. Tags trail shuffled tags by 0.29 points and guidance trails schema control by 1.54; in these observed cells, which are unmatched prompt modifications, adding operation language is associated with lower acceptance. The shuffle is a semantic perturbation rather than a placebo, and is inactive for 83 of 1,311 tasks whose homogeneous tag arrays make rotation leave the prompt identical.

Alternative labels do not close the gap. The follow-up, on an outcome-informed 121-task Live Simple subset, added a heuristic arm that tags top-level fields only (the main run also tagged nested fields) and then replaced those tags with AI-adjudicated ones on the same fields. On this subset, the main-run tags trail guidance by 1.82 percentage points (Figure 2). The top-level heuristic trails guidance by 2.48 points ($[-4.46, -0.50]$); the AI-adjudicated tags gain $+0.83$ against the top-level heuristic but still trail guidance by 1.65. Every tested unshuffled tag source has a lower observed acceptance rate than no-tag guidance on this subset. The main-run shuffled tags score above guidance (430 versus 424 accepted calls), whereas AI-shuffled tags score slightly below it (423); none of these comparisons evaluates authored gold labels (Appendix I).

6.1 Where turnover occurs

The two conditions differ only in the attached metadata, so for each turnover case we can ask which part of the emitted call changed. We compare the two generated calls target-blind, without reading the benchmark answer: the function changed, the emitted key set changed (*presence*), the keys matched but a value differed (*value*), or both. The decomposition is exploratory and post hoc, and describes which cells turned over rather than a cause.

Table 8 (Appendix F) localizes the negative result. On the dimension the operation vocabulary encodes (how a value is constructed for a given field), turnover is nearly balanced: 77 fixes against 78 breaks, a net of -1 . The net imbalance sits instead in changes to the emitted key set. Presence-only changes account for 72 fixes and 103 breaks (net -31), with dropped fields alone at a net -33 ; including the cases that change both presence and a value, key-set changes account for 88 fixes and 120 breaks (net -32). This is localization, not attribution: the comparison is between two emitted calls and identifies where outcomes moved, not why the metadata changed emission.

The dropped fields are not one phenomenon (Appendix F). Counted as field-level events (one call can drop several fields), 125 top-level fields were dropped. Twenty-one are *required* by the schema and every one is a break, a net -21 : presence was already specified, so adherence to a stated constraint fell. A further 94 are *optional with a schema default* (37 fixes, 57 breaks, net

-20) and none was optional without one; only this group leaves the omit-or-emit-default choice open, since required-field presence is itself part of Π and the schema had fixed it. The remaining 10 dropped events are undocumented fields and are all fixes.

One pair of tasks shows both directions arising from a single tag. Two Live Simple tasks share an opening location-format cue, “in the format of ‘City, State (abbr)’,” and receive the same coarse signature $\{\text{NORMALIZE}\}$ from the heuristic, although their complete descriptions differ. Asked to *find hotels in San Diego*, guidance emits “San Diego” and the tagged condition emits “San Diego, CA”, which BFCL accepts (a fix). Asked for *weather in Hyderabad in Fahrenheit*, guidance emits the accepted “Hyderabad, India” while the tagged condition emits “Hyderabad, CA” (a break). The behavior is consistent with the tag raising the salience of the documented format cue; whether that helps depends on a target format and a region policy that the bare label NORMALIZE does not carry: an operation name without its parameters can raise salience without determining the output.

Figure 2 collects the primary and sensitivity contrasts. The contrast is also negative in each of three sampled draws on the same panel (-0.73 , -0.99 , -0.87 , all excluding zero) under a different decoding policy. One exploratory post-hoc out-of-panel checkpoint, Llama-3.1-8B, has a positive point estimate ($+0.53$, $[-1.07, +2.14]$; 61 fixes, 54 breaks); the task-paired difference between the checkpoint effect and the fixed-panel mean effect is $+1.10$ points with a 95% interval of $[-0.59, +2.88]$. Neither evaluated contrast establishes improvement: the primary fixed-panel estimate is negative in direction while the post-hoc external estimate is positive and imprecise, and neither supports a general degradation claim or a model interaction (Appendices D and K).

The result therefore does not support the simple story that re-labeling ordinary schema cues as operation names improves aggregate AST acceptance on this panel. It does not show that an independently authored contract P_f , with explicit parameters and a stated presence policy, would fail, nor does it rule out a learned recovery method. Nor do the 186 fixes license a positive method claim: the 223 breaks are part of the same intervention.

7 Discussion

Read together, the three results say something narrower and more useful than “metadata does not help”. The primary comparison does not establish a benefit from a specific move: taking cues the schema already contains and re-presenting them as operation names. Observation 1 delimits that move: a schema-only proxy cannot supply distinctions absent from its input. RQ1 establishes the operational mismatch that motivates the test, and RQ2 shows why no single operation vocabulary should be read as a repair for the whole residual. The turnover decomposition then separates what the schema already constrained from what it did not; the optional-with-default cases raise questions about emission policy, but the decomposition does not establish that missing policy caused the changes. A useful authored interface would therefore need more than operation names: target formats and units, alias sets, query or composition grammars, deterministic derivation rules, and an explicit presence/default policy. Whether supplying them helps is a question about information not fully represented by the tested coarse labels, and in some cases possibly not exposed by the schema at all; that is what this study could not test.

8 Related work

Argument filling has a lineage we do not claim to displace. Dialog state tracking already distinguishes extracting a span from choosing from a candidate list (Zhang et al., 2020), a distinction our COPY and SELECT classes inherit, and LLM-based API argument filling needs an explicit grounding step beyond the API specification (Mok et al., 2024). What we add is not that arguments must be grounded, but a proposal to publish the permitted transformations at field granularity in the interface itself, with presence treated separately.

The closest methods differ from our direction on one axis: where the interface information comes from. JTPRO and Trace-Free+ *infer* interface text from execution evidence, by optimizing instructions from labeled traces and by transferring supervision from trace-rich to trace-free settings (Ghoshal et al., 2026; Guo et al., 2026), whereas $\hat{B}_f$ uses no learned knowledge; RQ3 therefore bounds a transparent re-encoding and neither contradicts Trace-Free+ nor shows that schema-only recovery fails in general. PACTool adapts schema naming to model familiarity

(Lee et al., 2026), where we would instead preserve authoritative semantics and expose permitted transformations; PACT constrains *which evidence source* may control an argument (Fan et al., 2026), close to our $\mathcal{E}$ component, with its focus on evidence provenance rather than on parameterized request-to-value transformations; Contract2Tool learns *tool-level* contracts of preconditions, effects, risk, and cost (Suresh Babu and Ganesh Iyer, 2026), where our object is field-level request-to-value binding; and IFEval-FC places checkable instructions in parameter descriptions (Skripko, 2025), where unlike our estimate the intended constraint is explicit. Failure taxonomies and process-level methods localize registry, structure, and value errors (Garimella et al., 2026; Kokane et al., 2025; Duan et al., 2026; Lin et al., 2026; Cheng et al., 2026), and Mem2ActBench tests whether long-term memory is applied to tool selection and parameter grounding (Shen et al., 2026); interface-organization work alters schema or tool structure (Sigdel and Baral, 2026; Wu et al., 2025); and constrained generation guarantees legal form while potentially altering task behavior (Geng et al., 2025; Tam et al., 2024). Appendix E compares these along information source, representation, and contract components. Our question is prior to method choice in each case: whether the ordinary interface makes the permitted transformation inspectable at all.

9 Conclusion

Not in general. BFCL-style exposed schemas and BFCL’s accepted-answer semantics do not induce the same first-call decisions: a correct-function partial structural diagnostic and AST acceptance disagree in both directions across the fixed panel, on every checkpoint, and again on a post-hoc out-of-panel one.

Re-exposing the schema’s own cues as five operation labels changes 409 paired outcomes but establishes no aggregate improvement on that panel. Fine-grained attribution of the residual is unstable under our AI-only protocol, and the post-hoc external checkpoint reverses the intervention’s point-estimate sign. These findings estimate neither the prevalence of missing binding policy nor the value of supplying it; they isolate the next experiment: independently authored, parameterized contracts with explicit presence, evidence, transformation, and validation rules.

Limitations

We study one local first call, not environment state, recovery, side effects, or long-horizon behavior. BFCL Live is public and may occur in pretraining, its filtered requests are not deployment traffic, and our universal JSON prompts differ from BFCL’s provider-specific handlers, so these are BFCL-Live-derived results under official scoring rather than leaderboard scores. Neither criterion is an execution oracle.

The panel is purposive and family-correlated and its largest checkpoint is 14B. Our task instructions are in English, but the retained BFCL requests include other languages (Portuguese, Chinese, Korean, and Thai requests are among the 258 Live Simple cases), and we conduct no language-stratified evaluation. The primary analysis uses greedy decoding; Appendix D reports three sampled draws. The intervals resample tasks conditional on the fixed panel and quantify no uncertainty over model families, decoding, or interface dependence. Greedy decoding is not bit-reproducible across the runtime stack: replay changed 49 of 6,555 generations and 7 AST outcomes. The operation vocabulary is non-exhaustive and unvalidated as authored metadata, and every tag we test is a schema-only estimate. The turnover decomposition is post hoc and checkpoint-specific for 282 of 339 affected tasks, so it may reflect prompting sensitivity rather than an interface-level failure mode.

RQ2’s labels come from two isolated AI coding passes rather than human annotators (Section 5), so human double-coding remains a precondition for any prevalence claim. The passes used GPT-5-based agents without an immutable snapshot, so their outputs are not bit-reproducible (Appendix L). No generated call was executed against an external tool.

Acknowledgments

We used large language model assistants for literature discovery, for developing and reviewing experimental, analysis, and verification code, for auditing the analysis pipeline, and for drafting and revising the manuscript. Their use for annotation and adjudication is described in Section 5 and Appendix L. The authors checked the study design, claims, reported results, and final text and take responsibility for the work.

References

- Aijia Cheng, Kailong Wang, Ling Shi, and Yongxin Zhao. 2026. [R2IF: Aligning reasoning with decisions via composite rewards for interpretable LLM function calling](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36995–37008, San Diego, California, United States. Association for Computational Linguistics.
- Hongtao Duan, Lu Jiang, Mingyong Zhang, Xiaobing Zhu, Tianpeng Bu, Hao Jiang, Xinyu Wei, and Lulu Hu. 2026. [MemTR: Enhancing tool-calling reliability via uncertainty-triggered FFN-space retracing](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 19476–19493, San Diego, California, United States. Association for Computational Linguistics.
- Linfeng Fan, Ziwei Li, Yuan Tian, Yichen Wang, Rongsheng Li, and Xiong Wang. 2026. [The granularity mismatch in agent security: Argument-level provenance solves enforcement and isolates the LLM reasoning bottleneck](#). *Preprint*, arXiv:2605.11039.
- Ritvik Garimella, Vedant Khandelwal, Anvi Kohli, and Amit Sheth. 2026. [SAAG: Structured agent assessment and grounding](#). *Preprint*, arXiv:2607.18245.
- Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. 2025. [JSON-SchemaBench: A rigorous benchmark of structured outputs for language models](#). *Preprint*, arXiv:2501.10868.
- Sandip Ghoshal, Anshul Mittal, Jyotika Singh, Miguel Ballesteros, Weiyi Sun, Fang Tu, Shailender Singh, Yassine Benajiba, Fahad Shah, Sujeth Bharadwaj, Sujith Ravi, and Dan Roth. 2026. [JTPRO: A joint tool-prompt reflective optimization framework for language agents](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 40573–40595, San Diego, California, United States. Association for Computational Linguistics.
- Aaron Grattafiori et al. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Ruocheng Guo, Kaiwen Dong, Xiang Gao, and Kamalika Das. 2026. [Learning to rewrite tool descriptions for reliable LLM-agent tool use](#). *Preprint*, arXiv:2602.20426.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7B](#). *Preprint*, arXiv:2310.06825.
- Shirley Kokane, Ming Zhu, Tulika Awalgaoonkar, Jianguo Zhang, Akshara Prabhakar, Thai Hoang, Zuxin

- Liu, Rithesh Murthy, Liangwei Yang, Weiran Yao, Juntao Tan, Zhiwei Liu, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, Caiming Xiong, and Silvio Savarese. 2025. [ToolScan: A benchmark for characterizing errors in tool-use LLMs](#). In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, Singapore. Version 2 of arXiv:2411.13547; version 1 (November 2024) appeared under the title SpecTool.
- Jonggeun Lee, Woojung Song, Jongwook Han, Hae-sung Pyun, and Yohan Jo. 2026. [Don’t adapt small language models for tools; adapt tool schemas to the models](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20695–20719, San Diego, California, United States. Association for Computational Linguistics.
- Jianghao Lin, Yuanyuan Shi, Xin Peng, Renjie Ding, Hairui Wang, Yuxuan Peng, Bizhe Bai, Weixi Song, Fengshuo Bai, Huacan Chai, Weinan Zhang, Fei Huang, and Ying Wen. 2026. [ToolPRM: Fine-grained inference scaling of structured outputs for function calling](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18792–18804, San Diego, California, United States. Association for Computational Linguistics.
- Meta AI. 2024. [Llama-3.1-8B-Instruct](#). Hugging Face model repository, revision 0e9e39f249a1.
- Mistral AI. 2024. [Mistral-7B-Instruct-v0.3](#). Hugging Face model repository, revision c170c708c41d.
- Jisoo Mok, Mohammad Kachuee, Shuyang Dai, Shayan Ray, Tara Taghavi, and Sungroh Yoon. 2024. [LLM-based frameworks for API argument filling in task-oriented conversational systems](#). *Preprint*, arXiv:2407.12016.
- OpenAPI Initiative. 2024. [OpenAPI Specification v3.1.1](#). OpenAPI Initiative specification.
- Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. [The Berkeley function calling leaderboard \(BFCL\): From tool use to agentic evaluation of large language models](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 48371–48392. PMLR.
- Qwen Team. 2024. [Qwen2.5-14B-Instruct](#). Hugging Face model repository, revision cf98f3b3bbb4.
- Qwen Team. 2025a. [Qwen3-0.6B](#). Hugging Face model repository, revision c1899de289a0.
- Qwen Team. 2025b. [Qwen3-1.7B](#). Hugging Face model repository, revision 70d244cc86cc.
- Qwen Team. 2025c. [Qwen3-4B-Instruct-2507](#). Hugging Face model repository, revision cdbee75f17c0.
- Yiting Shen, Kun Li, Wei Zhou, and Songlin Hu. 2026. [Mem2ActBench: A benchmark for evaluating long-term memory utilization in task-oriented autonomous agents](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8173–8190, San Diego, California, United States. Association for Computational Linguistics.
- Akshey Sigdel and Rista Baral. 2026. [Schema first tool APIs for LLM agents: A controlled study of tool misuse, recovery, and budgeted performance](#). *Preprint*, arXiv:2603.13404.
- Nikolai Skripko. 2025. [Instruction-following evaluation in function calling for large language models](#). *Preprint*, arXiv:2509.18420.
- Rahul Suresh Babu and Laxmipriya Ganesh Iyer. 2026. [Contract2Tool: Learning preconditions and effects for reliable tool-augmented LLM agents](#). *Preprint*, arXiv:2606.07904.
- Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. [Let me speak freely? A study on the impact of format restrictions on large language model performance](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.
- Austin Wright, Henry Andrews, and Ben Hutton. 2022a. [JSON Schema Validation: A vocabulary for structural validation of JSON](#). IETF Internet-Draft draft-bhutton-json-schema-validation-01, dialect 2020-12.
- Austin Wright, Henry Andrews, Ben Hutton, and Greg Dennis. 2022b. [JSON Schema: A media type for describing JSON documents](#). IETF Internet-Draft draft-bhutton-json-schema-01, dialect 2020-12.
- Yunnan Wu, Qile P. Chen, Deshank Baranwal, Jinlong Zhou, and Jian Yuan. 2025. [The art of tool interface design](#). In *Proceedings of the 1st Workshop for Research on Agent Language Models (REALM 2025)*, pages 63–79, Vienna, Austria. Association for Computational Linguistics.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. [Find or classify? Dual strategy for slot-value predictions on multi-domain dialog state tracking](#). In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

A Prompt summary

This section paraphrases the universal prefix and quotes the operation vocabulary verbatim. A single deterministic renderer produced every prompt, and its hash is recorded for each of the 32,775 cells.

Every condition begins with instructions to initialize exactly one call, return exactly one JSON object with tool and arguments, emit no other text or calls, select only a listed function, use documented keys, and respect required fields, nested structure, primitive types, enums, defaults, and units in descriptions. The four operation-aware conditions then use the following closed vocabulary:

```
COPY reproduces explicit request
evidence without semantic
transformation.
SELECT chooses a value from an
explicitly licensed closed set.
NORMALIZE applies only an explicitly
licensed format, case, representation,
date, time, or unit transformation.
COMPOSE constructs a list, object,
query expression, joined value, or
nested structure from multiple licensed
pieces.
DERIVE computes a value only through an
explicitly documented deterministic
rule.
```

The prompt says that operations may be combined only when documentation licenses every step and that tags, examples, defaults, distractors, and plausible conventions do not license unsupported information. It then prints the condition-specific compact schema registry, conversation turns, and the literal marker `JSON call:`. The vocabulary is closed.

The tagger defaults enums and Booleans to SELECT, containers to COMPOSE, numerics to NORMALIZE, and remaining fields to COPY. Frozen field-description phrases can add or replace classes, including DERIVE only for the heuristic’s documented-rule cues. These mechanical assignments are hypotheses, not field-level gold labels.

B Checkpoints and runtime

All checkpoints are the public Hugging Face repositories at pinned revisions, loaded with the tokenizer and chat template of the same revision, in bfloat16 under PyTorch 2.5.1 and Transformers 5.12.1, with greedy decoding, seed 101, and a 256-new-token cap: Qwen3-0.6B at c1899de289a0 (Qwen Team, 2025a), Qwen3-1.7B at 70d244cc86cc (Qwen Team, 2025b), Qwen3-4B-Instruct-2507 at cdbee75f17c0 (Qwen Team, 2025c), Mistral-7B-Instruct-v0.3 at c170c708c41d (Mistral AI, 2024), and Qwen2.5-14B-Instruct at cf98f3b3bbb4 (Qwen Team, 2024). Thinking mode is disabled through the chat template for the three Qwen3 checkpoints. The post-hoc out-of-panel checkpoint (Appendix K) is Llama-3.1-8B-Instruct (Grattafiori et al., 2024) at 0e9e39f249a1 (Meta AI, 2024), run under the same settings. Benchmark data and the official scorer are pinned to gorilla commit f7cf7359b7ac with bfcl-eval 2025.12.17. The heuristic tagger, the prompt renderer, and the row ledgers are retained by the authors and are not released with this paper.

C Controlled motivation

The earlier controlled harness contains 40 development requests, 40 held-out requests over the same ten interfaces, a 24-request surface-validation set of fresh requests over those same interfaces, 24 requests from ten unseen tool families, 24 parser-stress requests, and a 30-request human-written slice. Across seven small open models, adding full schemas raises literal exact-reference agreement over a vanilla tool description by 22.6 percentage points on both development and held-out and by 24.0 points on unseen families. Yet on development, 89.1% select the correct function and pass the harness’s partial diagnostic while only 71.0% equal the literal reference.

Those positive deltas are not the paper’s external headline. A model-free request parser solves 103/104 (99.0%) of the pooled development, held-out, and surface-validation cases (40/40, 40/40, and 23/24), versus 2/24 (8.3%) parser-stress and 4/30 (13.3%) human-written cases. This bounds the controlled suites as mechanism probes rather than evidence of broad tool-use reasoning.

Condition	Accepted	Total	AST %
Schema control	4,576	6,555	69.81
Guidance	4,475	6,555	68.27
Placeholder metadata	4,470	6,555	68.19
Schema-derived tags	4,438	6,555	67.70
Within-registry shuffled tags	4,457	6,555	67.99

Table 3: All five condition cells. Only schema tags minus guidance is the pre-specified primary contrast; the others are exploratory.

Omitted checkpoint	Fixes	Breaks	Δ pp [95% CI]
Qwen3-0.6B	140	175	-0.67 [-1.34, 0.00]
Qwen3-1.7B	152	177	-0.48 [-1.14, +0.23]
Qwen3-4B	156	198	-0.80 [-1.53, -0.10]
Mistral-7B	142	162	-0.38 [-1.05, +0.27]
Qwen2.5-14B	154	180	-0.50 [-1.18, +0.19]

Table 4: Exploratory leave-one-model-out sensitivity for schema tags minus guidance. Intervals retain all remaining fixed models per sampled task and are unadjusted for multiplicity.

D Primary turnover and robustness

Table 4 reports leave-one-model-out sensitivity and Table 5 the primary contrast by checkpoint. Alternative dependence analyses give -0.56 points with exact-registry clusters (95% CI $[-1.26, +0.12]$) and structural-schema clusters ($[-1.29, +0.15]$). Retaining one task per exact registry gives -0.44 ($[-1.43, +0.55]$); retaining one per structural schema gives -0.73 ($[-1.80, +0.34]$). These are sensitivity analyses, not replacements for the pre-specified task-cluster estimand.

The primary effect is -1.40 points on Live Simple (31 fixes, 49 breaks; 95% CI $[-2.87, +0.08]$) and -0.36 points on Live Multiple (155 fixes, 174 breaks; the complete combined ledger remains the inferential target). Registry size slices are unstable at small counts: the 10+ bin has only 16 tasks. We therefore do not infer a registry-size trend.

Replay drift. The schema-control prompts are byte-identical to those of an earlier execution of the same condition (0 prompt-hash mismatches over 6,555 units), so the pair measures observed run-to-run variation under nominally greedy decoding on this stack; a single replay does not bound that variation. Outputs are not fully identical: 49 generations differ and 7 AST outcomes flip (2 gained, 5 lost; net -3). The drift is confined to two checkpoints, Qwen3-0.6B (32 differing generations, 5 flips) and Qwen3-4B (17, 2), while

Checkpoint	Guidance	Tags	Δ (pp)
Qwen3-0.6B	46.76	46.61	-0.15
Qwen3-1.7B	67.43	66.51	-0.92
Qwen3-4B	76.74	77.12	+0.38
Mistral-7B	70.56	69.26	-1.30
Qwen2.5-14B	79.86	79.02	-0.84
Fixed panel	68.27	67.70	-0.56

Table 5: Primary contrast by checkpoint. These fixed-panel decompositions are exploratory; the related Qwen3 checkpoints are not independent model samples.

Qwen3-1.7B, Mistral-7B, and Qwen2.5-14B replay bit-identically. This is an order of magnitude smaller than the primary net of -37 ; we report it rather than call the stack deterministic, and the primary interval does not model this variation.

Generation cap. Thirty-six of the 32,775 cells reach the 256-new-token cap. Nine primary model-task pairs contain at least one capped cell, and all nine are rejected under both guidance and tags, so no observed primary fix or break involves a capped generation. We did not rerun with a larger cap, so the counterfactual effect of removing it is not measured.

Stochastic decoding. The primary analysis uses one greedy generation per prompt, so its interval does not model decoding variance. We examine sensitivity to decoding policy by re-running the two primary conditions on the same five checkpoints and 1,311 tasks under nucleus sampling (temperature 0.7, top- p 0.9) for three independent seeds: 39,330 additional generations. Both conditions share decoding settings and seed within a draw, so each draw is still a matched paired comparison.

All three sampled differences are negative, and unlike the greedy interval all three exclude zero (Figure 2). The contrast is therefore also negative under this sampled policy on the fixed panel; three seeds do not model the full decoding distribution, and the post-hoc out-of-panel checkpoint has the opposite point-estimate sign (Appendix K), so we read these sampled draws as evidence about the fixed panel rather than about the encoding in general. We do not promote them over the pre-specified estimand. The sampled runs apply a different decoding policy from the greedy estimand, their conditions were not re-frozen, and the greedy estimate lies just outside the sampled range, so the

Work	Information source	Interface representation	Contract components
JSON Schema / OpenAPI (Wright et al., 2022b,a; OpenAPI Initiative, 2024)	interface author	typed schema metadata	partial Π , Θ , V_P
Slot filling / DST (Zhang et al., 2020)	labeled dialogues	model-internal	G (find vs. select)
API argument filling (Mok et al., 2024)	labeled dialogues	prompt or training	$\mathcal{E}$ (grounding step)
PA-Tool (Lee et al., 2026)	model preference	rewritten names	naming only
JTPRO (Ghoshal et al., 2026)	rollout traces	free-text instructions	implicit in free text
Trace-Free+ (Guo et al., 2026)	trace-derived transfer	rewritten descriptions	implicit in free text
IFEval-FC (Skripko, 2025)	authored	description text	partial V_P (format checks)
SAAG (Garimella et al., 2026)	staged diagnosis	feedback messages	diagnosis, not specification
PACT (Fan et al., 2026)	provenance labels	argument-level trust policy	$\mathcal{E}$ (evidence provenance)
Contract2Tool (Suresh Babu and Ganesh Iyer, 2026)	docs and traces	tool-level contract	pre/effects, risk, cost
This paper, $\hat{B}_f$	the schema itself	five coarse labels	estimate of G 's labels only
Proposed contract P_f	interface author	typed and inspectable	Π , $\mathcal{E}$, G , Θ , V_P (design)

Table 6: The proposed direction is distinguished less by adding text to an interface than by where the information originates and which contract components it makes explicit. Entries name components where a work states them; several works carry related information implicitly, and the absence of a component marks a different focus, not a defect. The last row is a design proposal; this paper evaluates only the row above it.

two are not interchangeable estimates of one quantity.

E Positioning against related work

Table 6 places the closest work on the axes that matter here. Unlike learned or trace-optimized interface rewrites, the direction we propose is an author-supplied, field-level, machine-inspectable contract over evidence, transformation, and presence. This paper does not validate that contract. It shows why a schema-only projection of it is inadequate, and supplies the audit needed to test the contract cleanly.

F Presence turnover by schema status

Table 9 splits every top-level key that appears or disappears between the guidance and schema-tag conditions by its status in the ordinary schema, restricted to turnover cases whose function selection is unchanged.

Reading against the accepted answer sharpens the optional group: adding an optional field BFCL would accept omitted breaks 70 times against 17 fixes, while dropping one fixes 36 times against 14 breaks. The encoding pushes models to populate default-bearing optional fields the benchmark prefers left out. This describes where outcomes moved; it does not establish that a missing presence policy caused the movement. Appendix J further decomposes turnover by operation label, by checkpoint, and by prompt burden; notably, turnover is checkpoint-specific for 282 of 339 affected tasks.

Checkpoint	$\tilde{C}$ %	AST %	Gap (pp)
Qwen3-0.6B	85.13	48.44	36.69
Qwen3-1.7B	91.69	69.87	21.82
Qwen3-4B	94.97	77.73	17.24
Mistral-7B	91.91	72.62	19.30
Qwen2.5-14B	93.67	80.40	13.27
Fixed panel	91.47	69.81	21.66

Table 7: The structural-minus-AST gap under schema control holds for every checkpoint, so the aggregate result is not an artifact of the smallest model. Each checkpoint contributes 1,311 units.

Change in the emitted call	Fixes	Breaks	Net
Value only	77	78	-1
Presence: field dropped	16	49	-33
Presence: field added	42	42	0
Presence: added and dropped	14	12	+2
Presence and value	16	17	-1
Function selection changed	21	22	-1
Unparsed call shape	0	3	-3
Total	186	223	-37

Table 8: Target-blind decomposition of primary turnover. On the value dimension the operation vocabulary encodes, fixes and breaks are nearly balanced. The net imbalance sits in key-set changes, which Section 6.1 and Table 9 separate into a required-field component the schema already constrained and an optional-with-default component whose emission policy it does not fully determine.

Field-presence event	Fixes	Breaks	Net
Dropped, optional with default	37	57	-20
Dropped, required by schema	0	21	-21
Dropped, undocumented	10	0	+10
Added, optional with default	68	79	-11
Added, required by schema	5	0	+5
Added, undocumented	0	10	-10

Table 9: Top-level key changes on primary turnover cases with unchanged function selection, by schema status. Rows count field-change events; one call can contribute several events, so totals need not match call-level counts. Two distinct patterns are visible: required-field drops, which the schema already constrained, and optional-with-default changes, which it does not.

G AI-protocol category shares

Table 10 gives the design-weighted shares discussed in Section 5.

AI-protocol assigned category	Share (%)	95% interval
Clear request evidence/model error	31.40	22.89–39.91
Optional/default presence	26.63	22.08–31.17
Benchmark stricter than interface	14.78	7.48–22.08
Request ambiguity	13.98	7.60–20.35
Evaluator/checker issue	5.87	1.61–10.13
Plausible semantic equivalence	4.74	1.19–8.30
Interface underspecified	2.61	0.00–5.75

Table 10: How this AI protocol partitioned the residual. The fine taxonomy carries $\kappa = 0.368$ (Table 2), so these shares describe one protocol’s assignments and estimate neither human prevalence nor semantic ground truth. Design-weighted over 149 evidence cases representing 160 sampled units from an earlier frozen population of 1,463 post-diagnostic AST rejections.

H Reverse-cell taxonomy

Table 11 categorizes all 50 schema-control calls that official AST accepts while the structural diagnostic rejects from the diagnostic’s own recorded complaints, each case assigned once by first match in a fixed priority order. Every case selects the correct function, so the rejection is always structural.

Cause	<i>n</i>	%
<i>BFCL accepts beyond the documented interface</i>		
Enum value outside the documented enum	30	60.0
Null where the schema declares a primitive	16	32.0
<i>False negative of our closed-object reading</i>		
Undocumented nested key	4	8.0

Table 11: Causes of the reverse cell. Only the first group shows BFCL’s accepted answers exceeding the documented interface; the four nested-key cases are a limitation of our diagnostic, not benchmark permissiveness.

The cell is spread across checkpoints rather than produced by one model: 4, 11, 6, 16, and 13 cases for Qwen3-0.6B, Qwen3-1.7B, Qwen3-4B, Qwen2.5-14B, and Mistral-7B, respectively. The null-valued cases are the clearest: the model emits null for an optional field, our diagnostic enforces the declared primitive type, and the checker treats the null as an omission. The four undocumented-key cases are all nested and all arise from a single interface whose data property documents a placeholder and instructs callers to “replace with actual parameter names.” Emitting concrete keys there follows the documentation, so our closed-object reading, not BFCL, is what rejects them. Excluding these four leaves 46 reverse cases, which is what the two-sided claim rests on.

I AI-coded tag subset

Table 12 lists every condition available on the 121-task Live Simple subset used by the outcome-informed extension, over 605 model-task units each. The follow-up package reuses the main run’s guidance rows for these tasks, so all conditions are paired on identical units, checkpoints, seed, and decoding.

Condition	Accepted	AST %
Schema control	432	71.40
Guidance	424	70.08
Placeholder metadata	420	69.42
Shuffled tags (main run)	430	71.07
Schema-derived tags (main run)	413	68.26
Heuristic tags (follow-up, top-level)	409	67.60
AI-adjudicated tags	414	68.43
AI-shuffled tags	423	69.92

Table 12: All conditions on the 121-task subset, 605 units each. Exploratory: the follow-up arms were specified after the main-run outcomes were known.

The paired contrasts, with 20,000 task-cluster draws, are AI-adjudicated minus guidance -1.65 ($[-3.64, +0.17]$; 10 fixes, 20 breaks), heuristic minus guidance -2.48 ($[-4.46, -0.50]$), AI minus heuristic $+0.83$ ($[-0.33, +1.98]$), AI minus placeholder -0.99 ($[-2.81, +0.83]$), and AI minus AI-shuffled -1.49 ($[-3.14, -0.17]$). The follow-up’s heuristic arm differs from the main-run tags in one respect: it attaches the same frozen heuristic to the 132 top-level fields only, whereas the main run also tagged nested fields, so the two prompts differ only on the nine subset tasks with nested fields; the arms are separate generation runs. Two cautions follow. Every tested unshuffled tag source has a lower observed acceptance rate than the no-tag guidance condition on this subset, while the main-run shuffled tags (430/605) sit above it: this alternate mapping did not remove the gap, and without authored gold labels the comparison cannot rule out a higher-quality label source. And shuffled tags outscore the unshuffled ones in both the main run and the follow-up here, which argues against reading the effect as a response to the semantic content of the labels; with 121 tasks the intervals are wide and we do not push this further. The shuffle is inactive for 28/121 tasks in the main run and 13/121 in the follow-up.

J Exploratory decompositions

All three decompositions below are exploratory and unadjusted for multiplicity.

By operation label. Labelling each changed field by the signature the heuristic assigned it (Table 13) shows that the aggregate comparison, which established no improvement, is not uniform across the vocabulary. SELECT carries the largest net loss, NORMALIZE a smaller one, and COPY is mildly positive. This is consistent with the presence result: the tagger assigns SELECT to enums and Booleans, and Boolean fields with a declared default are exactly the optional-with-default fields whose emission the tags perturb. The denominator here is turnover, which is selected on outcome, so we report counts only and give no interval.

Label on changed field	Fixes	Breaks	Net
SELECT	68	108	-40
NORMALIZE	63	75	-12
COPY	63	58	+5
Function selection changed	21	22	-1
COMPOSE	12	13	-1
Undocumented field	10	10	0
Multi-label combinations	4	5	-1

Table 13: Turnover by the operation label attached to the field that changed. Rows count changed-field events, so one call can contribute to more than one row. No DERIVE-only case appears: the heuristic assigns it rarely.

Cross-model consistency. Turnover is overwhelmingly checkpoint-specific. Of 339 tasks showing any turnover, 282 do so for exactly one of the five checkpoints; only 18 are fixed in two or more and none in three or more, while 21 are broken in two or more and 2 in three or more. These categories overlap; 29 tasks are fixed for one checkpoint and broken for another. This argues against reading the effect as a reusable interface-level failure mode: on this panel it behaves much more like model-specific prompting sensitivity, a caution we apply to our own mechanism discussion.

Prompt burden. Binning all 6,555 paired units by added input tokens gives -1.43 points for ≤ 40 added tokens ($[-2.73, -0.16]$), -0.30 for 41–80, -0.09 for 81–160, and -0.80 above 160. The effect is not monotone in burden, and the *smallest* burden bin is the most negative, which does not support a simple attention-dilution account. We

do not read this as isolating burden, because the bins are confounded: single-candidate tasks carry both the fewest added tokens and the most negative effect (-1.40 for registry size 1, against -0.04 for sizes 2–3). Added tokens, registry size, and the Simple/Multiple split are not separable on this panel.

K Post-hoc out-of-panel checkpoint

The fixed panel is family-correlated: three Qwen3 checkpoints, one Qwen2.5, and one Mistral. To test whether the results depend on that mix we repeated the schema-control, guidance, and schema-tag conditions on all 1,311 tasks for a Llama-3.1-8B-Instruct checkpoint, a third family, under identical prompts, scorer, decoding, and seed. The checkpoint was chosen after the main outcomes were known, so this is an exploratory external-validity check outside the pre-specified estimand.

The RQ1 criterion gap replicates closely: the same checkpoint gives 92.45% structural acceptance against 71.17% AST, a 21.28-point gap against the fixed panel’s 21.66, with 288 diagnostic-only and 9 AST-only calls. The mismatch therefore reappears at similar magnitude here, which weakens but does not eliminate the concern that it is specific to the panel’s family mix.

The RQ3 contrast does not replicate in direction: guidance accepts 916/1,311 (69.87%) against 923 (70.40%) for tags, $+0.53$ points with 61 fixes and 54 breaks and an interval crossing zero. Neither the panel nor this checkpoint establishes an improvement, so the insufficiency conclusion is unchanged, but a claim that the encoding *degrades* acceptance would not survive this checkpoint. The turnover mechanisms also differ: dropped fields run 15 fixes to 8 breaks here, against 16 to 49 on the panel. The observed direction is therefore not robust across the evaluated model stacks. It is not an established model interaction either: a task-paired bootstrap of the checkpoint effect minus the fixed-panel mean effect gives $+1.10$ points with a 95% interval of $[-0.59, +2.88]$ (10,000 draws, seed 20260728), and the checkpoint was selected after the primary outcomes were known and differs from the panel in tokenizer, template, and run-time stack, which one comparison cannot separate.

L AI-coding provenance

The residual sampling frame crossed model, category, checker family, and registry-size bin. Sin-

gleton strata were censused; other nonempty strata received at least two units before proportional allocation. Exact evidence duplicates were collapsed only after unit sampling, and design weights still sum to the 1,463-unit population. The two isolated coding passes saw no model identity, task ID, stratum, frequency, prior label, or other coder output; Table 14 is the pre-adjudication Phase-A confusion matrix behind the three-class κ .

Pass A \ Pass B	Error	Accept.	Ambig.	Total
Model error	73	3	3	79
Acceptable	1	40	2	43
Ambiguous	1	1	25	27
Total	75	44	30	149

Table 14: Phase-A confusion matrix before adjudication (rows: pass A; columns: pass B; ambiguous: interface or request ambiguous). Agreement 138/149; chance agreement 0.389; unweighted Cohen’s $\kappa = 0.879$.

The completed files were hash-frozen before comparison. A separate AI adjudication resolved all 75 fine-category disagreements, after which an adversarial review flagged 16 boundaries and 15 were revised; the source coder files remained unchanged. Table 10 reports the final protocol output, while the Phase-B $\kappa = 0.368$ is retained as a warning about construct instability rather than promoted as reliability.

Two procedurally isolated AI passes labeled 132 top-level fields in 40 blinded Live Simple schemas (121 rows). Exact-set agreement was 97/132 and mean Jaccard was 0.872. Per-operation κ was 0.732 for COPY, 0.980 for SELECT, 0.933 for NORMALIZE, 0.935 for COMPOSE, and 0 for DERIVE; one pass never used DERIVE. In the new-arm extension frozen before outputs, AI-adjudicated tags achieved 414/605 versus 409/605 for the heuristic ($\Delta = +0.83$ points; 95% task-cluster CI $[-0.33, +1.98]$ from 20,000 draws, seed 20260729; 10 fixes, 5 breaks); exploratory AI minus shuffled was -1.49 points (task-cluster CI $[-3.14, -0.17]$; schema-cluster CI $[-4.22, +0.78]$). These AI-coded labels neither validate B_f nor replace an authored-signature experiment. The Limitations section’s model-version and reproducibility caveats also apply to these passes.