PLEASE PUT NOTHING HERE: WHEN TOOL DOCUMENTATION HELPS LLMS MAKE THE WRONG CALL

Anonymous authors
Paper under double-blind review

ABSTRACT

In tool interfaces, leaving a field out, sending `null`, sending an empty value, and sending the default can mean different things, depending on each tool's documented rules. A model can therefore send a call that is valid and executes, yet writes a state the user did not ask for. We introduce NothingBench, which pairs synthetic partial-update tasks that differ in one documented rule and scores the exact resulting state by deterministic execution. In *aligned* tasks the most direct call is correct. In *conflict* tasks the rule makes that call write a predictable wrong state. We test whether documenting every input form improves accuracy over stating only the rule the task depends on. Across 144 held-out task families (each an aligned and a conflict version of one task) and five model configurations, a table of equal length that gives the operation for all ten ways of writing a field raises aligned accuracy over a one-line rule but lowers conflict accuracy from 82.2% to 76.7%, below even a schema that gives no rule for omitted, null, empty, or default inputs (82.9%). The difference-in-differences (the table's effect on conflict tasks minus its effect on aligned tasks) is -9.9 points $[-13.2, -6.8]$ and negative for all five models. Accepted calls that reach the predicted wrong state rise from 19 to 77 of 720 conflict responses, while invalid calls fall from 44 to 18 and aggregate accuracy barely moves. Four-row tables recover much of the loss. On four newer models, the table's effect is smaller on average and varies by model. In a small case series on three protocol rules that depart from JSON Merge Patch, a short table scores higher than the one-line rule. Documentation changes should therefore be evaluated on paired conflict tasks and on the states they produce, not only on call validity or aggregate accuracy.

1 INTRODUCTION

Nothing will come of nothing.
— Shakespeare, *King Lear*, I.i

What does it mean to send nothing? Suppose a user tells an assistant, "I don't have a fax number. Record it as none." To update the contact record, the assistant must choose how to write "none": omit the field, send `null`, send an empty value, or send the default. Protocols disagree about what each one does. Under JSON Merge Patch, an omitted member is kept and a member sent as `null` is deleted (Hoffman & Snell, 2014), so `{"fax": null}` erases the entry instead of recording that there is none. Kubernetes Server-Side Apply may reset a field its manager stops sending (Kubernetes Authors). Stripe deletes a metadata key sent as an empty string (Stripe). Protocol buffers and GraphQL differ again (Google, b; GraphQL Foundation, 2021). In a small sample of 40 randomly chosen public `PATCH` operations from one API directory, only one stated what an omitted field does, and a targeted sample of 20 had none (Appendix K). In a related failure, a researcher who registered the license plate `NULL` was sent parking tickets for other vehicles, because a citation database recorded tickets with no plate as `NULL` (Barrett, 2019; droogie, 2019).

No single reading of these forms is right for every API, so a tool-using model must follow the tool's documented rules, which we call its *contract*. A wrong reading can delete a field the user wanted to populate or record an inherited default instead of an explicit setting. The call still parses, validates,

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.

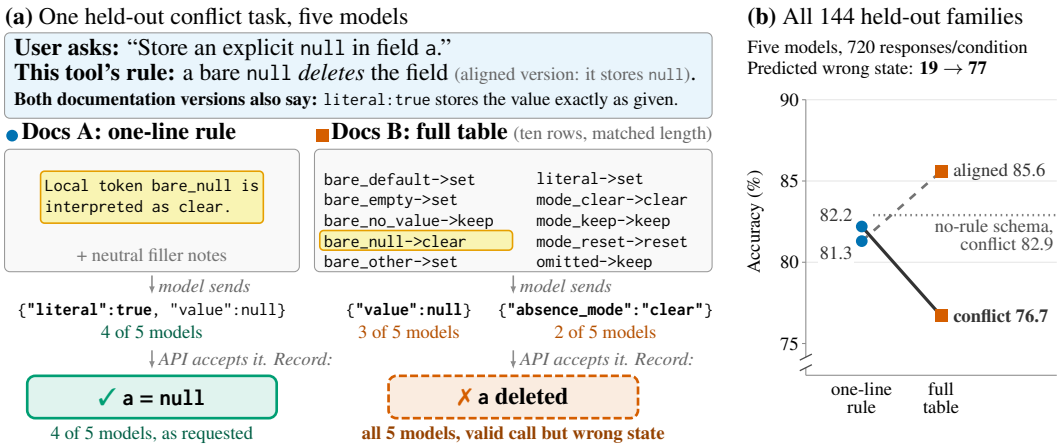

Figure 1: Valid calls, wrong state. **(a)** One held-out conflict task, answered by five models. The user asks to store `null`. This tool's contract says a bare `null` (one sent without `literal:true`) deletes the field. Both documentation versions also say that `literal:true` stores a value exactly as given. The two versions, a one-line rule and a ten-row table, are generated from the same contract and matched in length. Only the table also gives rules for the other five bare or omitted forms. With the one-line rule, four of five models send `literal:true`, and Qwen3-14B deletes the field with `absence_mode:"clear"`. With the table, all five send an accepted call that deletes the field. **(b)** Across all 144 held-out families, the table raises aligned accuracy but lowers conflict accuracy, even below a schema that gives no rule for omitted, null, empty, or default inputs. The accuracy axis starts at 75%. Full prompt in Appendix A.

and executes, so the mistake becomes part of the record, and a workflow that trusts successful execution may act on it later. Metrics that check only whether a call is valid never see the error.

We use NothingBench to test whether documenting every input form helps models choose the right update. Each task has an *aligned* version, in which the most direct call is correct, and a *conflict* version, in which one row of the contract is flipped so that the same call writes a predictable wrong state (Figure 1). In Figure 1, the conflict contract says a bare `null` (one sent without `literal:true`) deletes the field, so a model that sends a bare `null` to store `null` erases the field instead. Models see each task under five schemas generated from the same contract, and our main comparison uses two of them. A deterministic oracle executes every response. The *one-line rule* states only the row the task depends on. The *full table* lists all ten input–operation mappings. Filler text makes the two the same length. Naming only the row this task needs is a best case that documentation written in advance cannot match, so we also compare the table with a schema that states no rule for these inputs.

Findings. On 144 held-out synthetic families and five models: (1) *The full table hurts on conflict tasks.* Replacing the one-line rule with the equally long ten-row table raises aligned accuracy by 4.3 points but lowers conflict accuracy by 5.6 points, to 76.7%. That is below even the 82.9% of a schema that gives no rule for these inputs. The difference-in-differences is -9.9 points $[-13.2, -6.8]$ (§4). (2) *The new errors are valid calls.* Responses reaching the predicted wrong state rise from 19 to 77 of 720 conflict responses, invalid calls fall from 44 to 18, and accuracy over both conditions changes only from 81.7% to 81.1%. (3) *The effect depends on the setting.* Four-row tables recover much of the loss (§5). The table's effect varies across four newer models (§6.1). On three selected protocol rules that depart from JSON Merge Patch, a short table scores higher than the one-line rule (§6.2).

Contributions. We contribute the benchmark, whose records separate an absent field, an explicit value, and an inherited default (§3), the frozen comparison with its error breakdown (§4), and the follow-ups (§5–6).

2 Related work

Tool use and tool documentation. Tool-use benchmarks test API selection, planning, and multi-step execution (Li et al., 2023; Liu et al., 2024a; Maekawa et al., 2026; Patil et al., 2024; Qin et al., 2024). BFCL scores calls by AST and execution (Patil et al., 2025), and IFEval-FC checks instructions embedded in parameter descriptions (Skripko, 2025). Stateful suites such as τ-bench (Yao et al., 2025), ToolSandbox (Lu et al., 2025), and KC-Bench (Lyu et al., 2026) add multi-turn state, policies, and knowledge conflicts. Documentation enables zero-shot tool use (Hsieh et al., 2023). Concise instructions can outperform verbose ones (Yuan et al., 2025), and rewriting documentation from trial feedback can improve performance (Qu et al., 2025). We hold the tool and the requested update fixed and change one documented rule. Because each rule comes in a flipped pair, we know in advance the wrong state that the most direct call reaches, so we can test whether a documentation change makes that error more common.

Absence in interface specifications. Specification languages already treat absence as ambiguous. GraphQL distinguishes an explicit `null` argument from an omitted one and notes that the two may mean, for example, deleting versus not altering a field (GraphQL Foundation, 2021). JSON Schema's `default` is an annotation that validation does not fill in (JSON Schema). Protocol buffers distinguish fields with and without explicit presence (Google, b). Google's API guidelines scope a partial update with a field mask and treat an omitted mask as selecting the populated fields (Google, a; 2019). NothingBench asks whether models honor such a rule when the tool's documentation states it. AbsenceBench (Fu et al., 2025) studies a different kind of absence, whether models notice text omitted from a document.

Priors versus in-context specifications. When context contradicts their parametric knowledge, models can either cling to the prior or adopt the context too readily (Wu et al., 2024a; Xie et al., 2024). They also underperform on counterfactual variants of familiar tasks (McCoy et al., 2024; Wu et al., 2024b), struggle with in-context redefinitions (McKenzie et al., 2023), and override flipped labels only at scale (Wei et al., 2023). In code, documentation of changed APIs competes with memorized usage (Ashik et al., 2026; Liu et al., 2024b). Here, the correct operation follows from the tool's own contract, and the error has one predictable end state. How the contract is presented changes how often models follow it.

Format sensitivity and evaluation design. Meaning-preserving changes to prompt format can move accuracy substantially (He et al., 2024; Sclar et al., 2024), and so can constraints on output format (Tam et al., 2024). Our one-line rule and full table are such a change, but our paired contracts predict which wrong state an error reaches, and the sign of the effect depends on what the user asks for (§4). We build on contrast sets and behavioral tests (Gardner et al., 2020; Ribeiro et al., 2020), judge-free scoring (White et al., 2025; Zhou et al., 2023), and compiled benchmark generation (Li et al., 2025).

3 NothingBench

Tasks and states. Each task asks a model to update a four-field record in one `update_record` call. The executor distinguishes an absent field, an explicit override, and a value inherited from a default (it records which version of the default it inherited). `Set(null)` stores an explicit null, while `Clear` makes the field absent. Likewise, `Set(default)` stores an override, whereas `Reset` records the current default as inherited. These are different target states even when their displayed values coincide.

Explicit forms and bare values. Each field's entry in the call can name its operation through an *explicit form*: `absence_mode` (keep, clear, or reset), or `literal:true` with a value. Or it can pass a *bare* value, whose operation the tool's contract decides. The contract gives each of ten forms an operation, one row per form. Each task family depends on one of these rows, its *rule row*: the row for the omitted field, a bare `null`, a bare empty value, or the bare default value. Appendix B defines all ten forms, including the empty value of each field type.

Aligned, conflict, and displaced. Each request has a most direct call, such as a bare `null` for a request to store null. In the *aligned* contract, the rule row maps that form to the requested operation.

The *conflict* contract flips that one row while holding the request, initial state, and target fixed. Under it, the direct call writes a predictable wrong state, the *displaced state*. A task family is this pair of aligned and conflict tasks. The explicit form remains correct under both contracts (Figure 1 and the schematic in Appendix B). The labels aligned and conflict describe how the rule row relates to the request, not a measured model prior. Merge Patch's `null`→clear, for example, is aligned for a request to remove a value and conflict for a request to store null.

Each of the four rule rows appears with two requests, giving eight *strata* named by form and request, such as *null · store* and *null · remove* (Appendix Table 5). Each stratum has 18 held-out test families and 6 neutral controls whose tasks do not touch the rule row. The three strata requesting storage of null, an empty value, or the default value are the *store tasks*, and the two in which the target field is omitted are the *omission strata*.

Five representations of one contract. Models see every family under five schemas, all generated from the same contract (Appendix B), and our main comparison uses two of them. The *one-line rule* (ONE-LINE) states only the rule row, in one sentence, and gives no rule for the other five bare or omitted forms. The *full table* (FULL) lists all ten surface forms and their operations. Filler text makes these two equal in length to within two characters, both in the documentation and in the full request. Three more schemas serve as reference points. A *no-rule schema* documents the explicit forms but states no rule for bare values, so the model sees the same prompt in both conditions. Only the hidden contract that executes a bare value differs, so its aligned and conflict scores show how the model reads a bare value when told nothing. Two *operation-naming* schemas, a flat operation enum and a typed union, make every entry name its operation, so the contract never has to interpret a bare value.

Scoring. Models answer with one JSON object in text. A fixed parser extracts it without repair, and the oracle executes it on the hidden record. Each response is labeled *correct*, *displaced*, *invalid* (rejected by the contract), *other wrong state*, or *parse failure*. Everything from task generation to analysis is deterministic, and no model judges or repairs another's response.

Design choices. Fictional contracts and field names reduce reliance on memorized API documentation, single-turn pairs isolate the changed rule row, and JSON in text keeps the parser the same across providers. Matching length in characters leaves tokenization and information density free to differ. Every request tells the model to "apply the active mapping exactly", and 1,420 of 1,920 spell out the requested value, as in `(value=null)`.

Estimands and registration. Let $a_{r,c}$ be exact-state accuracy for representation r and condition $c \in \{\mathrm{al}, \mathrm{cf}\}$ (aligned, conflict), averaging over models within each family and then over families. The representation effect is $\Delta_c = a_{\text{FULL},c} - a_{\text{ONE-LINE},c}$. Our main registered contrast is

$$\mathrm{DiD} = \Delta_{\mathrm{cf}} - \Delta_{\mathrm{al}}. \tag{1}$$

A negative DiD means the table's effect is less favorable on conflict tasks, although conflict accuracy itself need not fall. We prefer it to a conflict-only contrast because a representation can also change how often calls are well formed, which moves both conditions. On the development set it was -13.3 $[-19.6, -7.5]$. Our original primary estimand was each representation's aligned-minus-conflict margin, with a success criterion of a 95% lower bound of at least 5 points in at least three models (gate D3 in Table 7). Only Qwen3.8-27B met it. We added the DiD to a seven-contrast Holm family after analyzing the development set and froze it before sending any held-out request (Appendix C). Intervals use 10,000 family-cluster bootstrap resamples and are conditional on the tested model configurations. Everything outside the Holm family, including per-model estimates, is descriptive. The follow-ups in §5–6 were registered after the primary analysis and frozen before any of their own outputs existed, except that the source-grounded series (§6.2) repeats requests already sent to the three API models. Several follow-ups lost one to three responses to provider errors. They are exploratory because their registrations required every response to be scored. Appendix Table 6 lists coverage and status.

Models and protocol. The primary models are Claude Sonnet 4.5, DeepSeek V4.1 Flash, GPT-5.6 Terra, and two locally served checkpoints, Qwen3.8-27B and Qwen3-14B. Each model answers each rendered request (one family, condition, and representation) once. We request temperature 0, at most 1,024 output tokens, and thinking off where the provider allows it. DeepSeek is served through a

third-party gateway, and the Qwen models are pinned checkpoints served locally (Appendix B). The 144 held-out test families and 48 controls give 1,920 rendered requests and 9,600 responses (9,599 scored). A separate 64-family development set was analyzed first, and we froze the held-out set and all estimands before sending any held-out request. Appendix Table 6 lists all ten studies.

4 THE FULL TABLE INCREASES VALID CALLS THAT WRITE THE WRONG STATE

Table 1: Held-out exact-state accuracy (%) on 144 synthetic test families, pooled over five models, with 95% bootstrap intervals. Margin = aligned − conflict. †Descriptive, outside the registered contrasts. Changes use unrounded accuracies. The one-line rule and the full table are the main comparison. The no-rule schema states no mapping, and the enum and union name the operation.

	Accuracy (%)				Aligned − conflict	
Representation	Aligned	95% CI	Conflict	95% CI	Margin	95% CI
One-line rule	81.3	[77.1, 85.1]	82.2	[77.8, 86.3]	−1.0	[−6.8, 5.0]
Full table	85.6	[81.3, 89.6]	**76.7**	[71.4, 81.7]	+8.9	[2.4, 15.3]
No-rule schema†	84.0	[79.7, 88.1]	82.9	[78.3, 87.2]	+1.1	[−4.7, 6.8]
Flat operation enum	92.2	[87.9, 96.0]	91.5	[87.2, 95.4]	+0.7	[−5.3, 6.5]
Typed union	90.6	[86.3, 94.4]	90.4	[86.1, 94.3]	+0.1	[−5.7, 6.0]

The table lowers conflict accuracy. Relative to the one-line rule, the full table changes conflict accuracy by -5.6 points $[-8.8, -2.6]$ and aligned accuracy by $+4.3$ (Table 1). The DiD is -9.9 points $[-13.2, -6.8]$ (Holm $p < .001$). Conflict accuracy falls for four of the five primary models. The fifth, Qwen3-14B, gains 11.8 points on conflict tasks but 21.5 on aligned tasks, so all five per-model DiDs are negative, and each interval excludes zero (unadjusted, Figure 3). Qwen3-14B follows a different pattern (§6.1). Without it, the table leaves aligned accuracy unchanged (0.0) and lowers conflict accuracy by 9.9 points. Among executable calls, which exclude invalid calls and parse failures, the conflict loss is -8.4 points $[-11.8, -5.3]$. On the neutral controls, whose tasks do not touch the rule row, the aligned-minus-conflict difference has an interval containing zero for all 25 model-by-representation pairs, so the flipped row has no detectable effect on unrelated updates.

The table shifts responses toward competing operations. With the no-rule schema or the one-line rule, models almost always write the target field through an explicit form, whose meaning does not depend on the contract (536 and 540 of 540 conflict responses outside the omission strata, Appendix Table 10). With the full table, models use explicit forms less often. Bare values rise from 7 to 81 aligned responses and from 0 to 49 conflict responses. The aligned row maps the bare value to the requested operation, and the conflict row maps it to a competing operation, the one that produces the displaced state. Not every displaced response uses the bare value. Of the table's 77 displaced conflict responses, 45 use the bare value and 32 write the table row's operation explicitly. Under the one-line rule, all 18 displaced responses outside the omission strata use an explicit form. The table ends up below even the no-rule schema on conflict tasks (76.7% vs. 82.9%), a descriptive comparison.

Valid calls, wrong state. Under the one-line rule, 19 of 720 conflict responses (2.6%) end in the displaced state. Under the table, 77 (10.7%) do (Table 2). Of these 77, 47 leave a field absent instead of storing the requested null (42) or empty value (5). The other 30 record an inherited default instead of an explicit override. Invalid calls fall from 44 to 18. Every displaced call was accepted and executed, so the error shows up only in the resulting record. Aligned accuracy rises from 81.3% to 85.6%, so accuracy over both conditions barely moves, from 81.7% (1,177/1,440) to 81.1% (1,168/1,440), and hides the conflict loss.

The damage lands where the user asks to store a value. All 77 displaced responses fall in the three store tasks, where the conflict row says that sending the requested value bare clears or resets the field instead of setting it (Figure 2). The largest drop is for *null · store*, where conflict accuracy falls from 81.1% to 44.4% and 42 of 90 responses clear the field instead. Where the user asks to clear, reset, or keep, the table helps or is neutral (*null · remove*: 85.6% to 94.4%). The effect is an

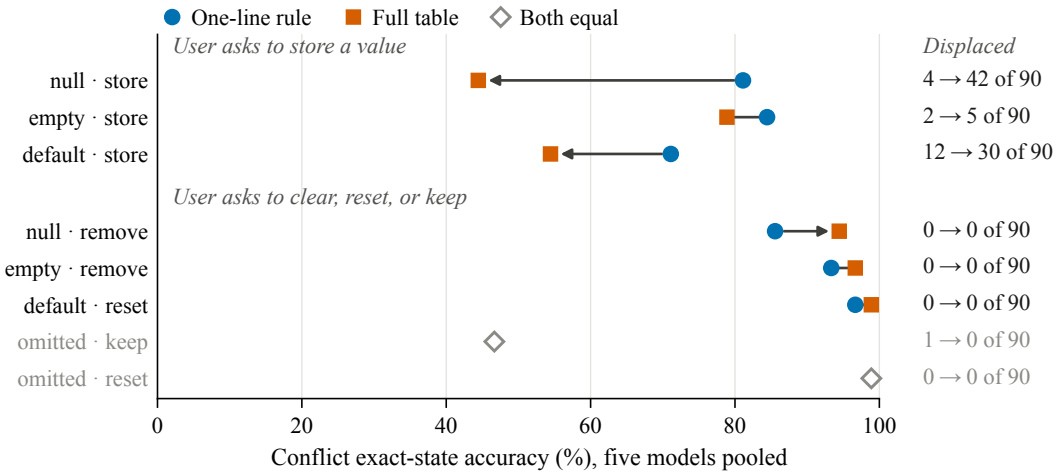

Figure 2: Conflict accuracy by stratum, one-line rule (start of arrow) to full table (head), pooled over five models (90 responses per stratum). Labels give displaced conflict responses. The loss concentrates on store tasks. The omission strata (grey) have the same conflict accuracy under both representations (footnote 1).

interaction between what the user asks for and what the table says the bare value does. In Figure 1, two models even emit `absence_mode:"clear"`, executing the table row rather than the request.[1]

Table 2: Executed outcomes, with 720 responses per row (counts, with percentages in parentheses). Displaced denotes the state the direct call reaches under the conflict contract. An aligned response is displaced when it reaches that same state, which means the model applied the competing operation. Invalid and displaced outcomes are disjoint. No model abstained.

Representation	Condition	Correct	Displaced	Invalid	Other wrong	Parse fail.
One-line rule	aligned	585 (81.3)	10 (1.4)	61 (8.5)	64 (8.9)	0 (0.0)
One-line rule	conflict	592 (82.2)	19 (2.6)	44 (6.1)	65 (9.0)	0 (0.0)
Full table	aligned	616 (85.6)	13 (1.8)	23 (3.2)	66 (9.2)	2 (0.3)
Full table	conflict	552 (76.7)	77 (10.7)	18 (2.5)	71 (9.9)	2 (0.3)

Operation-naming schemas. The flat enum and typed union reach 91.5% and 90.4% pooled conflict accuracy. The enum's $+9.3$-point gain over the one-line rule ($[7.4, 11.4]$, Holm $p < .001$) is concentrated in Qwen3-14B, which gains 41.0 points. The other four configurations average a 1.4-point gain.

5 Shorter tables recover much of the synthetic loss

Is the harm specific to the ten-row table? A follow-up on 120 new synthetic families (96 test, 24 controls) in the null, empty, and default strata compared four representations of equal length: the one-line rule, the ten-row full table, and two four-row tables. The *specified* four-row table gives an operation for every row. The *marked* four-row table gives an operation only for the rule row and marks the other three "not specified by this contract". This exploratory study scored 4,799 of 4,800 responses (§3).

The full table's harm replicates, on store tasks only. On the new families, the full table again trails the one-line rule, with a DiD of -13.1 points $[-17.7, -8.5]$ (Table 3). The whole loss is on store tasks (DiD -26.3, against 0.0 elsewhere), and displaced conflict responses rise from 12 to 73 of 480.

[1] Under the reset contract, a field left out of the call is reset, so a correct answer must list every untouched field. Models rarely do this under either representation, so these strata add almost nothing to the difference between representations but dilute the pooled estimates. Dropping their 36 families makes the conflict change -7.4 points and the DiD -12.6 (Appendix H). The frozen analysis keeps them.

Table 3: Table-design follow-up: accuracy (%) of each representation on 96 new test families, pooled over five models, with displaced conflict responses (of 480) and the DiD against the one-line rule (points, with 95% bootstrap intervals). Store tasks are *null · store*, *empty · store*, and *default · store*. †Descriptive, unregistered contrast using the same bootstrap. Registered contrasts and task splits are in Appendix Table 11.

	Conflict acc. (%)		Aligned	Displaced	DiD vs. one-line	
Representation	Store	Clear/reset	acc. (%)	(of 480)	Points	95% CI
One-line rule	79.2	93.3	85.4	12	ref.	
Full table (10 rows)	55.0	96.3	87.9	73	−13.1	[−17.7, −8.5]
Specified 4-row table	73.3	93.8	86.3	29	−3.5†	[−6.0, −1.0]
Marked 4-row table	74.2	90.8	85.4	20	−3.8	[−6.0, −1.5]

Four rows recover most of it. Replacing the ten-row table with a four-row one moves the DiD about 9.5 points toward zero overall and 22.1 points on store tasks. The two four-row tables move it by similar amounts, $+9.6$ points $[5.2, 14.2]$ for the specified table and $+9.4$ $[5.0, 14.0]$ for the marked table. Every model's estimate is positive for both, and for the marked table the gain excludes zero in four of five models, all but Qwen3-14B (Appendix E). Conflict responses that use the bare form fall from 52 with the full table to 7 (specified) and 2 (marked).

No detected additional benefit from unspecified marking. The marked-minus-specified contrast is -0.2 points $[-3.5, 2.7]$. The marked table still trails the one-line rule (DiD -3.8 $[-6.0, -1.5]$). Shortening the table also changes the listed mappings, their order, the header, and the filler, so the recovery does not isolate row count as the cause.

6 Other models and source-grounded rules

6.1 The representation effect varies across models

The additional cohort has smaller, heterogeneous effects. After the primary analysis, we ran an additional cohort of four newer models, GLM 5.3, Gemini 3.8 Flash, Kimi K3, and Grok 4.5, on the same held-out families with the one-line rule and the full table. All 3,072 responses were scored. GLM 5.3 reproduces the primary pattern (DiD -9.7 $[-16.0, -3.5]$, Figure 3). Gemini 3.8 Flash shows a small effect (-2.8), while the Kimi K3 and Grok 4.5 intervals include zero. The cohort DiD is -2.1 $[-4.7, 0.3]$, although the table still lowers the cohort's conflict accuracy (-2.6 $[-4.9, -0.3]$). Pooled over all nine models, the DiD is -6.4 $[-8.8, -4.2]$.

The table helps and hurts models in different ways. Qwen3-14B, the weakest primary model (48.6% conflict accuracy with the one-line rule, against 88.2–93.1% for the others), is the only one whose conflict accuracy rises with the table. Most of its gains come from responses that were invalid with the one-line rule and correct with the table, including 31 of its 35 aligned gains and 14 of its 19 conflict gains (Appendix Table 14). Its DiD is still negative (-9.7). Qwen3.8-27B shows the reverse. With the table, 23 of its conflict responses turn from correct to wrong, 20 of them to the displaced state, and only one turns from wrong to correct. Two models are too few to tell how the effect depends on capability.

The effect of reasoning is unresolved. Although thinking was requested off, DeepSeek V4.1 Flash and GPT-5.6 Terra returned reasoning tokens in many primary responses. An on/off follow-up could compare the two modes only for Sonnet, whose DiD was -6.9 points $[-13.9, 0.0]$ with thinking on versus -9.0 off. Turning thinking on also changed the temperature and the maximum response length, so the study cannot tell whether reasoning removes the effect (Appendix F).

6.2 Source-grounded rules score higher with a short table

A case series implements six selected rules from four public sources as local deterministic fixtures, each shown with a verbatim excerpt. Three rules follow JSON Merge Patch (omitted→keep, `null`→clear, empty→set). Three depart from it: Kubernetes Server-Side Apply (omitted→reset),

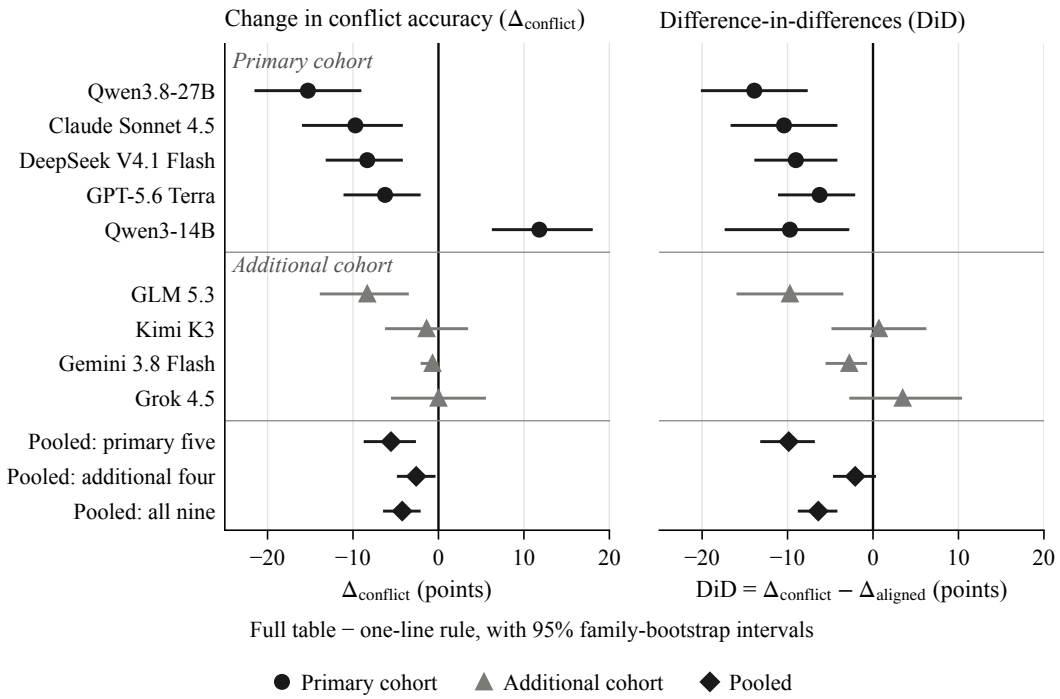

Figure 3: Full table minus one-line rule on synthetic held-out tasks (144 families per model, in points with 95% family-bootstrap intervals). *Left:* conflict-accuracy change. *Right:* conflict-minus-aligned DiD. Four primary models lose conflict accuracy, while all five have negative DiDs. The additional cohort of four newer models is heterogeneous. Pooled intervals condition on the configurations in each group. Source-grounded contrasts are reported separately in Table 4 and Appendix Table 13.

Stripe metadata (empty→clear), and a GitHub Enterprise Server webhook update (omitted→reset to the declared default) (GitHub, 2022; Hoffman & Snell, 2014; Kubernetes Authors; Stripe). The mappings follow the source documentation. Each of 144 families requests one update through a raw JSON patch without explicit forms. The one-line rule adds one sentence. The character-matched *source-status table* adds four rows, for omitted, null, empty, and default input, that state the source's rule and mark the forms it does not address as unspecified. For the three API models, these requests repeat a run whose outputs had already been seen, so the analysis is descriptive (Appendix C).

Table 4: Descriptive source-grounded case series: exact-state accuracy (%) per rule (120 responses each). Δ is table minus rule, and only the pooled rows (72 families each) have 95% intervals.

Source	Row	Documented action	One-line rule	Source-status table	Δ	95% CI
Rules that follow Merge Patch						
RFC 7396	empty string	store empty string	100.0	99.2	−0.8	
	`null`	clear	100.0	99.2	−0.8	
	omitted	keep	93.3	90.0	−3.3	
Rules that depart from Merge Patch						
GitHub webhook	omitted	reset to declared default	33.3	57.5	+24.2	
Kubernetes SSA	omitted	reset to default	43.3	60.0	+16.7	
Stripe metadata	empty string	clear	85.8	100.0	+14.2	
Follow Merge Patch, pooled (secondary)			97.8	96.1	−1.7	[−3.9, 0.6]
Depart from Merge Patch, pooled (primary)			54.2	72.5	+18.3	[15.6, 21.1]

The gain is largest for one model. On the rules that depart from Merge Patch, accuracy is 18.3 points higher with the source-status table ($[15.6, 21.1]$), rising from 54.2% to 72.5% (Table 4), although only two models, not the three the registration required, have a lower bound above 5 points. On the Merge Patch rules, already near ceiling, it has no clear effect (-1.7 $[-3.9, 0.6]$). The gain is $+54.2$ for Claude Sonnet 4.5, $+20.8$ for Qwen3.8-27B, $+11.1$ for GPT-5.6 Terra, and $+2.8$ each for

DeepSeek and Qwen3-14B (Appendix Table 13). Sonnet's one-line errors are mostly default values written as literal overrides (Appendix Figure 6). With thinking on, it answers 140 of 144 one-line requests correctly and its gain falls to $+4.2$.

6.3 A CALL-INTERFACE FOLLOW-UP DOES NOT REPRODUCE THE SOURCE-GROUNDED GAIN

An exploratory follow-up compared explicit forms with raw patches on 120 new synthetic clear/reset families (96 test, 24 controls) for seven API models, the three primary API models and the additional cohort (8,399 of 8,400 responses). Raw patches cannot express the conflict store tasks. With raw patches, a four-row table changes conflict accuracy by $+1.0$ points $[-0.6, 2.5]$ relative to the one-line rule, and the DiD is -0.3 $[-2.1, 1.5]$ (Appendix Tables 16–17). With the one-line rule, conflict accuracy is already near ceiling here (99.7% with explicit forms, 92.6% without), against 54.2% in the source-grounded series, so this comparison cannot explain the source-grounded gain.

7 DISCUSSION

Two possible explanations. The full table increases both bare-value errors and explicit choices of the competing operation, and shorter tables reduce bare-value use. First, by documenting rows such as `bare_other->set` that the one-line rule leaves out, the table may make bare values look like a normal way to write a field, so models send them for the target field too. Second, the listed operations may pull models toward the displayed row instead of the request. The largest failure, *null · store*, fits both, because there the conflict row agrees with Merge Patch. An exploratory diagnostic finds that a table loses to a sentence with the same length and content (DiD -7.5 $[-14.4, -1.9]$), which points at the format, but none of its six contrasts survives the Holm correction (Appendix I).

What the benchmark measures. With the one-line rule, models sidestep the rule row by writing explicit forms (540 of 540 conflict responses outside the omission strata), so conflict tasks become hard mainly when a representation draws models back to bare values (45 of the 77 displaced full-table responses). NothingBench diagnoses documentation changes by tying a change in performance to one predicted wrong state.

Consequences and evaluation checks. A wrong write can outlast the model response. An assistant updating live records could erase a field it was asked to populate, and later steps could act on the altered record without a validation error to prompt correction. The default errors are harder to see, because the displayed value is correct while the record encodes inheritance instead of an explicit choice. Downstream losses and deployment frequency remain unmeasured. For documentation authors, listing every form hurt the synthetic conflict tasks, four rows recovered most of the loss, and naming the operation scored highest, mostly through one model. On the source-grounded rules that depart from Merge Patch, which had no explicit forms and a far lower baseline, a short table scored higher. A call-interface follow-up did not reproduce the source-grounded gain (§6.3), so check documentation changes on paired conflict tasks with the intended model and interface.

Limitations. The tasks use one synthetic interface, shared development/test templates, a four-field record, and JSON in text rather than native function calling. Each configuration answered each request once, so intervals capture family variation but not reruns or a population of models, and hosted models may change behind their names. The omission strata stay in the frozen analysis but should be scored separately in new evaluations (Appendix H). The source-grounded series covers six selected rules whose requests three API models had already answered. The multi-turn follow-up (Appendix J) did not include the one-line rule. External generality and the causal mechanism remain open.

AI USE STATEMENT

Generative AI tools helped write and review the benchmark code, draft and edit the paper, search the related literature and verify citations, write figure and table code, draft analyses of already-frozen results, and simulate reviewer feedback on drafts. Because the authors are not native English speakers, these tools also helped with language and phrasing throughout. No language model runs in the

scoring path. Parsing, execution, and statistics are deterministic code, and no model generated scored responses except the evaluated models, judged a response, or altered a frozen input. The labels for the 60 audited `PATCH` operations (Appendix K) were finalized by human adjudication. The authors reviewed all AI-assisted work and take responsibility for the paper, its claims, and its artifacts.

Ethics statement

The study uses fictional records and public protocol documentation, involves no human subjects or personal data, and releases no model outputs beyond benchmark responses to synthetic tasks.

Reproducibility statement

The supplementary material contains the compiler, frozen manifests with request hashes, model and transport configuration, parser, oracle, raw and scored records, analysis scripts, and the scripts that generate every table and figure in this paper from frozen reports. Parsing, execution, and analysis can be rerun without a language model or provider credentials, and a replay reproduces all 3,199 development and 9,599 held-out scores with zero divergence.

References

Ahmed Nusayer Ashik, Shaowei Wang, Tse-Hsun Chen, Muhammad Asaduzzaman, and Yuan Tian. When LLMs lag behind: Knowledge conflicts from evolving APIs in code generation. arXiv:2604.09515, 2026.

Brian Barrett. How a 'NULL' license plate landed one hacker in ticket hell. WIRED, https://www.wired.com/story/null-license-plate-landed-one-hacker-ticket-hell/, August 2019. Accessed September 2026.

droogie. Go NULL yourself, or: How I learned to start worrying while getting fined for other's auto infractions. DEF CON 27, https://media.defcon.org/DEF%20CON%2027/DEF%20CON%2027%20presentations/DEFCON-27-droogie-go-null-yourself.pdf, 2019. Accessed September 2026.

Harvey Yiyun Fu, Aryan Shrivastava, Jared Moore, Peter West, Chenhao Tan, and Ari Holtzman. Absence Bench: Language models can't see what's missing. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics (EMNLP)*, 2020.

GitHub. Update a global webhook, GitHub Enterprise Server 3.4 REST API. https://docs.github.com/enterprise-server@3.4/rest/reference/enterprise-admin#update-a-global-webhook, 2022. Accessed September 2026.

Google. FieldMask. Protocol Buffers documentation, https://protobuf.dev/reference/protobuf/google.protobuf/#fieldmask, a. Accessed September 2026.

Google. Application note: Field presence. Protocol Buffers documentation, https://protobuf.dev/programming-guides/field_presence/, b. Accessed September 2026.

Google. AIP-134: Update. https://google.aip.dev/134, 2019. Accessed September 2026.

GraphQL Foundation. GraphQL specification, October 2021 edition. https://spec.graphql.org/October2021/, 2021. Accessed September 2026.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X. Wang, and Sadid Hasan. Does prompt formatting have any impact on LLM performance? arXiv:2411.10541, 2024.

Paul Hoffman and James Snell. JSON merge patch. RFC 7396, https://www.rfc-editor.org/rfc/rfc7396, 2014. Accessed September 2026.

Cheng-Yu Hsieh, Si-An Chen, Chun-Liang Li, Yasuhisa Fujii, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Tool documentation enables zero-shot tool-usage with large language models. arXiv:2308.00675, 2023.

JSON Schema. Annotations: `default`. https://json-schema.org/understanding-json-schema/reference/annotations. Accessed September 2026.

Kubernetes Authors. Server-side apply. https://kubernetes.io/docs/reference/using-api/server-side-apply/. Accessed September 2026.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Xiang Lisa Li, Farzaan Kaiyom, Evan Zheran Liu, Yifan Mai, Percy Liang, and Tatsunori Hashimoto. AutoBencher: Towards declarative benchmark construction. In *International Conference on Learning Representations (ICLR)*, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024a.

Zeyu Leo Liu, Shrey Pandit, Xi Ye, Eunsol Choi, and Greg Durrett. CodeUpdateArena: Benchmarking knowledge editing on API updates. arXiv:2407.06249, 2024b.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics (NAACL)*, 2025.

Yaxing Lyu, Shengjie Zhou, Binbin Toh, Pengyu Zhu, and Lijun Li. KC-Bench: A dynamic interactive benchmark for evaluating knowledge conflicts in LLM agents. arXiv:2609.03588, 2026.

Seiji Maekawa, Jackson Hassell, Pouya Pezeshkpour, Tom Mitchell, and Estevam Hruschka. Towards reliable benchmarking: A contamination free, controllable evaluation framework for multi-step LLM function calling. In *International Conference on Learning Representations (ICLR)*, 2026.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121, 2024.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*, 2023.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *International Conference on Machine Learning (ICML)*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations (ICLR)*, 2024.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. From exploration to mastery: Enabling LLMs to master tools via self-driven interactions. In *International Conference on Learning Representations (ICLR)*, 2025.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations (ICLR)*, 2024.

Nikolai Skripko. Instruction-following evaluation in function calling for large language models. arXiv:2509.18420, 2025.

Stripe. Metadata. https://docs.stripe.com/metadata. Accessed September 2026.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 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*, 2024.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently. arXiv:2303.03846, 2023.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha V. Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *International Conference on Learning Representations (ICLR)*, 2025.

Kevin Wu, Eric Wu, and James Zou. ClashEval: Quantifying the tug-of-war between an LLM's internal prior and external evidence. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024a.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? Exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024b.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations (ICLR)*, 2024.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations (ICLR)*, 2025.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Kan Ren, Dongsheng Li, and Deqing Yang. EASYTOOL: Enhancing LLM-based agents with concise tool instruction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. arXiv:2311.07911, 2023.

A THE TEASER FAMILY IN FULL

Family `F26-final-null-right-0111` (held-out, stratum *null · store*). The user message, identical in all four rendered requests (aligned/conflict × one-line/full), is:

```
The policy outcome for the workshop record R26-0111 is specified indirectly
by these requested changes. [...]
REQUESTED_UPDATES:
- store an explicit null in f26_0111_a (value=null)
- leave f26_0111_b exactly as it is
- set f26_0111_c to the requested replacement (value={"level": 3, "zone": "new-14"})
- store an explicitly empty value in f26_0111_d (value=false)
Return one update_record call and do not mutate unrelated state.
```

All four requests share the documentation prefix: *"Fresh fictional update_record contract. Apply the active mapping exactly; familiar API conventions are irrelevant. Operations are Keep (leave unchanged), Set(value) (store an override), Clear (remove presence), or Reset (inherit the current default). Entries carry a field and one of three explicit channels: absence_mode names Keep, Clear or Reset; literal:true stores the supplied value; a bare value with no literal flag is read through the local contract."* The one-line rule then reads `Local token bare_null is interpreted as set` (aligned) or `clear` (conflict), followed by five neutral notes. In the conflict version, the full table's documentation after the prefix reads, verbatim:

```
Complete surface-form truth table, no examples: bare_default->set,
bare_empty->set, bare_no_value->keep, bare_null->clear, bare_other->set,
literal->set, mode_clear->clear, mode_keep->keep, mode_reset->reset,
omitted->keep. NOTES: This record belongs to a fictional dataset used
only for evaluation. Units are unspecified.
```

The aligned version has `bare_null->set`. The reference call, correct in all four, writes field `a` as `{"literal":true,"value":null}`, and the direct call writes it bare. Under conflict/full, DeepSeek V4.1 Flash answered (whitespace reformatted for display):

```
{"name": "update_record", "arguments": {"record_id": "R26-0111", "updates": [
  {"field": "f26_0111_a", "value": null},
  {"field": "f26_0111_b", "absence_mode": "keep"},
  {"field": "f26_0111_c", "value": {"level": 3, "zone": "new-14"}},
  {"field": "f26_0111_d", "value": false}]}}
```

which clears field `a` (the displaced state). Claude Sonnet 4.5 and Qwen3.8-27B wrote the same bare null, and GPT-5.6 Terra and Qwen3-14B wrote `absence_mode:"clear"`. Under conflict/one-line, all models except Qwen3-14B used the literal form, and Qwen3-14B wrote `absence_mode:"clear"`.

B BENCHMARK DETAILS

For the aligned version of the teaser family, the documentation suffix after the shared prefix is listed below. Parenthetical lengths count the complete serialized request:

No-rule schema (2,587 characters): neutral notes only. No mapping is stated, so aligned and conflict requests are byte-identical.

One-line rule (2,597): `Local token bare_null is interpreted as set.`

Full table (2,596): the ten-row table of Figure 1.

Flat operation enum (2,598): `Name the operation explicitly: op=keep, op=set with value, op=clear, op=reset. No surface form is interpreted; invalid op/value combinations are rejected.` Entries are `{field, op, value}`.

Typed union (2,599): `{op:keep}, {op:set,value}, {op:clear}, or {op:reset}`. Only the set branch carries a value.

The first three share the entry schema `{field, absence_mode, literal, value}`. Its ten surface forms are an omitted field (`omitted`), an entry with a field but no value (`bare_no_value`), a bare `null`, a bare empty value (the empty or falsy value of the field's type, such as `false` for a boolean), a bare copy of

the current default, any other bare value (`bare_other`), the three `absence_mode` values, and `literal:true` with a value. The system prompt asks for exactly one JSON object, either one `update_record` call or an abstention.

Serving. Claude Sonnet 4.5 uses the dated endpoint `claude-sonnet-4-5-20250929`, GPT-5.6 Terra the Codex endpoint, and DeepSeek V4.1 Flash and the additional cohort a third-party gateway. Qwen3.8-27B (NVFP4) and Qwen3-14B are pinned Hugging Face revisions served with SGLang. Every request asks for temperature 0 and at most 1,024 output tokens. The supplementary configuration lists exact model strings, checkpoint revisions, and the serving image digest.

Figure 4 shows how each task is paired, and Table 5 defines the eight strata.

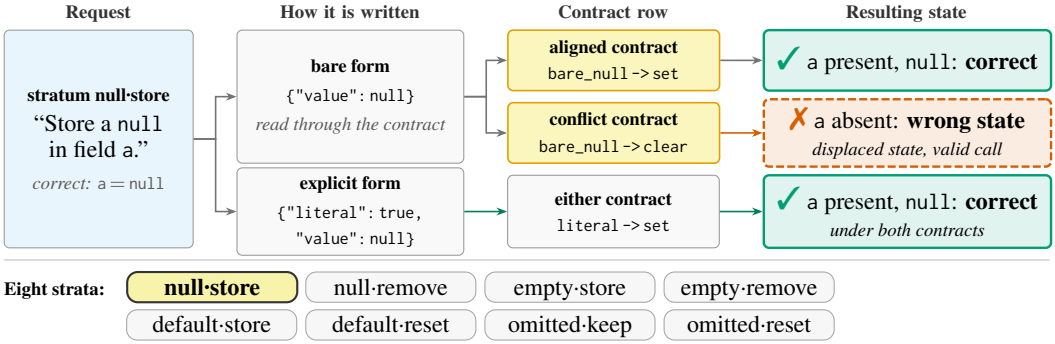

Figure 4: How one *null · store* task gets an aligned and a conflict version. The aligned row maps bare null to set, and the conflict row maps it to clear. An explicit literal form is correct under both. The eight strata pair each rule row with two requests.

Table 5: The eight strata. The direct call writes the request in the rule row's bare form. The aligned contract maps that form to the requested operation. The conflict contract flips the row, so the direct call produces the displaced state. Documented protocol rules fall on both sides: `null`→clear and omitted→keep are JSON Merge Patch, empty→clear is Stripe metadata, and omitted→reset is Kubernetes Server-Side Apply.

Stratum	Task asks to	Direct call	Aligned row	Conflict row	Displaced state
omitted · keep	leave a field unchanged	omit the field	omitted→keep	omitted→reset	field reset to default
omitted · reset	restore the default	omit the field	omitted→reset	omitted→keep	field unchanged
null · remove	remove the value	bare `null`	`null`→clear	`null`→set	null stored
null · store	store a null	bare `null`	`null`→set	`null`→clear	field cleared
empty · store	store an empty value	bare empty value	empty→set	empty→clear	field cleared
empty · remove	remove the value	bare empty value	empty→clear	empty→set	empty value stored
default · store	store the default value	bare default value	default→set	default→reset	field reset to default
default · reset	restore the default	bare default value	default→reset	default→set	default stored as override

C STUDIES, COVERAGE, PREREGISTRATION, AND DEVIATIONS

Table 6 lists every study with its coverage, and Table 7 every registered gate. Figure 5 shows selected contrasts, grouped by estimand. Several rows can come from one study.

We froze the held-out families, the seven contrasts of Table 8, and the main-study gates before sending any held-out request. Separately registered follow-ups have their own gates. Across all studies, six criteria were not met (Table 7). The positive-claim gate D3 asked for a representation with a consistently positive aligned-minus-conflict margin across models. That criterion was not met. The focal representation result uses the distinct DiD contrast added after development and frozen before held-out access (§3). The source-grounded study also fell short of its registered criterion and is reported as a descriptive case series. The isolation diagnostic lacks 3 of 3,200 responses and the table-design and call-interface follow-ups each lack one, so all three are exploratory. The reasoning-mode follow-up stopped with one usable model and is also exploratory.

Deviations. (1) D3's text says "three of four models" because it predates the fifth model. GPT-5.6 Terra was added before any five-model output existed, and the analyzer applies "at least three" to five

Table 6: The ten studies: where they are reported, which models they use, and expected and scored responses. Requests with no model reply after the allowed retries have no score and are not imputed. The additional cohort's gateway needed many retries, but every one of its responses was scored. Gate outcomes are explained in Table 7. Development used the same gates as the held-out run and passed all of them except the margin gate.

Study	Section	Models	Unit	Expected	Scored	Missing	Status
Held-out	§4	5 primary	responses	9,600	9,599	1	margin gate not met
Development	§3	5 primary	responses	3,200	3,199	1	margin gate not met
Additional cohort	§6.1	4 newer	responses	3,072	3,072	0	complete
Reasoning mode	§6.1	2 API	responses	3,648	3,648	0	stopped (1 eligible)
Source-grounded	§6.2	5 primary	responses	1,440	1,440	0	gate not met
Table design	§5	5 primary	responses	4,800	4,799	1	1 response missing
Call interface	§6.3	7 API	responses	8,400	8,399	1	1 response missing
Omission rule	App. H	5 primary	responses	960	960	0	complete
Isolation diagnostic	App. I	5 primary	responses	3,200	3,197	3	3 responses missing
Stateful transfer	App. J	5 primary	episodes	960	955	5	gates pass, 5 missing

Table 7: Preregistered gates and outcomes. D3 is the preregistered positive-margin gate. "Lower bound" is the lower end of a 95% family-bootstrap interval, in points.

Study	Gate	Outcome
Held-out	Integrity, provider errors, parser checks	pass
	Neutral controls unaffected	pass
	Structural invariance: operation-naming schemas score the same on average under both conditions	pass, but only because the two omission strata err in opposite directions (App. H)
	D3: some representation's aligned-minus-conflict margin has lower bound ≥ 5 in at least three models	not met: only Qwen3.8-27B
Additional cohort	Provider errors	pass
Reasoning mode	Reasoning evidence on $\geq 95\%$ of thinking-on responses for two models	stopped: GPT-5.6 Terra at 89.8% (synthetic) and 48.3% (source-grounded)
Source-grounded	At least three models with a lower bound above 5 on the gain for rules that depart from Merge Patch	not met: only two models clear 5, and GPT-5.6 Terra's lower bound is 4.2
Table design	Integrity, coverage $\geq 99\%$ per model, neutral controls unaffected, parser	pass, and the registered rule selects "design explains the sign"
	Every response scored	incomplete: 4,799 of 4,800
Call interface	Integrity, coverage, neutral controls unaffected, parser, decision rule	pass, "interface does not explain the sign", incomplete: 8,399 of 8,400
Omission rule	Integrity, coverage, parser, interpretation rule	pass, interpretation "mixed"
Isolation diagnostic	Every response scored	incomplete: 3,197 of 3,200
Stateful transfer	Integrity and all registered gates	pass

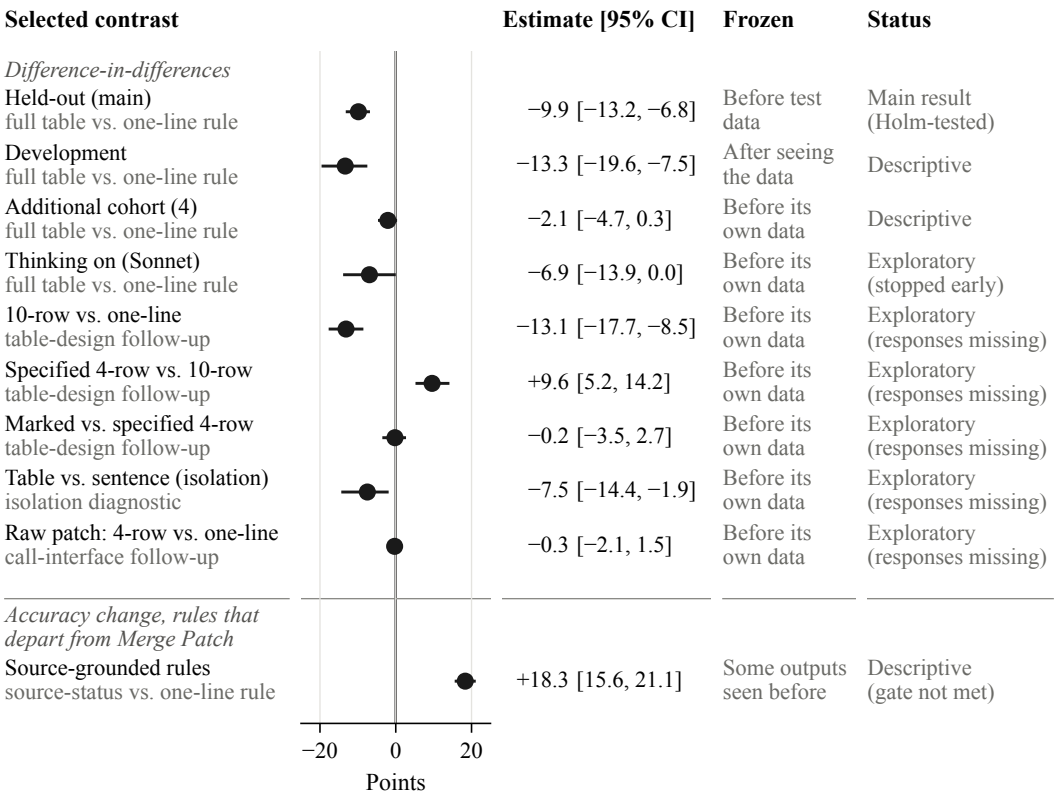

Figure 5: Selected study contrasts (percentage points, with 95% family-bootstrap intervals), grouped by estimand. "Frozen" gives the registration timing. "Status" distinguishes the main Holm-tested result from descriptive and exploratory analyses. Exploratory studies are missing responses or stopped early. The omission and stateful-transfer studies are listed in Table 6 but have no point in this plot.

models. (2) Two DiD contrasts joined the Holm family before the held-out split was opened, after development showed that a conflict-only contrast can reflect call-format effects rather than contract effects. (3) The additional cohort was registered after the primary analysis. Its models were chosen by availability and family diversity before their outputs existed. (4) The analyzed source-grounded run reuses byte-identical requests from an earlier run in which the two local models were unavailable but the three API models' results were seen. It is therefore a blind confirmation only for the two local models. (5) Thinking was requested off for every model, but DeepSeek V4.1 Flash and GPT-5.6 Terra returned reasoning tokens (§6.1). Temperature 0 was requested but not verified for every provider. (6) The table-design, omission-rule, reasoning-mode and call-interface follow-ups were registered after the main analysis, from its results, and frozen before any of their own outputs. The table-design study excludes the omission strata for the reason in Appendix H. (7) The reasoning-mode follow-up registered three models. Before any thinking-on request was sent, we dropped the Qwen3.8-27B blocks for runtime reasons, so the registered rule then required both API models to be eligible. (8) The call-interface follow-up covers only clear and reset strata, and its raw-patch conflict contracts swap two rows rather than flipping one, so that exactly one surface form is correct. The supplementary material contains the full decision log, freeze records, and superseded runs.

D Details of the main result

These tables support §4: the preregistered contrasts (Table 8), accuracy by stratum (Table 9), and how models write the target field (Table 10).

Table 8: The seven contrasts preregistered before the held-out split was opened, with two-sided family-bootstrap p values (resolution $1/10{,}001$) and Holm adjustment. Intervals and p values come from separately seeded resamples and can disagree near zero, as for typed union minus flat enum.

Registered contrast	Estimate [95% CI]	p	Holm p	Reject at .05
Aligned − conflict margin, one-line rule	−1.0 [−6.8, 5.0]	.763	.763	no
Aligned − conflict margin, full table	+8.9 [2.4, 15.3]	.009	.036	yes
Full table − one-line rule, conflict	−5.6 [−8.8, −2.6]	<.001	.002	yes
Flat enum − one-line rule, conflict	+9.3 [7.4, 11.4]	<.001	<.001	yes
Typed union − flat enum, conflict	−1.1 [−2.2, −0.1]	.057	.154	no
Flat enum − one-line rule, DiD	−1.7 [−3.3, 0.0]	.051	.154	no
Full table − one-line rule, DiD	−9.9 [−13.2, −6.8]	<.001	<.001	yes

Table 9: Held-out accuracy (%) by stratum, pooled over five models (90 responses per stratum and condition), and conflict responses in the displaced state. "Task asks to" is the operation requested on the target field. The omission strata stay near three quarters combined accuracy because models rarely list untouched fields (Appendix H), with zero pooled conflict change and a small aligned contribution to the DiD.

Stratum	Task asks to	Aligned acc.		Conflict acc.		Conflict displaced	
		One-line	Full	One-line	Full	One-line	Full
omitted · keep	keep	96.7	97.8	46.7	46.7	1	0
omitted · reset	reset to default	46.7	48.9	98.9	98.9	0	0
null · remove	clear	80.0	94.4	85.6	94.4	0	0
null · store	store null	83.3	87.8	81.1	44.4	4	42
empty · store	store empty value	87.8	85.6	84.4	78.9	2	5
empty · remove	clear	85.6	96.7	93.3	96.7	0	0
default · store	store default value	73.3	74.4	71.1	54.4	12	30
default · reset	reset to default	96.7	98.9	96.7	98.9	0	0

Table 10: How models write the target field on the held-out null, empty, and default strata (test families, five models): through an explicit form (`literal:true` or `absence_mode`), as a bare value read through the contract, or not at all. Explicit forms are right or wrong regardless of the contract row. The last column splits displaced responses by form.

		Explicit	Bare	Field			Displaced	
Representation	Condition	form	form	left out	Parse fail.	Total	explicit	bare
No-rule schema	aligned	536	4	0	0	540	15	0
No-rule schema	conflict	536	4	0	0	540	18	4
One-line rule	aligned	533	7	0	0	540	10	0
One-line rule	conflict	540	0	0	0	540	18	0
Full table	aligned	458	81	0	1	540	13	0
Full table	conflict	490	49	0	1	540	32	45

E Details of the table-design follow-up

These tables support §5: the registered contrasts with their split by task (Table 11) and each representation's accuracy and bare-form use (Table 12). Per model, the marked table's DiD gain over the full table is $+10.4$ $[3.1, 17.7]$ for Claude Sonnet 4.5, $+10.4$ $[2.1, 18.8]$ for DeepSeek V4.1 Flash, $+7.3$ $[2.1, 12.5]$ for GPT-5.6 Terra, $+17.7$ $[6.2, 29.2]$ for Qwen3.8-27B, and $+1.0$ $[-6.2, 7.3]$ for Qwen3-14B. For the specified table the gains are $+4.2$ $[-4.2, 12.5]$, $+13.5$ $[5.2, 22.9]$, $+6.2$ $[2.1, 11.5]$, $+21.9$ $[11.5, 32.3]$, and $+2.1$ $[-5.2, 9.4]$.

Table 11: Registered contrasts of the table-design follow-up (DiD, treatment minus baseline, 95% bootstrap interval), overall and split by the operation the task requests (48 families each). Holm correction covers the four registered contrasts, and the marking contrast is secondary.

Contrast (DiD, points)	All tasks [95% CI]	Store tasks	Clear/reset tasks	Holm p
Full table − one-line rule	−13.1 [−17.7, −8.5]	−26.3	0.0	<.001
Specified 4-row − full table	+9.6 [5.2, 14.2]	+22.1	−2.9	<.001
Marked 4-row − full table	+9.4 [5.0, 14.0]	+22.1	−3.3	<.001
Marked 4-row − one-line rule	−3.8 [−6.0, −1.5]	−4.2	−3.3	.003
Marked 4-row − specified 4-row	−0.2 [−3.5, 2.7]	0.0	−0.4	–

Table 12: Table-design follow-up: accuracy, displaced conflict responses, and bare-form entries per representation (96 test families, five models).

Representation	Aligned acc.	Conflict acc. [95% CI]	Conflict responses displaced	Conflict responses bare form
One-line rule (1 row)	85.4	86.3 [83.1, 89.2]	12	0
Full table (10 rows)	87.9	75.6 [70.0, 81.0]	73	52
Specified 4-row table	86.3	83.5 [80.0, 86.9]	29	7
Marked 4-row table	85.4	82.5 [78.8, 86.0]	20	2

F Details by model

These tables support §6.1: per-model contrasts (Table 13), the Qwen transitions (Table 14), and the reasoning-mode follow-up (Table 15).

Table 13: Per-model contrasts against the one-line rule (points, unadjusted 95% intervals). The first three columns compare the full table on the synthetic held-out tasks, and Figure 3 plots the conflict and DiD columns. The last column compares the source-status table on the source-grounded rules that depart from Merge Patch. GPT-5.6 Terra's aligned interval is $[0.0, 0.0]$ because its aligned correctness is identical family by family under both representations.

Model	Aligned Δ	Conflict Δ	DiD (conflict − aligned)	Source-grounded Δ
Primary cohort				
Qwen3.8-27B	−1.4 [−3.5, 0.0]	−15.3 [−21.5, −9.0]	−13.9 [−20.1, −7.6]	+20.8 [12.5, 30.6]
Claude Sonnet 4.5	+0.7 [−2.1, 4.2]	−9.7 [−16.0, −4.2]	−10.4 [−16.7, −4.2]	+54.2 [43.1, 65.3]
DeepSeek V4.1 Flash	+0.7 [−2.1, 3.5]	−8.3 [−13.2, −4.2]	−9.0 [−13.9, −4.2]	+2.8 [−2.8, 8.3]
GPT-5.6 Terra	0.0 [0.0, 0.0]	−6.3 [−11.1, −2.1]	−6.3 [−11.1, −2.1]	+11.1 [4.2, 19.4]
Qwen3-14B	+21.5 [13.9, 29.2]	+11.8 [6.3, 18.1]	−9.7 [−17.4, −2.8]	+2.8 [0.0, 6.9]
Additional cohort				
GLM 5.3	+1.4 [−2.1, 4.9]	−8.3 [−13.9, −3.5]	−9.7 [−16.0, −3.5]	–
Kimi K3	−2.1 [−5.6, 0.7]	−1.4 [−6.3, 3.5]	+0.7 [−4.9, 6.3]	–
Gemini 3.8 Flash	+2.1 [0.0, 4.9]	−0.7 [−2.1, 0.0]	−2.8 [−5.6, −0.7]	–
Grok 4.5	−3.5 [−6.9, −0.7]	0.0 [−5.6, 5.6]	+3.5 [−2.8, 10.4]	–

In the primary run, DeepSeek V4.1 Flash returned reasoning tokens on 1,916 of 1,919 responses and GPT-5.6 Terra on 1,167 of 1,920, despite thinking being requested off. In the reasoning-mode follow-up, Sonnet's thinking-on answers contained a median of about 1,850 reasoning characters

Table 14: One-line-to-full transitions on the 144 test families for the two Qwen models. Recoveries give the error class a response left, and regressions the error class it entered.

		Wrong → right				Right → wrong				
Model	Condition	total	invalid	displaced	other	total	invalid	displaced	other	Net
Qwen3-14B	aligned	35	31	1	3	4	3	0	1	31
Qwen3-14B	conflict	19	14	3	2	2	1	0	1	17
Qwen3.8-27B	aligned	0	0	0	0	2	0	0	2	−2
Qwen3.8-27B	conflict	1	0	0	1	23	0	20	3	−22

per synthetic answer. GPT-5.6 Terra had a median of 32 reasoning tokens in both modes, counting responses with none. It showed reasoning on 89.8% of its synthetic thinking-on responses and so failed the registered reasoning check, which required reasoning evidence on at least 95% of thinking-on responses. The planned Qwen3.8-27B runs were not done. The off-mode reference in Table 15 comes from this new follow-up run. For GPT-5.6 Terra that rerun gives a DiD of -3.5, against -6.3 in the primary run on the same families, so reruns under nominally identical settings can differ by several points, a source of variation our intervals do not capture. Thinking-on requests also left the temperature unset (the provider default) where thinking-off requests set it to 0, and the maximum response length changes with mode, so we do not treat this as an isolated reasoning manipulation.

Table 15: Reasoning-mode follow-up: share of synthetic thinking-on responses with reasoning evidence, and the full-minus-one-line DiD on the held-out families with thinking off and on (95% bootstrap intervals). The source-grounded rows are with thinking on. GPT-5.6 Terra, shown descriptively, failed the registered check of reasoning evidence on at least 95% of thinking-on responses.

Quantity	Claude Sonnet 4.5	GPT-5.6 Terra
Responses with reasoning evidence, thinking on (%)	100.0	89.8
Eligible as a thinking configuration	yes	no
DiD, thinking off	−9.0 [−15.3, −2.8]	−3.5 [−6.9, 0.0]
DiD, thinking on	−6.9 [−13.9, 0.0]	−6.3 [−11.1, −2.1]
Change in DiD (on − off)	+2.1 [−6.9, 11.1]	−2.8 [−6.9, 0.7]
Source-grounded Δ, rules departing from Merge Patch (thinking on)	+4.2 [−1.4, 11.1]	+8.3 [2.8, 15.3]
Source-grounded one-line rule, responses correct (thinking on)	140/144	138/144

G Details of the source-grounded and interface studies

Figure 6 shows one source-grounded task whose rule departs from Merge Patch, and Tables 16 and 17 give the call-interface follow-up (§6.3).

Task (family `k8s-ssa-omitted-reset-01`): *"restore* `f26_12004_b` *to the current policy default"*. Current value `"requested-12052"`, default `"baseline-12015"`. Both representations open with the same verbatim Server-Side Apply excerpt: a removed field that no other manager owns "is either deleted from the live object or reset to its default value, if it has one".

One-line rule	**Source-status table**
`For this task, when field not provided is used, restore the source-documented default.`	`field not provided -> restore the documented default; explicit null -> not specified by this source; empty string -> not specified by this source; default-valued input -> not specified by this source.`
Claude Sonnet 4.5: `"patch": {"f26_12004_b": "baseline-12015"}` ⇒ default written as a literal, an action the source does not document	Claude Sonnet 4.5: `"patch": {}` ⇒ field omitted, the documented reset (correct)

Figure 6: A source-grounded task whose rule departs from Merge Patch, verbatim but trimmed. Under the one-line rule, the model writes the default value as a literal, the most common error on the two reset rules. The table marks default-valued input as not specified by the source, and the same model omits the field instead.

Table 16: Call-interface follow-up: accuracy (%) on 96 clear/reset test families, pooled over seven API models.

Interface and representation	Aligned acc.	Conflict acc. [95% CI]
Explicit forms, one-line rule	99.6	99.7 [99.3, 100.0]
Explicit forms, 4-row table	99.7	99.6 [99.0, 100.0]
Raw patch, one-line rule	95.4	92.6 [90.3, 94.6]
Raw patch, 4-row table	96.7	93.6 [91.5, 95.4]
Raw patch, 10-row table	96.4	90.6 [88.7, 92.6]

Table 17: Call-interface follow-up: registered contrasts (DiD, 95% bootstrap interval, Holm-adjusted p).

Contrast (DiD, points)	All models [95% CI]	Holm p
Raw patch: 4-row − one-line	−0.3 [−2.1, 1.5]	1.000
Interaction (patch − explicit forms)	−0.1 [−1.9, 1.7]	1.000
Explicit forms: 4-row − one-line	−0.3 [−1.0, 0.4]	1.000
Raw patch: 10-row − 4-row	−2.7 [−4.9, −0.4]	.087

H SCORING OF THE OMISSION STRATA

In the omission strata, the oracle keeps or resets every field the model leaves out of `updates`, according to the contract's `omitted` row. Under `omitted`→reset, a single-field request is correct only if the call also lists the three untouched fields with `absence_mode:"keep"`. The four models other than Qwen3-14B do so in 1 of 360 such responses across representations (358 responses list only the requested field), while scoring 98–100% on all other omission items. The one-line rule and the full table state the omission rule, and models still send sparse calls (72 of 72 responses under the one-line rule and 71 of 72 under the full table, plus one parse failure). Across the two omission strata, one-line and full have the same pooled conflict correctness (131/180), but aligned correctness differs (129/180 versus 132/180). Their contribution to the full conflict change is zero, while the aligned difference contributes $100(-3/720) = -0.42$ percentage points to the full DiD. The no-rule and operation-naming schemas do not state the rule, so their aligned and conflict requests are identical and each response is scored against both contracts. This scoring explains why omission accuracy stays near three quarters in every representation (71.7–73.3% for the one-line rule and the full table). The structural-invariance gate checks that the operation-naming schemas score the same on average under both conditions. It passes only because the two omission strata err in opposite directions. The same scoring also accounts for all eight failing omission scenarios in the transfer study (Appendix J).

Descriptive omission sensitivity. Excluding the 36 omission families leaves 108 families and 540 responses per representation–condition. One-line correctness is 456 aligned and 461 conflict, and full correctness is 484 aligned and 421 conflict. Thus conflict accuracy changes by $100(-40/540) = -7.41$ percentage points and the DiD is $100(-40 - 28)/540 = -12.59$ points. Bootstrap intervals were not recomputed for these descriptive estimates. All frozen estimates retain the original families and scores.

A follow-up tested whether the sparse calls come from how the rule is stated. It built 32 new single-field families (16 per polarity) and added one condition-specific sentence to every representation ("Any editable field you do not list in updates is reset to its current default" or "…left unchanged"), so aligned and conflict requests always differ. All 960 responses were scored, with no provider error. Under the keep rule every model is correct on every request. Under the reset rule, accuracy for the four models other than Qwen3-14B rises from 0% in the main study to 59–63% (Table 18). The registered interpretation rule classifies this as mixed: DeepSeek V4.1 Flash and GPT-5.6 Terra largely follow the plain rule, Claude Sonnet 4.5 follows it about half the time, and both Qwen models never do.

Table 18: Omission-rule follow-up: accuracy (%) on single-field requests under the reset rule once the rule is stated plainly in every representation (32 families per model), with the main study's value for the same request type.

Model	One-line rule	Full table	Flat enum
Claude Sonnet 4.5	53.1	62.5	56.3
DeepSeek V4.1 Flash	96.9	100.0	93.8
GPT-5.6 Terra	100.0	75.0	96.9
Qwen3.8-27B	0.0	0.0	0.0
Qwen3-14B	0.0	0.0	0.0
Four models (excl. Qwen3-14B)	62.5 [57.8, 67.2]	59.4 [53.1, 65.6]	61.7 [56.3, 66.4]
Same four, main study (original rule wording or none)	0.0	0.0	0.0

I Exploratory isolation diagnostic

A factorial diagnostic crossed ten documentation variants with both conditions on 32 new synthetic families and the five primary models (3,197 of 3,200 responses scored). Each contrast is a DiD of treatment minus baseline (Figure 7). Re-running the main table-minus-rule contrast on the new families gives -11.3 points $[-20.0, -3.1]$. A table minus a sentence with the same length and content gives -7.5 $[-14.4, -1.9]$. The other four contrasts are small: mapping density (full table with the target row first minus a padded table form of the one-line rule) $+3.1$, neutral length $+0.2$, target row first minus last $+3.1$, and marking the active row $+0.6$. None of the six contrasts is significant after Holm correction (adjusted $p = .065$ and $.117$ for the first two).

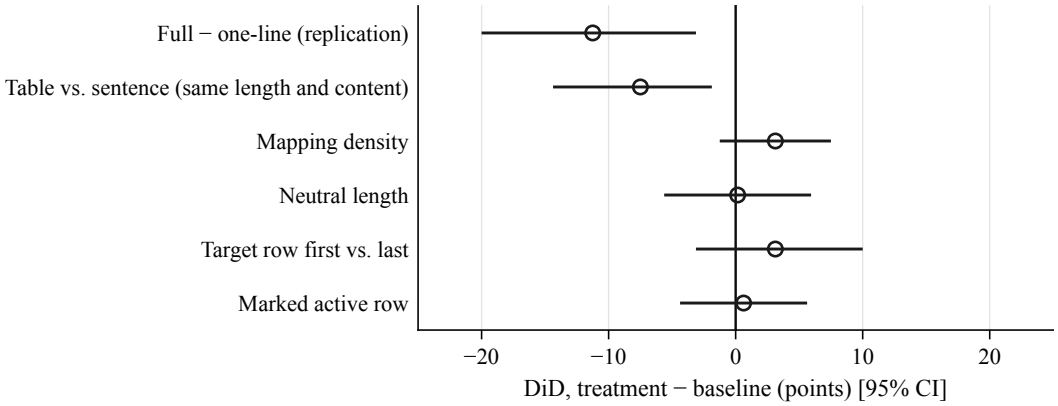

Figure 7: Exploratory factor contrasts (DiD as in Eq. 1, treatment minus baseline, with unadjusted 95% bootstrap intervals). None of the six contrasts survives the registered Holm correction.

J Stateful transfer diagnostic

A frozen multi-turn suite of 64 scenarios ran the full table and both operation-naming schemas for the five primary models (955 of 960 episodes scored). In every representation, the four models other than Qwen3-14B reached 84.4–87.5% terminal accuracy and solved 24 of the 32 non-control scenarios (the scenarios that exercise the rule). All eight failures are omission scenarios that fail at their first turn because of the omission scoring in Appendix H, before any rule-row turn. Qwen3-14B was much weaker throughout. Without the one-line rule, the suite cannot test the main comparison.

K Protocol audit

We coded eight normative sources to check that the rule rows reflect documented interface semantics (Google, a;b; 2019; GraphQL Foundation, 2021; Hoffman & Snell, 2014; JSON Schema; Kubernetes Authors; Stripe). We also inspected an APIs.guru snapshot: 40 randomly sampled and 20 targeted public `PATCH` operations. Human review found an explicit rule for omitted fields in one of the 40 randomly sampled operations and none of the 20 targeted operations.