

ConFormBench: Evaluating Evidence-Supported Record Maintenance in Conversational Form Filling

Timothy Veigel 

LMU Munich
BMW Group
Munich, Germany
timothy.veigel@ifi.lmu.de

Sven Mayer 

TU Dortmund University
Research Center for Trustworthy AI
Dortmund, Germany
sven.mayer@tu-dortmund.de

Abstract

Conversational form-filling systems must update structured records across dialogue turns. A user may add, correct, or retract information, while the existing record may itself contain stale or unsupported values. We call correct maintenance of this state record commitment fidelity: after each turn, the complete record should contain all and only commitments supported by the schema and dialogue history. We introduce CONFORMBENCH, a diagnostic benchmark that tests this capability. Each item provides a form schema, a prior record, dialogue history, and a new user utterance; the system must return the complete updated record. CONFORMBENCH contains 180 source-grounded synthetic update turns across four institutional form families, covering 339 scoreable fields and 1,683 changed leaf-field commitments. We mechanically validate all items and audit a coverage-controlled 60-item subset through blind independent reconstruction and discrepancy adjudication. We also audit the evaluator against the adjudicated labels. On the 144-turn held-out split, five systems achieve 95.1–97.0% all-field accuracy, yet only 18.1–33.3% of records are entirely correct. Errors concentrate on clearing values, using history, coordinating gates, routing repeated instances, and repairing prior-state errors. Thus, field-level scores substantially overstate record-maintenance reliability.

1 Introduction

Conversational form-filling systems maintain evolving records that downstream systems may treat as the current case state. After each turn, the system must decide what to write, preserve, revise, or remove. Suppose a prior medical intake lists a medication as active. If the user says they stopped taking it, preserving it is an error; an unrelated symptom leaves the medication entry unchanged. The system must reconcile the schema, prior record,

dialogue history, and current utterance into the complete next record. We call this capability *record commitment fidelity*: every retained, changed, or cleared field in the post-turn record must be supported by the schema and conversational evidence. The supplied prior record is incumbent state to reconcile with dialogue history: retain commitments that remain supported, repair those for which evidence determines a different value, and clear those that are withdrawn, contradicted, or unsupported by the public packet. Systems violate this contract when they write plausible but unsupported values, carry over stale commitments, fail to clear withdrawn information, repair the wrong field, or attach a correct value to the wrong repeated instance. Correct handling may therefore leave a field unknown, retain its value, or clear it, depending on the dialogue history. Existing evaluations cover important neighboring abilities, including dialogue state tracking (Budzianowski et al., 2018; Rastogi et al., 2020), structured conversational semantics (Cheng et al., 2020; Andreas et al., 2020; Yu et al., 2019b,a), rule-guided clarification (Saeidi et al., 2018), document and form understanding (Jaume et al., 2019; Stanisławek et al., 2021; Mathew et al., 2021), and GUI or tool-using agents (Li et al., 2025; Yao et al., 2025; Lu et al., 2025; Ma et al., 2024; Patil et al., 2025). These benchmarks test belief states, parses, document answers, tool calls, or end-to-end task success. CONFORMBENCH isolates the record-update step: given an explicit prior form record and public conversational evidence, produce the full resulting record and score both missed supported updates and unsupported commitments. We introduce CONFORMBENCH, a diagnostic benchmark for record commitment fidelity in conversational form filling. It contains 180 synthetic update turns grounded in forms for crash reporting, grant administration, patient administration, and patent disclosure. For each turn, a system receives the form schema, prior record,

dialogue history, and a new user utterance, and returns the full post-turn record. Across all turns, the gold records contain 1,683 leaf-field changes relative to the prior records; a held-out architecture run is evaluated on approximately 9,800 leaf fields in total. The items test additions, refinements, corrections, retractions, gates, repeated instances, history-grounded updates, tempting unsupported commitments, and silent mismatches between the prior state and dialogue history. We also make the benchmark auditable. Non-author annotators independently reconstructed the resulting records for a predefined 60-item subset spanning our four form families and the benchmark stressors. They saw the same public inputs as the evaluated systems. We manually adjudicated every field discrepancy between an independent reconstruction and the gold record, distinguishing annotator errors from normalization problems, ambiguities in the schema or commitment contract, and errors in the gold record. When adjudication exposed a benchmark problem, we repaired the affected artifact before release. The evaluator uses deterministic normalization and constrained open-text scoring, stores its decision traces, and is audited against the adjudicated human labels. Alongside the data and scorer, we release baseline runners, candidate outputs, audit artifacts, extension guidelines, and CONFORMBENCH STUDIO¹. Studio is a browser interface for inspecting items and results and for adding new schemas and scoring policies. Appendix G summarizes the released artifacts. On the 144-item held-out split, five GPT-5.4-mini systems reach 95.1–97.0% all-field accuracy, but only 26–48 of 144 records are exact. Strict changed-field F_1 ranges from 78.5% to 86.0%. Failures concentrate around clearing, history use, gates, repeated-instance identity, and silent state-history repair, showing how preservation-heavy field metrics can overstate record-maintenance reliability. Our contributions are threefold:

1. We define *record commitment fidelity* and introduce CONFORMBENCH, a 180-turn diagnostic benchmark. Each item combines a schema, prior record, dialogue history, current utterance, and complete gold record. Gold records contain 1,683 leaf-field changes; we release stressor annotations and a fixed development/test split.
2. We validate a 60-item subset through indepen-

dent reconstruction and adjudication, audit the evaluator against the resulting human labels, and release the decision traces, baseline artifacts, extension guidelines, and CONFORMBENCH STUDIO needed to inspect and extend the benchmark.

3. Across five architectures, we show that high all-field accuracy conceals frequent failures of the complete record, especially for correction, retraction, history repair, gates, and repeated-instance routing.

2 Related Work

Dialogue state and conversational semantics.

Dialogue state tracking benchmarks evaluate how conversational systems maintain slot–value state over services or task schemas, as in MultiWOZ and the Schema-Guided Dialogue dataset (Budzianowski et al., 2018; Rastogi et al., 2020). Subsequent work has also shown that aggregate state metrics can obscure errors in the actual state changes a system makes (Aksu and Chen, 2024). Related semantic-parsing benchmarks, including TreeDST (Cheng et al., 2020), SMCaFlow (Andreas et al., 2020), SParC (Yu et al., 2019b), and CoSQL (Yu et al., 2019a), extend this line to hierarchical states, dataflow programs, and contextual parses. These benchmarks establish the importance of conversational context and cumulative state and typically evaluate belief states, programs, queries, or execution results. CONFORMBENCH evaluates the full post-turn form record produced from an explicit prior record, dialogue history, schema, and current utterance.

Clarification, document understanding, and form extraction.

Rule-guided conversational QA benchmarks such as ShARC study whether a system can answer from rules or identify when more information is needed (Saeidi et al., 2018). Document and form-understanding benchmarks such as FUNSD (Jaume et al., 2019), KLEISTER (Stanisławek et al., 2021), and DocVQA (Mathew et al., 2021) evaluate extraction or question answering over static form-like documents. CONFORMBENCH complements these tasks by evaluating preservation of prior commitments, revision or clearing of defeated values, and avoidance of unsupported new commitments in live institutional records.

¹Code: <https://github.com/tveigel/conformbench>; data: <https://doi.org/10.5281/zenodo.22166024>.

Form-filling and tool-using agents. Recent agent benchmarks evaluate interaction with tools, user simulators, web forms, or task environments, including FormFactory (Li et al., 2025), τ -bench (Yao et al., 2025), ToolSandbox (Lu et al., 2025), AgentBoard (Ma et al., 2024), and BFCL (Patil et al., 2025). These settings ask whether an agent can act successfully in an environment or complete a task. CONFORMBENCH complements them by evaluating the resulting record for supported updates, required preservation, and complete evidence-backed commitments, independent of interaction policy or tool trace.

3 Benchmark

3.1 Task

Each CONFORMBENCH item is a single update step for one structured record. The public packet contains a form schema S , a prior record state r_{t-1} , the dialogue history $h_{<t}$, and the current user utterance u_t . The system must output the full post-turn record $\hat{r}_t$, which is compared with the gold resulting record r_t^{gold} . Prompt-only and tool-using systems are evaluated by the record they produce. The schema is part of the public evidence. It specifies scoreable fields with question text, answer type, options where applicable, field-level instructions, normalization rules, and optional schema-level semantic expectations. Some open-text scoring uses information units, defined in Appendix A.4. Structural schema nodes define groups, gates, branches, and repeated groups. A repeated group instantiates multiple rows of the same child fields, such as one row per involved vehicle, participant, medication, inventor, or agreement. The task separates the evidence for a record from the prior record that must be checked. Let $x_t = (S, P, h_{<t}, u_t)$ denote the public update packet: the schema S , supplied prior record P , dialogue history $h_{<t}$, and current utterance u_t . The evidence boundary $E_t = (S, h_{<t}, u_t)$ consists of the schema and the complete visible dialogue. It applies to every post-turn commitment, including values carried over from P ; presence in the prior record alone is insufficient support. Formally, let $\mathcal{S}(S)$ be the scoreable slots defined by the schema. For each slot s , let $A_t(s)$ contain the normalized values permitted by the schema and supported by the dialogue. A candidate record C is faithful iff $C_s \in A_t(s)$ for every $s \in \mathcal{S}(S)$, after field-specific normalization. In practice, the system retains a prior value when the

evidence continues to support it, replaces it when the evidence determines another value, and clears it when it is withdrawn, contradicted, or unsupported. Throughout the paper, “supported” and “defensible” mean admissible under this contract. Repeated structures make record commitment fidelity stricter than scalar value extraction. A supported value attached to the wrong repeated instance is an incorrect record, even if the value itself is supported somewhere in the dialogue. Gates likewise affect the resulting record because opening or closing a gated section can determine which child fields should be preserved, updated, or cleared. Prior states vary in how much of their content survives this audit. They may be empty (S1), partially correct (S2), partially wrong in a way repaired by current evidence (S3), or silently inconsistent with the dialogue history or schema before the current utterance is processed (S4). S4 includes both direct state–history contradictions and prior commitments that are simply unsupported by any public evidence. Gold records follow the same evidence-auditing rule: the resulting record contains supported commitments, removes unsupported or withdrawn commitments, and repairs corrupted prior values only when the public packet admits a unique result.

3.2 Dataset

CONFORMBENCH combines compact adaptations of public forms and domain standards with authored synthetic dialogues and diagnostic update scenarios. The frozen benchmark contains 180 items across four institutional form families: crash reporting, grant administration, patient administration, and patent disclosure. Appendix A.1 summarizes the source provenance of the schemas. Items were constructed target-first. The authors first specified each scenario, target fields, prior and gold records, required transitions, dialogue structure, and evidence placement. For each item, the authors then selected either Claude Opus 4.7 or GPT-5.4 without a prespecified assignment rule, based on drafting preference. The model supplied surface phrasing only. The authors manually revised every dialogue, checked it against the predetermined transitions, and reviewed its compatibility with source-derived field semantics, options, conditional structure, and relevant form-filling roles. Each of the 180 items has a distinct current utterance and full dialogue transcript, although generic opening messages may repeat. Eight non-author annotators subsequently audited the coverage-controlled



Figure 1: One CONFORMBENCH item (crash-report form, S3 prior state). A single user utterance supports four operation types: *correct* overwrites a contradicted value, *set* fills previously empty fields, *retract* clears values whose evidence is withdrawn, and *preserve* retains consistent entries. The system must emit the complete 25-field record. Free-text fields use information units for partial-credit scoring (Appendix A.4).

Table 1: CONFORMBENCH at a glance, computed from the frozen 180-item split metadata. See the appendix for the full schema provenance, schema formalism, stressor definitions, and frozen-set metadata distributions.

Property	Value
Items and domains	180 total; 45 each for crash reporting, grant administration, patient administration, and patent disclosure; 36 development / 144 held-out test
Schema scale	339 scoreable regular fields; 19 repeated groups; 29 gates; 1 branch
Dialogue context	1–27 prior messages before the current utterance; 0–13 prior user turns; 81 history-required items
Gold changes	1,683 gold-changed leaf-field commitments; 1,034 inside repeated groups
Prior-state conditions	S1 empty 22; S2 partial-correct 71; S3 wrong prior repaired by current evidence 49; S4 silent state–history/schema mismatch 38
Primary update families	Addition 73; refinement 33; correction 46; retraction 28
Repeat involvement	none 59; single repeated group 68; multiple repeated groups 53

60-item subset described in Section 3.3. Items were authored against a coverage plan fixed before any system outputs were observed. The plan covers record-maintenance phenomena that aggregate state metrics can obscure: additions, refinements, corrections, retractions, local and history-grounded evidence, incomplete or wrong prior states, gates, repeated-instance routing, uncertainty, distractors, and tempting unsupported commitments. We use

these labels to check coverage and analyze errors; system scores come from the full resulting records. Statistical comparisons keep all field decisions from an item together as outcomes of one record-update turn. We separately report the number of fields and changed commitments involved in the full-record task. For each item, the release includes the public packet, the gold resulting record, and the field-level evidence for every edit to the prior record. Its metadata records the update family, condition of the prior state, location of the supporting evidence, and whether the turn involves difficult routing, gates, repeated instances, distractors, or opportunities to add unsupported information. The primary update-family labels summarize benchmark coverage. Because many items combine several operations, operation-level analyses use the actual field transitions. Appendices B.2 and B.3 give the full stressor definitions and frozen-set distributions.

3.3 Gold Validation

Gold validation follows the benchmark’s commitment contract: another reader must be able to recover the resulting record from the public packet. All 180 items undergo mechanical schema and materialization checks, and we verify the full benchmark against the predefined coverage plan. For a deeper audit, non-author annotators independently

Table 2: Independent recoverability audit. Non-author annotators recover most changed fields, while exact full-record reconstruction remains difficult; adjudication required only three author-gold revisions.

Audit signal	Result
Audit size	60 items; 120 independent post-turn records from non-author annotators
Scored burden	920 changed-field decisions plus 7,336 preserve decisions
Changed fields recovered	747/920 strict (81.2%); 763/920 lenient (82.9%)
By edit type, strict	set 405/498 (81.3%); change 217/270 (80.4%); clear/retract 125/152 (82.2%)
One-pass human reference	48/120 strict exact records (40.0%); 52/120 lenient (43.3%)
Gold repair after adjudication	3 author-gold revisions; normalization, schema, contract, and IU issues repaired before final scoring

reconstructed a 60-item subset. The subset contains 15 items from each form family, with two annotators assigned to each family, yielding 120 independent post-turn records. Annotators saw the same public inputs as the systems, without access to gold records, stressor labels, author notes, or one another’s submissions. For each changed field, they entered the resulting value and its supporting evidence; for repeated structures, they also decided which row should receive the value. After annotation, we froze the submissions and converted them into full records. When a submission differed from the gold record, the annotator first rechecked the public packet without seeing the gold. The gold author and both non-author annotators then resolved persistent disagreements together. We repaired gold records, normalization rules, schemas, the commitment contract, or scoring rubrics when the evidence warranted a change. The reported human results use the frozen submissions from before these discussions. We did not conduct a separate second-adjudicator pass. Recovery varied substantially between annotators. In the patent-disclosure subset, one annotator strictly recovered 133/138 changed fields (96.4%), whereas the other recovered 93/138 strictly (67.4%) and 94/138 with lenient credit (68.1%). This gap suggests that the public-evidence contract permits high changed-field recovery, but that performance is sensitive to careful contract application, interface burden, and the demands of managing a complete record. These percentages concern changed fields; the corresponding exact-record rates were 60.0% and 26.7%. Most fields remain unchanged on a

turn, so overall agreement would be dominated by straightforward preservation decisions. The audit therefore reports changed-field recovery and exact records. Annotators recovered most changed fields, although fewer than half of the blind independent reconstructions were exact; a single scoreable error prevents an exact-record match. Adjudication traced most failures to localized omissions, wrong clears, routing decisions, or incomplete open-text answers. Three discrepancies required revisions to the author gold records. The remaining discrepancies were either adjudicated as annotator errors or led to repairs of normalization, schema, contract, or IU artifacts before release.

3.4 Scoring and Evaluator Audit

The evaluator compares three complete record states for each item: the prior record P , gold resulting record G , and candidate resulting record C . When an architecture produces patches, tool calls, or another intermediate representation, the run wrapper first applies them to the prior state to obtain C . For repeated groups, a constrained LLM-assisted procedure aligns the frozen candidate rows with the gold rows before their child fields are scored. Appendix D describes the alignment inputs, treatment of unmatched rows, and stored traces. Most field decisions are deterministic. Booleans, numbers, dates, closed choices, empty values, whitespace- and case-normalized strings, accepted deterministic alternatives, and controlled sets or lists are scored without a language-model call. If the gold value differs from the prior and the candidate preserves the prior value, the field is marked incorrect as a missed supported update. Open-text and schema-alias cases may use the constrained language-model comparator, which logs each decision as evaluator policy. Across the held-out runs, 1.78% of field checks receive a direct semantic LLM verdict, and 25.46% pass through either semantic comparison or LLM-assisted row alignment. Robustness analyses apply pessimistic and credited treatments to these decisions. The gap between field and record accuracy remains, although crediting LLM-assisted fields reverses the close ordering of Update Tool and Diff-then-Verify (Table 14–16). Partial verdicts identify auditable near misses and enter the lenient diagnostics and validation analyses. For example, a candidate may remove a contradicted prior value but omit part of the supported replacement, or it may satisfy only part of an open-text rubric. Strict metrics count

only *correct* verdicts. Because evaluator errors would directly affect the reported system scores, we test the evaluator against the adjudicated human labels. We apply the normal evaluator to the 120 independent submissions from the 60-item audit subset and compare its field verdicts with the adjudicated labels. Across nine saved evaluator runs, this produces 3,906 comparisons. Agreement per run ranges from 98.6–99.8% for the exact labels and from 99.1–100.0% for the binary distinction between *correct* and non-correct. Across all runs, deviations involve 12 distinct exact-label cases and 9 distinct binary cases. The remaining disagreements concern normalization and the boundaries between correct, partial, and incorrect, especially for open text. Appendix D provides the complete evaluator specification, decision sources, deviation cases, scoring-route accounting, and sensitivity analyses for model-assisted decisions. Our primary metrics are exact-record success and changed-field precision, recall, and F_1 . A record is exact when every normalized field verdict is correct, so every post-turn commitment is supported by the public packet and schema. This metric captures evidence fidelity; estimating operational harm requires domain-specific severity weights. We report all-field micro scores and preservation behavior as context because these preservation-heavy views can make weak update behavior appear reliable. After fixing the field verdicts, we count missed supported updates, unsupported commits, collateral edits, failed corrections and retractions, repeat-group alignment errors, gate inconsistencies, and failures to repair silent state–history mismatches. Appendix D.7 gives the formal metric definitions.

4 Experimental Setup

The baseline runs use the frozen 180-item benchmark described in Section 3.2. We create a deterministic 20/80 split with 36 development items and 144 held-out test items, selected from frozen meta-data only and balanced for form domain and primary update family. It also approximately matches prior-state, history, repeat-group, load, conflict, and support-distance coverage, without using model outputs or scores. Prompt and architecture adjustments use the development split; headline metrics use the held-out split. Each item is evaluated independently from the schema, dialogue history, prior record, and current utterance.

Table 3: Baseline architectures. The rows differ in how the system expresses candidate record commitments, not in the evidence available to the system.

Architecture	Interface	Question
Direct JSON	Full record	Can the model perform the update contract directly?
Update Tool	Sparse writes	Does sparse editing reduce collateral edits?
Diff-then-Verify	Sparse diff plus one verification pass	How does a two-stage sparse-diff baseline perform?
Split Agent	Partitioned writers	Does lower schema load help?
Triage-Briefed Split	Triage plus global writer	Does attention routing help without separate writers?

Baseline architectures. We compare five systems that vary the output and verification interface while holding the record-update task fixed (Table 3). We also run a deterministic NOUPDATE lower bound that copies the prior record; it is excluded from architecture bootstrap comparisons. Direct JSON emits the complete record in one response. Update Tool emits sparse `update_questionnaire_answers` calls whose omitted paths are materialized from the prior state. Diff-then-Verify materializes Update Tool’s sparse diff, then gives the complete candidate to one fresh GPT-5.4-mini verifier call, which may emit at most one patch. The prompt was frozen on the 36-item development split; the system was evaluated on the 144-item held-out split under the common architecture configuration. Split Agent sends dependency-safe schema partitions to separate workers and merges their sparse updates. Triage-Briefed Split uses the same partitions for triage, then lets one global updater commit sparse updates over the selected schema regions. The controlled architecture comparison holds the public evidence, item order, materialization code, evaluator implementation, and requested evaluator model constant. GPT-5.4-mini generates all system outputs with a maximum output of 32k tokens and no reasoning; both stages of Diff-then-Verify use this configuration. Before scoring, the run wrapper converts every full-record output, sparse write set, or tool trace into a complete post-turn record. One evaluator-runtime detail differs: for compatibility with the frozen runner, the Diff-then-Verify helper omitted the configured `reasoning_effort` parameter before trans-

mission, and its saved comparator traces contain zero reasoning tokens. Appendix E documents this difference, and Appendix D reports the evaluator-route sensitivity analyses. We also run a Direct JSON model-strength ablation with Claude Opus 4.7 (medium reasoning), GPT-OSS-120B (medium reasoning), and Qwen3-32B (provider-default reasoning) using the same evaluator implementation. The release includes candidate records, prompt templates and source prompt builders, tool traces where applicable, evaluator traces, run dates, and requested and provider-resolved model identifiers. Appendix E reports snapshots, settings, prompt scaffolds, tool signatures, and partition rules. We use paired item-level bootstrap confidence intervals for architecture differences, resampling held-out items with replacement and recomputing each metric on the resampled item set. For strict micro F_1 , each item’s changed-field counts are kept together during bootstrap recomputation.

5 Results

We ask whether field-level scores predict reliable records, whether simple architecture changes solve the update contract, and which item shapes remain hard across architectures.

5.1 High Field Accuracy Hides Low Record Fidelity

Table 4 gives the held-out comparison. All five systems reach 95.1–97.0% all-field accuracy, yet only 26/144–48/144 turns are exact records. Direct JSON has the strongest exact-record point estimate. Diff-then-Verify has the highest observed changed-field F_1 and the lowest observed missed-update and unsupported-commitment counts, while Split Agent remains the weakest row. Pairwise bootstrap resampling supports Diff-then-Verify over Split Agent on all three metrics and over Triage-Briefed Split Agent on changed-field F_1 and collateral edits. Exact-record differences from Direct JSON, Update Tool, and Triage-Briefed Split Agent remain too uncertain to rank reliably. Alternative treatments of LLM-assisted evaluator decisions reorder some close systems, but exact-record success remains below 41% (Appendices D and F.4). Table 2 provides a one-pass human reference: 48/120 (40.0%) exact records from 120 blind reconstructions of a separate 60-item subset, with annotator rates of 20.0–60.0%. These observed rates provide context for the system results; the human audit

and 144-item system test use different splits and changed-field measures, and maximum human performance remains unmeasured. Diff-then-Verify obtains 44/144 exact records and 86.0% changed-field F_1 ; Direct JSON retains the highest exact-record point estimate at 48/144. A deterministic NOUPDATE lower bound exposes the preservation bias: copying the prior record gives 85.9% all-field accuracy, 100.0% preservation, and 0 collateral edits, but 0/144 exact records and 0.0 changed-field F_1 . Appendix Table 19 characterizes generator sensitivity under Direct JSON: exact-record success ranges from 20.1% to 36.8% across the tested generators. Model effects may vary by architecture.

5.2 Where Records Break

The same bottlenecks recur across architectures (Table 5). Appendix F reports the transition counts. At the item level, difficulty is also shared across architectures. Across five systems, 69/144 items are solved by at least one, 75 by none, and 12 by all five. Solvability falls with history, repeats, high load, mixed revisions, clear transitions, and S4 mismatches. For high+scale items, observed solvability is 16.7%, versus null predictions of 44.4% by field count and 17.5% by changed/preserve burden. Appendix F reports the full analyses, including the load-normalized results in Appendix F.5.

6 Discussion

The record is the unit of reliability. Complete-record correctness captures the joint reliability of the maintained case state: one missed correction, stale prior value, unsupported write, failed retraction, wrong gate state, or value routed to the wrong repeated instance can make that state unreliable. Preservation-heavy metrics reward the easiest decision while obscuring errors that matter when the record is later treated as authoritative. In downstream use, a wrong committed value can be worse than a blank one because it is presented as current and evidence-backed. Field errors can differ in operational severity. Their domain-specific consequences require future study.

Failures center on state management. Across the tested systems, errors concentrate in clearing, retraction, history repair, gate propagation, and repeated-instance routing. These require explicit clear and retract semantics, provenance for retained values, prior-state auditing, controller-dependent consistency checks, and durable repeated-entity

Table 4: Controlled five-system architecture comparison on 144 held-out items. All use GPT-5.4-mini generation with 32k maximum output tokens and no reasoning, and GPT-5.4 evaluation. Changed P/R/ F_1 is strict. Rate denominators are 1,227–1,229 gold-changed and 9,820–9,834 final-field decisions, respectively.

System	Exact record	All fields	Changed P/R/ F_1	Missed updates	Unsupported commits
Direct JSON	48/144 (33.3%)	96.2%	84.5/80.2/82.3	241 (19.6%)	147 (1.5%)
Update Tool	43/144 (29.9%)	96.7%	87.3/83.0/85.1	214 (17.4%)	141 (1.4%)
Diff-then-Verify	44/144 (30.6%)	97.0%	87.2/84.8/86.0	188 (15.3%)	140 (1.4%)
Split Agent	26/144 (18.1%)	95.1%	76.5/80.6/78.5	243 (19.8%)	281 (2.9%)
Triage-Briefed Split Agent	36/144 (25.0%)	96.5%	84.7/82.9/83.8	211 (17.2%)	166 (1.7%)

Table 5: Diagnostic slices on the held-out test split, not prevalence estimates or stable rankings. Rows give units and denominators; ranges span five GPT-5.4-mini systems.

Failure mode	Unit / denominator	Success range	Main implication
Clear transitions	Aligned leaf fields, 56–59	54.2–67.9%	Systems preserve stale values
History recovery	Applicable items, 67	34.3–40.3%	Prior state is treated as too authoritative
Gate execution	Applicable items, 64	32.8–45.3%	Controllers and dependent fields drift
Retraction	Applicable items, 20	35.0–55.0%	Withdrawn evidence is hard to clear
Repeat execution	Applicable items, 93–95	36.8–52.1%	Instance routing remains brittle
S4 silent mismatch exact	S4 items, 30	6.7–23.3%	Hidden prior-state errors are rarely repaired

identity. Sparse edits reduce collateral rewrites; evidence reconciliation still requires deciding when prior commitments lose support.

Decomposition requires shared state. Split Agent illustrates how partitioning can remove context needed for prior-state repair, gates, and repeated-instance identity, consistent with documented coordination costs in decomposed agents (Kim et al., 2025). Triage-Briefed Split focuses local attention while preserving global commitment decisions and cross-field checks.

7 Conclusion

CONFORMBENCH evaluates conversational form filling as evidence-backed record maintenance from a schema, prior record, and dialogue. Across five tested systems, high field accuracy masks frequent complete-record failures, especially for clearing, history use, gates, and repeated instances. Reliable record maintenance requires retention, correction, and clearing; these capabilities remain open challenges across the tested interfaces.

Limitations

CONFORMBENCH is a stressor-balanced diagnostic benchmark. Its aggregate scores characterize 180 items across four form schemas and are not estimates of real-world error prevalence; some diagnostic slices have modest denominators. The conditional-logic analysis covers 29 gates and one branch, supporting gate-specific conclusions while leaving multiway branch routing and mutual exclusion for a dedicated evaluation slice. The architecture comparison fixes GPT-5.4-mini and its generation configuration across simple prompting and tooling patterns. Architecture-by-model interactions, richer form-flow controllers, retrieval state managers, long-horizon planners, and benchmark-trained systems remain open. Diff-then-Verify adds a second model call, and its frozen evaluator helper omitted the configured reasoning parameter; its effective evaluator reasoning setting therefore differs from the other architecture rows. Per-item dialogue-surface generators were selected without a fixed assignment rule, leaving interactions between benchmark wording and model family uncontrolled. The five-system architecture-overlap and shape-slice analysis is descriptive. Exact-record success measures evidence fidelity; cross-domain harm assessment requires independently validated workflow-specific criticality labels and cost functions. The stressor annotations cover structural and evidence-placement conditions, with linguistic analysis of correction form, negation, coreference, ellipsis, and pragmatic inference left to future work. The English benchmark primarily reflects US and EU institutional forms; dialectal, non-native, and real-user language coverage also remains future work. Record errors may affect the patient, applicant, or other record subject. CONFORMBENCH targets settings in which the form-filling interaction is the audit trail for the maintained record. Deployments that import values from external databases, human clerks, or hidden earlier interactions require those

sources in the evidence packet or additional source-priority rules. Field-local metric counts reflect record consequences, so a gate or repeated-instance routing error can create several downstream errors. The evaluation covers one-step record updates and excludes dialogue policy, next-question selection, user satisfaction, and long-horizon recovery.

Ethical Considerations

CONFORMBENCH uses synthetic dialogues and update scenarios over compact adaptations of public institutional forms; no real patient records, crash reports, grant applications, or invention disclosures are included. Non-author human involvement was limited to independent reconstruction for benchmark validation. The audit was performed by eight expert annotators recruited through professional networks affiliated with the author team: seven PhD candidates and one postdoctoral researcher, with three universities represented. Annotators saw only the public benchmark packet shown to systems: schema, prior record, dialogue history, and current utterance. They did not see provisional gold records, stressor labels, author notes, model outputs, or each other’s submissions. Annotators confirmed in writing that their annotations may be used for research validation and released in pseudonymized form, and participation was voluntary with the option to decline or withdraw annotation artifacts from release. We release pseudonymized annotation reports and adjudication records while keeping personal names, contact information, institutional affiliations, and private coordination records confidential. Annotators received no separate payment. We treated the audit as low-burden voluntary expert peer validation of synthetic benchmark artifacts within professional research networks. The project did not undergo formal ethics-board review. Additional annotation and release details are reported in Appendix C.2.

Use of AI Assistants

The authors used AI assistants for implementation support, code editing, limited prose editing, and dialogue surface realization. As detailed in Section 3.2, the authors fixed each item’s scenario, states, transitions, evidence placement, and labels; Claude Opus 4.7 or GPT-5.4 supplied candidate wording, which the authors manually revised. The authors made all decisions concerning benchmark design, gold records, research questions, statistical analy-

ses, and scientific claims, verified the experimental results, and take responsibility for all released code, artifacts, analyses, and manuscript content.

References

- Taha Aksu and Nancy Chen. 2024. [Granular change accuracy: A more accurate performance metric for dialogue state tracking](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 7939–7948, Torino, Italia. ELRA and ICCL.
- Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, and 26 others. 2020. [Task-oriented dialogue as dataflow synthesis](#). *Transactions of the Association for Computational Linguistics*, 8:556–571.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. [MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.
- Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, Jason D. Williams, Hong Yu, Diarmuid Ó Séaghdha, and Anders Johannsen. 2020. [Conversational semantic parsing for dialog state tracking](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8107–8117, Online. Association for Computational Linguistics.
- Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. [FUNSD: A dataset for form understanding in noisy scanned documents](#). In *2019 International Conference on Document Analysis and Recognition Workshops*, volume 2 of *ICDARW*.
- Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A. Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Yun Liu, Mark Malhotra, Paul Pu Liang, Hae Won Park, Yuzhe Yang, Xuhai Xu, Yilun Du, Shwetak Patel, Tim Althoff, Daniel McDuff, and Xin Liu. 2025. [Towards a science of scaling agent systems](#). *Preprint*, arXiv:2512.08296.
- Bobo Li, Yuheng Wang, Hao Fei, Juncheng Li, Wei Ji, Mong-Li Lee, and Wynne Hsu. 2025. [Formfactory: An interactive benchmarking suite for multimodal form-filling agents](#). In *Proceedings of the 33rd ACM International Conference on Multimedia, MM ’25*, page 13273–13280, New York, NY, USA. Association for Computing Machinery.

- 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. 2025. [ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1160–1183, Albuquerque, New Mexico. Association for Computational Linguistics.
- Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. [AgentBoard: An analytical evaluation board of multi-turn LLM agents](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 74325–74362.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. [DocVQA: A dataset for VQA on document images](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2200–2209.
- 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.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. [Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.
- Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. [Interpretation of natural language rules in conversational machine reading](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.
- Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. [Kleister: Key information extraction datasets involving long documents with complex layouts](#). In *Document Analysis and Recognition – ICDAR 2021*, Lecture Notes in Computer Science, pages 564–579. Springer.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2025. [\$\tau\$ -bench: A benchmark for tool-agent-user interaction in real-world domains](#). In *The Thirteenth International Conference on Learning Representations*.
- Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, and 5 others. 2019a. [CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1962–1979, Hong Kong, China. Association for Computational Linguistics.
- Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019b. [SPaRC: Cross-domain semantic parsing in context](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.

Appendix Contents

G Released Artifacts and CONFORM-BENCH STUDIO

23

A Benchmark Schemas, Provenance, and Information Units	12
A.1 Form-Family Provenance	12
A.2 Schema Node Types	12
A.3 Repeated Groups, Gates, and Branches	12
A.4 Semantic Information Units	12
B Dataset Construction, Stressors, and Frozen Metadata	12
B.1 Item Construction Controls	12
B.2 Human-Coded Stressors	13
B.3 Frozen 180-Item Metadata	14
B.4 Development/Test Split	14
C Gold Validation, Adjudication, and Human Annotation	14
C.1 Independent Reconstruction Protocol	14
C.2 Annotator Recruitment, Consent, and Release Policy	15
D Evaluator, Scoring Routes, and Diagnostic Metrics	16
D.1 Record-State Comparison	16
D.2 Repeat-Group Alignment	17
D.3 Deterministic and LLM-Assisted Field Scoring	17
D.4 Scoring-Route and Alignment-Exposure Accounting	17
D.5 Partial Verdicts and IU Verdict Derivation	17
D.6 Evaluator Audit	17
D.7 Diagnostic Metric Definitions	18
E Baseline Systems, Prompts, and Model Metadata	19
E.1 Architecture Interfaces	19
E.2 Tool Signature and Materialization	20
E.3 Prompt Scaffold	20
E.4 Model Snapshots and Run Settings	21
E.5 Partition Rules	22
F Additional Results, Bootstrap Intervals, and Load-Normalized Analysis	22
F.1 Field-Transition Accuracy	22
F.2 Strict Versus Lenient Scoring	22
F.3 Architecture-Family Solvability	22
F.4 Paired Bootstrap Intervals	22
F.5 Load-Normalized Exact-Record Analysis	23

A Benchmark Schemas, Provenance, and Information Units

A.1 Form-Family Provenance

The four schemas are compact adaptations of public forms or domain standards. The crash-report schema draws on MMUCC, CADaS, ANSI D16.1, UNECE/WHO road-safety definitions, and the European Accident Statement. The grant-administration schema draws on European Commission Horizon Europe RIA/IA application materials. The patient-administration schema is grounded in SNOMED CT, HL7 FHIR R4, WHO IMAI, and CDC outpatient screening constructs. The patent-disclosure schema adapts the UCI Record of Invention disclosure form. We simplify these sources into benchmark schemas while preserving institutional structure relevant to record maintenance: repeated vehicles, grant participants and work packages, allergies and medications, inventors, funding sources, agreements, and disclosures. The dialogues and update scenarios are synthetic diagnostic intakes constructed target-first and reviewed for source compatibility and role plausibility as described in [Section 3.2](#). [Table 6](#) gives compact schema cards for the four forms. The released benchmark includes complete machine-readable JSON schemas with the full field-level form text, answer types, options, field policies, normalization rules, gates, repeat definitions, and schema-level information units.

A.2 Schema Node Types

The public questionnaire schema is a tree stored under a top-level questions array. Only *regular* nodes are scoreable fields; the other node types are structural. The frozen benchmark uses four questionnaire schemas: crash report, grant administration, patient administration, and patent disclosure. Together, they contain 339 scoreable regular fields, 19 repeated groups, 29 gates, and one branch.

Regular nodes. A regular node is an atomic scoreable field. It specifies a field identifier, question text, answer type, and field policy describing the intended semantic scope of the answer. Depending on type, it may also specify closed options, normalization rules, deterministic alternatives, or schema-level expectations.

Group nodes. A group is a semantic container for related fields, such as a road-type section containing location type, posted speed limit, and traffic

control. They do not themselves collect answers.

A.3 Repeated Groups, Gates, and Branches

Repeated groups. A repeated group instantiates the same child fields multiple times, for example, one row per vehicle, participant, medication, inventor, agreement, or funding source. The number of instances can be fixed by schema design or derived from a controller field, such as the number of vehicles involved in a crash. Each instance is independently scoreable, and correct record maintenance requires routing values to the correct instance.

Gates and branches. A gate conditionally activates child fields based on a controller field, whereas a branch routes to one of several mutually exclusive subflows. The schemas contain 29 gates and one branch. The analysis, therefore, focuses on gates; the branch contributes schema coverage.

A.4 Semantic Information Units

Some free-text fields have schema-level information units (IUs). IUs are small, inspectable semantic components expected by a field. They are used to make free-text recoverability and scoring auditable. Schema-level IUs are authored independently of benchmark items; item-level IU gold records which IUs are required for a specific gold value, the expected content for each IU, and the supporting evidence span. IUs do not replace the field value. They expose why a free-text value is complete, incomplete, contradicted, or over-specified. For example, a non-vehicle property-damage field may expect the damaged object, damage extent, and scene relation. If the utterance says “the impact bent the metal mailbox at 14 Market Street,” the item-level IU gold can record the following evidence-backed commitments.

For IU-scored fields, the constrained comparator reports structured facts about coverage, contradiction, improvement over the prior value, and unsupported or field-irrelevant extra content. Deterministic rules then derive the final field verdict, as summarized in [Table 8](#).

B Dataset Construction, Stressors, and Frozen Metadata

B.1 Item Construction Controls

Items are authored under pre-specified coverage controls before any system outputs are observed. The controls cover form family, prior-state condition, primary update family, revision operation, his-

Table 6: Compact schema cards for the four CONFORMBENCH form families. Summarized are the form surfaces used in the benchmark; the released JSON schemas contain the full field-level form text and scoring policies.

Form family	Source basis	Schema size	Main structure preserved in the benchmark schema
Crash report	MMUCC, CADaS, ANSI D16.1, UNECE/WHO road-safety definitions, European Accident Statement	25 fields; 1 repeated group; 3 gates; 0 branches	Crash circumstances, road type and traffic controls, property damage, and repeated vehicle rows with role, vehicle type/make/model, licence plate, pre-crash manoeuvre, approximate speed, and damage.
Grant administration	Horizon Europe RIA/IA application materials	174 fields; 7 repeated groups; 10 gates; 1 branch	Submission and call context, declarations, participants, budget, ethics/security, Part B excellence/impact/implementation, annexes, and repeated rows for participants, work packages, deliverables, milestones, risks, subcontracting, and equipment.
Patient administration	SNOMED CT, HL7 FHIR R4, WHO IMAI, CDC outpatient screening constructs	49 fields; 4 repeated groups; 8 gates; 0 branches	Pain and respiratory symptoms, falls and mobility, oral health and vision, nutrition and swallowing, and repeated rows for pain details, allergies, current medications, and dietary preferences or restrictions.
Patent disclosure	UCI Record of Invention disclosure form	91 fields; 7 repeated groups; 8 gates; 0 branches	Funding sources, agreements, proprietary materials, relevant invention dates, public disclosures, invention description, signatures/witnesses, and repeated rows for UCI inventors, non-UCI inventors, funding sources, agreements, materials, potential licensees/sponsors, and witnesses.

Table 7: Example information-unit decomposition for an open-text property-damage field.

IU	Gold content	Evidence span
Damaged object	metal mailbox	“metal mailbox”
Damage extent	bent	“bent”
Scene relation	at 14 Market Street	“at 14 Market Street”

tory requirement, evidence locality, conflict, repeat-group involvement, form-relative load, and the human-coded stressors below.

B.2 Human-Coded Stressors

Stressors are fixed before system runs and support coverage tracking and stratified error analysis. They do not enter scoring. The stressors below are manually coded because they require human interpretation, and every non-baseline value requires an annotation note identifying the observable cue.

1. **Target binding pressure.** Describes how much context is required to identify the intended field, entity, event, or prior mention in the general non-repeat case. Levels are: *none*, *implicit unique target*, *prior mention target*, and *competing candidates*. This stressor excludes repeated-row routing, which is coded separately.
2. **Commitment boundary pressure.** Describes whether the item stresses the boundary between a supported structured commitment and conservative abstention. Levels are: *none*, *partial*

value, *uncertain or qualified*, and *unresolved scope or target*. This stressor concerns whether evidence supported a commitment at all, not which target the commitment belongs to.

3. **Distractor competition.** Describes whether explicit schema-compatible non-gold content competes with the intended update. Levels are: *none*, *weak distractor*, *same-schema-region distractor*, and *competing entity or group*. It requires explicit competing content in the public packet.
4. **Unsupported-alternative affordance.** Describes whether the public packet contains salient non-entailed alternatives that a system might be tempted to commit. Levels are: *none*, *single-slot affordance*, *multi-slot affordance*, and *cross-entity or default affordance*. This stressor is metadata rather than a scoring rule.
5. **Repeat-instance routing pressure.** Describes whether correct interpretation must distinguish, preserve, create, attach, or avoid edits to repeated structured instances or groups. Levels are: *none*, *unique instance*, *same-group competition*, and *cross-group or multi-instance*. This stressor may be non-zero even when the gold diff does not touch a repeat-group leaf, because preserving a repeated structure or avoiding an unsupported repeated-row edit can be the intended pressure.
6. **State-history mismatch pressure.** Describes whether the supplied prior state silently dis-

Table 8: Verdict derivation for semantic-IU fields. The comparator reports structured IU facts; deterministic rules derive the final verdict.

Condition	Verdict
Any required IU is contradicted	incorrect
All required IUs are fully covered and no bad extra is present	correct
All required IUs are fully covered but grounded irrelevant or dumped content is present	partial
Some required IUs are missing, but the candidate newly adds or improves at least one required IU	partial
Candidate is closer to gold than preserving the prior value, but still incomplete	partial
Unsupported extra appears together with otherwise useful progress	partial
No IU progress, no field-level improvement, or unsupported-only answer	incorrect

agrees with the visible conversation or schema before the current utterance is processed. Levels are: *none*, *silent scalar mismatch*, *silent structural mismatch*, and *silent repeat or cross-group mismatch*. This stressor is distinct from explicit correction: the dialogue does not point out the bug, but the mismatch is recoverable by comparing visible history, prior state, and schema.

To keep the labels non-overlapping, we apply four separation rules. First, target binding pressure covers general non-repeat binding, whereas repeat-instance routing pressure covers repeated-row routing. Second, commitment boundary pressure asks whether evidence supported a commitment, whereas unsupported-alternative affordance captures salient non-entailed alternatives. Third, distractor competition requires explicit competing content, whereas unsupported-alternative affordance may remain non-zero without an explicit distractor. Fourth, state-history mismatch pressure captures silent disagreement between the public conversation and supplied prior state; it is not distractor competition because the competing value is a record-state error, not an explicitly asserted alternative in the dialogue. Changed-field count, primary update family, prior-state condition, history requirement, support distance, conflict presence, repeat-group involvement, and revision operation are materialized directly from the gold annotation contract.

B.3 Frozen 180-Item Metadata

Table 9 and 10 report the item metadata distribution for the frozen 180-item benchmark from which the development and test splits are drawn. These counts are model-independent item descriptors.

B.4 Development/Test Split

We create a deterministic 20/80 split with 36 development items and 144 held-out test items. The

split is selected from frozen metadata only and is balanced for form domain and primary update family, while approximately matching prior-state condition, history requirement, repeat-group involvement, load, conflict, and support-distance coverage. Prompt and architecture changes are made on the development split; all headline metrics use the held-out split.

C Gold Validation, Adjudication, and Human Annotation

C.1 Independent Reconstruction Protocol

The independent reconstruction audit uses a 60-item coverage-controlled slice of the frozen benchmark, with 15 items from each form family. Each form stratum is annotated by two non-author annotators, yielding 120 independent post-turn records. Annotators see the same public packet as systems: schema, prior record, visible history, and current utterance. They do not see provisional gold records, stressor labels, author notes, or each other’s annotations. For changed fields, annotators enter the resulting value and supporting evidence. For repeated structures, they also make row-routing decisions. For information-unit-scored fields, recoverability follows the rubric in Appendix A.4. After annotation, each independent submission was frozen and materialized into a full post-turn record. For every non-exact field comparison, the differing contributor first rechecked the public packet blindly, without seeing the other submission or provisional gold. Persistent disagreements were then resolved by the gold author and both non-author annotators assigned to the form stratum. They determined whether the public packet supported the annotator value, the provisional gold value, both as acceptable variants, or neither. Provisional gold and other artifacts were revised when warranted. We report the frozen pre-discussion submissions; there was no second-adjudicator pass.

Table 9: Derived metadata distribution in the frozen 180-item benchmark.

Descriptor	Distribution
Form domain	crash report 45; grant administration 45; patient administration 45; patent disclosure 45
Prior-state condition	S1 empty 22; S2 partial-correct 71; S3 partial-incorrect 49; S4 silent mismatch 38
Primary update family	addition 73; refinement 33; correction 46; retraction 28
Revision operation	append 44; refine 16; overwrite 27; retract 16; mixed 77
Form-relative load bucket	low 51; medium 83; high 30; scale 16
Evidence locality	no history required 99; history required 81; support distance 0/1/2/3 = 99/16/7/58
Visible history length	1–27 prior messages (median 5); 0–13 prior user turns (median 2)
Conflict present	false 130; true 50
Repeat-group involvement	none 59; single group 68; multi group 53
Changed leaf fields	min 1; median 6; max 89; total changed leaf fields 1,683
Repeat-group changed leaf fields	min 0; median 3; max 81; total repeat-group changed leaf fields 1,034

Table 10: Human-coded stressor distribution in the frozen 180-item benchmark.

Stressor	Distribution
Target binding pressure	none 84; implicit unique target 7; prior mention target 64; competing candidates 25
Commitment boundary pressure	none 140; partial value 11; uncertain or qualified 24; unresolved scope or target 5
Distractor competition	none 104; weak distractor 14; same-schema-region distractor 24; competing entity or group 38
Unsupported-alternative affordance	none 99; single-slot affordance 13; multi-slot affordance 42; cross-entity/default affordance 26
Repeat-instance routing pressure	none 86; unique instance 19; same-group competition 26; cross-group or multi-instance 49
State-history mismatch pressure	none 147; silent scalar mismatch 9; silent structural mismatch 10; silent repeat or cross-group mismatch 14

Gold-record revisions. A gold-record revision is made only when adjudication shows that public evidence supported a value omitted or misstated by provisional gold. For example, in a grant-administration item, an annotator recovered an organisation webpage from a stated Gender Equality Plan policy URL. The provisional gold omitted the homepage, but adjudication accepted it as inferable from public evidence and repaired the gold record. Such cases are rare: the independent audit required three author-gold revisions.

Rubric repairs. A rubric repair exposes a semantic distinction needed for auditable comparison while leaving the recovered record value unchanged. For example, if an information-unit rubric omits a required witness-observation distinction, the rubric is refined so that the expected commitment becomes visible and scoreable.

Normalization and contract repairs. Some disagreements expose an underspecified normalization or contract rule. These repairs clarify accepted variants, empty-value equivalence, field-specific formatting, or the boundary between a supported commitment and conservative abstention. Rows affected by such repairs are rechecked before release.

Table 11 and 12 summarize the disagreement pat-

terns exposed by the independent audit. A single local field error blocks strict exact-record recovery. Across pairs, only 22 fields were missed by both annotators after adjudication, while 132 incorrect sides were one-sided errors.

Interface burden. Annotator feedback identified interface burden as a possible source of human-side misses. The independent annotation interface was compact, and annotators reported difficulty maintaining a global overview of all fields while editing individual commitments. This may have contributed to one-sided omissions or repeated-instance routing slips. On the patent-disclosure stratum, one annotator recovered 133/138 changed fields strictly (96.4%), while the paired annotator recovered 93/138 strictly (67.4%; 94/138 under lenient credit, 68.1%). This within-stratum gap points to annotator-side cognitive load, interface burden, and local omissions as contributors to independent-audit errors.

C.2 Annotator Recruitment, Consent, and Release Policy

Annotator recruitment, consent, and compensation. The independent reconstruction audit used eight non-author expert annotators recruited through professional networks affiliated with the

Table 11: Independent reconstruction disagreement anatomy by annotator pair. A scored side is one adjudicated annotator-side field decision after materialization into a full record. Shared-error fields are fields where both annotators in a pair were incorrect after adjudication; one-sided error sides are errors made by only one annotator. Exact records use strict full-record correctness. The patent-disclosure pair shows the largest within-stratum asymmetry: 133/138 changed fields strictly recovered by the stronger annotator versus 93/138 by the paired annotator (94/138 under lenient credit).

Pair	Scored sides	Reviewed rows	Incorrect sides	Shared-error fields	One-sided error sides	Exact records strict	Changed fields strict
A/B	4,014	100	76	15	46	9/30 (30.0%)	253/318 (79.6%)
C/D	1,266	36	17	3	11	17/30 (56.7%)	113/140 (80.7%)
E/F	882	39	32	4	24	9/30 (30.0%)	155/186 (83.3%)
G/H	2,094	60	51	0	51	13/30 (43.3%)	226/276 (81.9%)
Total	8,256	235	176	22	132	48/120 (40.0%)	747/920 (81.2%)

Table 12: Outcomes for 235 manually adjudicated disagreement rows. Most were annotator-side errors or normalization repairs; only three required author-gold revisions.

Adjudication outcome	Rows
Annotator error	200
Normalization gap	21
Contract gap	7
Schema gap	4
Author-gold revision	3

author team. Seven annotators were PhD candidates and one was a postdoctoral researcher, with three universities represented. None authored the audited items. Annotators worked independently and did not see provisional gold records, stressor labels, author notes, model outputs, or another annotator’s output. Consent and recruitment were documented in written Slack, Teams, or Matrix messages. Annotators confirmed that their annotations may be used for research validation and released in pseudonymized form. The authors retain private coordination messages for accountability. The public archive uses pseudonymous annotator identifiers and omits names, email addresses, institutional affiliations, private messages, and payment or coordination records. Annotators received no separate payment. The audit was conducted as voluntary expert peer validation of synthetic benchmark artifacts within professional research networks. Participation was low burden and voluntary, and annotators could decline or withdraw their annotation artifacts from release. The paper reports coarse professional role, recruitment source, workload, and institutional coverage. Approximate task times appear below.

Table 13: Human annotation workload for the 60-item independent recoverability audit. Times are approximate per-annotator task times.

Form family	Annotators	Minutes
Crash report	2	100, 60
Patient administration	2	60, 50
Grant administration	2	120, 160
Patent disclosure	2	140, 120

Instructions and interface. The supplementary materials include the full annotator instructions, reliability protocol, and calibration/adjudication procedures. Annotators used the software, an adapted clone of CONFORMBENCH STUDIO. The instructions tell annotators to start from the prior record, change only fields supported by the public packet, preserve every other field, attach evidence to every non-preserve transition, resolve repeat routing and gate/controller consequences, and flag ambiguity or contract gaps rather than guessing.

Released annotation artifacts. The public artifact bundle includes the locked 60-item audit-slice manifest, pseudonymized annotation summaries, adjudication tables, evaluator-comparison reports, and side-label overrides. Personal annotator names and private communication remain confidential. The pseudonymized summaries and adjudication CSVs are the paper-facing source of truth for the reported human audit.

D Evaluator, Scoring Routes, and Diagnostic Metrics

D.1 Record-State Comparison

The evaluator is a record-state comparator. For each turn it compares three states: the prior record P , the gold resulting record G , and the candidate

resulting record C . The evaluator consumes a complete schema-conforming candidate record. If a system uses patches or tool calls, the run wrapper materializes them into C before evaluation. Assistant prose and tool traces are not scored.

D.2 Repeat-Group Alignment

Before repeat-group child fields are scored, gold rows are aligned to candidate rows in a constrained post-hoc evaluator step. The candidate record is materialized and frozen first. The LLM matcher receives the public dialogue context, prior rows, gold rows, candidate rows, identity anchors, alignment-key hints, and schema notes, and returns a one-to-one mapping plus explicit unmatched-gold and unmatched-candidate buckets. It uses row identity, with exact identifiers such as licence plates, emails, proposal IDs, participant numbers, and PICs decisive when present, and leaves unsupported or ambiguous matches unresolved. Matched rows retain their candidate values and are scored field by field; unmatched gold rows yield missing-instance child-field errors, and unmatched candidate rows yield spurious-instance child-field errors for non-empty children. Released traces store alignment inputs, outputs, mappings, and verdicts.

D.3 Deterministic and LLM-Assisted Field Scoring

Each scalar field and aligned repeat child follows the same route. First, if G differs from P but C preserves P , the field is marked incorrect as a missed supported update. Second, if C matches G after field-specific normalization, the field is marked correct. Third, reliable deterministic comparison covers booleans, numbers, dates, closed choices, empty values, normalized strings, accepted deterministic alternatives, and controlled unordered sets or lists. Finally, eligible open-text, schema-alias, or semantic-IU mismatches are escalated to a constrained comparator. Non-IU text comparisons follow the field policy; IU-specific comparison and verdict derivation are defined in Appendix A.4. Fields lacking item-level IU gold bypass generic semantic-IU judgment. The comparator applies the gold rubric without changing candidate values.

D.4 Scoring-Route and Alignment-Exposure Accounting

Final-verdict provenance and broader pipeline dependence are distinct. Across the five held-out runs, 876/49,128 checks (1.78%) receive direct

semantic verdicts. Separately, 12,058 checks (24.54%) are child fields compared after an actual LLM row-alignment call. Because 426 checks follow both routes, their union is 12,508/49,128 (25.46%); the remaining 36,620 checks are fully deterministic. Formally, an alignment call selects a partial one-to-one mapping M from gold rows to frozen candidate rows. Candidate values remain fixed, each candidate row maps to at most one gold row, and unmatched rows receive fixed missing/spurious-instance treatment. Conditional on M , non-semantic child verdicts are ordinary field comparisons $v_f(g_j, c_{M(j)})$. The alignment-exposure count comprises 12,058 downstream field comparisons dependent on 1,033 constrained mapping calls. Pessimistic and credited treatment of direct semantic fields leave every exact-record rate below 41%. Crediting every alignment-dependent field raises exact-record success to 27.8–53.5%; alignment-independent field accuracy is 95.9–97.7%. Crediting LLM-assisted fields reorders the close Update Tool and Diff-then-Verify rows while preserving the qualitative field-record gap and the strongest and weakest rows. The main pattern persists under both LLM-assisted routes. These bounds characterize aggregate sensitivity; the 98.6–99.8% exact-label agreement against adjudicated human judgments in Table 16 provides complementary evidence about verdict reliability.

D.5 Partial Verdicts and IU Verdict Derivation

Partial verdicts are retained as auditable near misses and for lenient sensitivity analyses. Strict headline metrics and fully correct turns require a *correct* verdict. For semantic-IU fields, the constrained comparator reports IU coverage, contradiction, improvement over the prior value, and unsupported or field-irrelevant extra content. Deterministic rules then derive the final field verdict, as summarized in Table 8.

D.6 Evaluator Audit

We validate the evaluator using the independent reconstruction submissions reported in Section 3.3. Each annotator’s submission is treated as a candidate system output and scored against the released gold. The resulting automated field verdicts are compared with adjudicated human-review labels. A side-field check is one adjudicated annotator-side field comparison after repair rows and excluded

Table 14: LLM-stage exposure in the five held-out runs. Alignment-dependent checks are child fields scored after an actual LLM row mapping. Any-stage exposure is their union with direct semantic verdicts; fully deterministic is its complement.

Architecture	Final field checks	Direct semantic verdict	Alignment-dependent	Any LLM-stage exposure	Fully deterministic
Direct JSON	9,826	178	2,368	2,462	7,364
Update Tool	9,824	169	2,416	2,500	7,324
Diff-then-Verify	9,820	163	2,429	2,507	7,313
Split Agent	9,834	192	2,416	2,519	7,315
Triage-Briefed Split Agent	9,824	174	2,429	2,520	7,304

Table 15: Exact-record sensitivity to LLM-assisted evaluator stages. Pessimistic treatment marks direct semantic fields non-correct; credited views treat the indicated fields as correct. These are robustness bounds, not replacement headline scores.

Architecture	Original strict exact records	Direct semantic incorrect	Direct semantic credited	Alignment-dependent credited
Direct JSON	48/144 (33.3%)	34/144 (23.6%)	55/144 (38.2%)	77/144 (53.5%)
Update Tool	43/144 (29.9%)	32/144 (22.2%)	56/144 (38.9%)	66/144 (45.8%)
Diff-then-Verify	44/144 (30.6%)	32/144 (22.2%)	53/144 (36.8%)	63/144 (43.8%)
Split Agent	26/144 (18.1%)	20/144 (13.9%)	33/144 (22.9%)	40/144 (27.8%)
Triage-Briefed Split Agent	36/144 (25.0%)	28/144 (19.4%)	49/144 (34.0%)	58/144 (40.3%)

Table 16: Evaluator validation against adjudicated independent-reconstruction labels. Exact agreement distinguishes *correct*, *partial*, and *incorrect*; binary agreement collapses partial and incorrect into non-correct.

Check	Result
Validation inputs	60-item audit slice; 120 independent annotator submissions
Saved evaluator runs	9 runs; 3 mixed baseline/stability, 3 GPT-only, 3 Claude-only
Side-field checks	434 per run; 3,906 across all runs
Exact-label agreement	98.6–99.8% per run; 24 deviations across all runs, covering 12 distinct side-field cases
Binary agreement	99.1–100.0% per run; 15 deviations across all runs, covering 9 distinct side-field cases
Main residual pattern	Normalization and correct/partial/incorrect boundary cases in open text

contract/schema issues have been accounted for. Gold states come from authoring, independent reconstruction, and adjudication. The evaluator audit measures how closely the scoring machinery reproduces the adjudicated field judgments for system comparison and makes remaining scorer risk visible. Any future change to materialized gold, normalization, IU rubrics, or comparator policy requires rerunning the audit.

D.7 Diagnostic Metric Definitions

Diagnostic counts are post-evaluation bookkeeping over normalized record states. Let $\mathcal{S}$ be the set of evaluated commitment-core leaf slots, including scalar fields and aligned repeat-group slots of the form `group[i].field`. For each slot $s \in \mathcal{S}$, let p_s , g_s , and c_s denote the prior, gold post-turn, and can-

didate post-turn values. The evaluator applies the same normalization used for field scoring, including empty/unknown equivalence where specified. Let $x \equiv y$ denote normalized equality, and let $E(z)$ indicate normalized emptiness.

Changed(x, y) := $\neg(x \equiv y)$,

Filled(x, y) := $E(x) \wedge \neg E(y)$,

Emptied(x, y) := $\neg E(x) \wedge E(y)$,

Modified(x, y) := $\neg E(x) \wedge \neg E(y) \wedge \neg(x \equiv y)$.

These predicates operate on materialized record states. If a system emits an edit script or tool-call trace, the trace is first materialized into C , and the predicates are computed from the resulting record. The field-transition view is derived from the prior-to-gold relation: *set* is $E(p_s) \wedge \neg E(g_s)$; *clear* is $\neg E(p_s) \wedge E(g_s)$; *change* is $\neg E(p_s) \wedge \neg E(g_s) \wedge \neg(p_s \equiv g_s)$; and *preserve* is $p_s \equiv g_s$. A transition is counted correct when the candidate field receives a strict *correct* verdict. For repeated groups, s is defined after the scorer’s candidate-to-gold instance alignment. Missed gold rows are represented by gold-indexed slots and non-empty hallucinated rows are added as candidate-indexed slots, so repeat-instance failures can make per-run transition denominators differ slightly even though each transition label is still computed from prior and gold values in that aligned field view. Each definition below includes one local example. The examples use human-readable field names and show

only the fields needed to understand the diagnostic; all reported metrics are still computed over the full post-turn record.

Missed supported update. $\text{Changed}(p_s, g_s)$ and the candidate slot is not correct. This includes omissions and wrong values for slots that the gold state requires the system to update. *Example:* in a patient-medication update (FM-PA-10), the patient says salbutamol is a puffer and “I do two puffs when I get wheezy.” The gold salbutamol dose is 2 puffs; one candidate leaves it null.

Unsupported commit. $\text{Changed}(p_s, c_s)$ and the candidate slot is not correct. If the gold slot was preserved, this is also a collateral edit; if the gold slot changed, this records that the system committed an unsupported alternative value. *Example:* in a duplicate-participant grant update (RM-GR-R16), the user says a grant has “exactly two participants” and that P3 AquaSense Labs is not a separate participant. One candidate instead writes a WP2 staff-effort row as *P3 AQL*, not *P2 AQUA*.

Collateral edit. A field is a collateral edit when gold preserves it but the candidate changes it: $\neg\text{Changed}(p_s, g_s)$ and $\text{Changed}(p_s, c_s)$. This measures whether systems change slots that should have remained unchanged. *Example:* in a patent-disclosure distractor item (FM-PF-10), a casual aside says Jan tripped over a home-automation setup. The gold non-UCI inventor count remains empty, but one candidate sets it to 1.

State-history mismatch ignored. On items tagged with state-history mismatch pressure, the candidate preserves a prior-state value that the visible history and schema identify as a silent record error. *Example:* in a patent-date repair item (RM-PF-R17), the visible history says first successful operation was April 8, 2025, while the supplied prior record silently says August 4, 2025. A representative candidate adds the requested keywords but preserves August 4, 2025.

Failed correction. The item primary update family is correction, $\text{Modified}(p_s, g_s)$, and the candidate slot is not correct. This covers retained prior errors and incorrect replacement values. *Example:* in a crash-location correction item (FM-CR-03), the user says the crash was at the intersection of Main and 3rd, not the straight stretch. The gold location is the intersection; a representative candidate preserves the less specific prior location.

Failed retraction. The item primary update family is retraction, $\text{Emptied}(p_s, g_s)$, and c_s remains non-empty. This covers retaining the with-

drawn value or replacing it with a new concrete value. *Example:* in a grant security-retraction item (FM-GR-18), the user says AI triage advice should not be treated as a security issue or misuse scenario. The gold clears the security-issue note and sets the security booleans false; a representative candidate preserves the prior security concern.

Gate inconsistency. The candidate misses a controller value, opens or closes the wrong gated section, or produces child-field commitments inconsistent with the supported gate state. *Example:* in a patent agreement-gate item (FM-PF-22), the user says “We have a Material Transfer Agreement” with Genomic Tools, and that the MTA copy exists but is not attached. The gold opens the agreements gate and records one agreement row with type *Material Transfer Agreement*, party *Genomic Tools*, copy available *true*, and copy attached *false*; a representative candidate leaves the agreements gate *false* and emits no agreement row.

Repeat-group alignment error. An expected or predicted repeat-group instance is not matched by the evaluator’s alignment procedure. This includes missing gold instances, hallucinated candidate instances, and wrong instance routing that prevents a one-to-one match. *Example:* in a crash other-vehicle routing item (FM-CR-09), the user says “that other car” was a white Toyota Corolla, was changing lanes, and had front-right damage. The gold attaches all three facts to the other-vehicle row; one candidate splits them between two rows.

Changed-field precision, recall, and F_1 are computed from gold-changed slots and predicted-changed slots. Preservation error rate is the collateral edit count divided by the number of preserved slots. Unsupported-commit rate is the unsupported-commit count divided by the number of evaluated commitment-core leaf fields. Task-specific success indicators are derived for applicable items, including correction success, retraction success, history-recovery success, gate-execution success, repeat-group execution success, and input-incoherence probe success when applicable.

E Baseline Systems, Prompts, and Model Metadata

E.1 Architecture Interfaces

Direct JSON. Uses the full system prompt summarized in the prompt scaffold below. It sees the full schema guide and schema-completed prior state, then returns exactly one complete JSON ob-

ject for the post-turn record. Its final instruction is to audit the prior state against the schema, visible history, and current utterance and return the full resulting record using all and only evidence-supported values.

Update Tool. Uses the same shared policy and full item context, but replaces the Direct output interface with a sparse-edit interface: all committed changes must be expressed through `update_questionnaire_answers`. Omitted paths are not modified by that tool call, so clearing or repairing an unsupported prior value must be explicit. If no concrete change is evidence-supported, the model should make no tool call. The run materializes the tool-updated state into the full post-turn record before scoring.

Diff-then-Verify. Runs Update Tool as the first stage and materializes its complete candidate. A fresh GPT-5.4-mini verifier receives the schema, original prior record, candidate, visible history, and current utterance. It may make at most one sparse patch call through the same update tool. The patch is applied to the candidate, not the original prior record. Both stages share the generator settings (32k output tokens; no reasoning).

Split Agent. Uses one worker per cached questionnaire partition. Each worker receives the same visible conversation and the same shared policy, but only the schema slice and prior-state fields for its assigned top-level keys, plus a list of other top-level keys for lane awareness. Workers call the same update tool, but the partition wrapper rejects paths outside the worker’s assignment and rejects whole-repeat-group writes. Workers’ assigned fields are merged into a complete record before scoring.

Triage-Briefed Split Agent. Uses the same cached partitions plus reusable schema-level partition briefs. A high-recall triage prompt asks each partition to output only a JSON object with `needs_update`, `reason`, `likely_fields`, and optional `risk`, and to prefer true when uncertain. Triage agents do not write the record. A single global updater then sees the union of selected partition schemas, selected prior-state fields, the triage rationales, and the shared update-tool interface; other top-level fields are preserved

E.2 Tool Signature and Materialization

Tool-using architecture rows expose one tool:

```
update_questionnaire_answers(updates:
object) -> object
```

The updates object maps record paths to resulting values. Regular top-level scalar or table fields use their bare schema field id. Repeat-child fields use `repeat_group_id[index].field_id`. A repeat row is deleted with `repeat_group_id[index] = "__DELETE_INSTANCE__"`. Update Tool and the Triage-Briefed global updater may also set a repeat-group id to a sparse array of row-patch objects; omitted child fields in such patches are preserved. Split Agent partition workers must instead use explicit child paths or explicit row deletion. In all tool rows, omitted paths are preserved, and null is the explicit cleared value for scalar fields when the evidence supports clearing. Tool responses include status, `applied_update_count`, and, when applicable, rejected paths, shape errors, or materialized repeat-count changes.

E.3 Prompt Scaffold

Dialogue-construction scaffold. Expanded per-item interaction logs were not retained. The following is therefore a reconstruction from the variables used and the authors’ recollection, not a verbatim per-item prompt:

ConFormBench evaluates evidence-supported maintenance of structured form records from a schema, prior record, dialogue history, and current utterance. Create an {X}-turn conversation for scenario {Y}. Target the diagnostic condition {coverage-matrix row} and the intended field updates {fields and values or transitions}. Crucially, no other field may be plausibly updatable from the conversation, and there must be exactly one objectively justifiable set of updates.

The authors selected Claude Opus 4.7 or GPT-5.4 per item without a fixed assignment rule, based on drafting preference. The models supplied surface phrasing only; the authors retained all semantic and labeling decisions and manually revised every dialogue.

Direct JSON prompt. The Direct JSON rows use a fixed system prompt plus a generated item-specific human message. The system prompt is:

You are Direct, a single-call no-tool baseline for structured form updating. The public task context contains evidence and incumbent state. The schema field guide, visible conversation history, and current user utterance are the evidence used to support commitments. The prior record is the incumbent state to audit against that evidence. Your task is to audit the supplied prior record and return the post-turn record containing all and only commitments supported by the schema, visible history, and current utterance. Shared decision policy:

- Audit the prior state. Preserve a prior value only when it remains supported by the visible history, current utterance, and schema. If a prior value is unsupported, withdrawn, contradicted, or schema-inconsistent, clear or repair it as supported by the public packet.
- Treat visible history as evidence, not just background. If the prior record is missing, unsupported by, or contradicts a value determined by the schema, visible history, and current utterance, repair or clear it as supported.
- Commit only values supported by public evidence. Do not infer plausible but unstated values, do not complete partial addresses or names from world knowledge, and do not fabricate identifiers.
- Use null when the evidence clears, retracts, denies, withdraws, or fails to support a prior scalar value.
- Put each fact in the most specific schema field whose question and field instruction ask for that fact. Do not copy broad narrative details into child fields unless the child field itself asks for them.
- Never write <open> as a public value. Do not replace a supported prior value with an uncertain guess. If no value is supported for a scalar field, use null.
- Repeat groups are arrays of entity rows. Keep row identity stable: update an existing row when evidence identifies the same entity; create or delete rows only when evidence or a schema repeat count supported it.
- For repeat groups whose schema says mode=from_slot, update the count field when evidence supports a count. If a count is known but row attributes are not, leave child fields null rather than inventing labels.
- Table fields are arrays of row objects. Retain existing rows only when they remain supported by the evidence; clear, repair, add, or remove rows when the evidence and schema support that row-level change.

Output interface for Direct:

- Return exactly one complete JSON object representing the resulting record.
- Include every top-level regular field id and every repeat-group id shown in the prior record state. Do not include section/group/gate/branch ids.
- Repeat groups are null when absent/unknown, or arrays of row objects when rows are present. Every repeat row must include every child field id.
- Table fields are null when absent/unknown, or arrays of row objects when rows are present. Every table row must include every column id.
- Use null for unknown or empty scalar fields. Do not use placeholder strings.
- Output only the JSON object, with no commentary or explanation.

The human message is generated for each item with the following template:

```
=== SCHEMA FIELD GUIDE ===
<schema guide with field ids, answer types,
options, field instructions, repeat-group
modes, gate/branch navigation labels, table
columns, and semantic-IU summaries>
=== STATE SHAPE CONTRACT ===
<exact top-level state keys, repeat-row keys,
and table-row column keys>
=== PRIOR RECORD STATE ===
<schema-completed prior state JSON>
=== CONVERSATION HISTORY ===
<visible prior turns, or (none)>
=== CURRENT USER UTTERANCE ===
<current user utterance>
=== TASK ===
Audit the prior state against the schema, visible history,
and current utterance. Return the full resulting record
JSON using all and only evidence-supported post-turn
values.
```

Shared prompt scaffold. The original four architecture rows are prompt-stable by construction: they use the same shared decision policy printed in the Direct JSON system prompt above and the same generated item-context scaffold. The only

intended differences are the role line, output interface, and, for partitioned rows, the schema/prior projection and routing metadata. In compact form:

```
SYSTEM = role line + shared decision policy
+ output interface
HUMAN = optional field assignment + schema
field guide + state shape contract + prior
state projection + visible history + current
utterance + final instruction
```

Diff-then-Verify prompt. The verifier uses the shared decision policy but treats the materialized candidate as its working state. It audits every commitment and may emit at most one patch. The public artifact release contains the prompt templates and source prompt builders, selected partitions, tool calls, candidate records, and evaluator traces for the released runs. The archived generation runs store raw model responses and per-call metadata. Full per-turn generation input-message transcripts require rerunning the solver with message tracing enabled. This appendix summarizes the interfaces and points to the released item-specific field guides.

E.4 Model Snapshots and Run Settings

All five architecture rows use the same requested generator, gpt-5.4-mini, with 32k maximum output tokens, a 300s timeout, temperature null, and reasoning disabled. This includes both the Update Tool stage and the verifier stage of Diff-then-Verify. Per-call model_config traces record the actual 32k generation limit; the 16k cost_tracking_max_tokens value retained in some comparison manifests is an accounting ceiling used for pre-run cost guards. Table 18 records the generator snapshots and settings used for the Direct JSON architecture row and the model-strength ablation rows in Table 19. We report the provider-documented public snapshot or pinned model ID, the ID returned in our run traces, and the serving fingerprint when the provider exposed one. This matches common EMNLP reporting practice for LLM experiments: name the exact model identifier or snapshot where available, the decoding/runtime settings, and the prompt or interface template. Values are taken from provider documentation checked on 2026-05-25, the released run_metadata.json files, and the per-turn model_config / response metadata traces. All rows in Table 18 use the same solver, conformbench.systems.direct:solve, on the held-out conformbench_v1_20_80_seed20260524 split.

Compute and cost accounting. All reported systems use the listed hosted models at inference time; the study performs no training or fine-tuning. Table 17 reports the call, token, and USD cost accounting captured in the released `cost_report.json` files. Provider-reported cost metadata is available for the original four GPT-5.4-mini rows. Diff-then-Verify has complete call and token counts and uses the bundled pricing-helper estimate because provider cost metadata is absent. The saved Anthropic and Groq ablation traces contain call and token counts. We report their USD costs as unavailable because the traces lack provider cost fields and the released accounting avoids external price-sheet estimates. The NOUPDATE lower bound makes no LLM calls. Exact GPU hours, accelerator types, and parameter counts for closed hosted models are unavailable to the authors. For hosted open-model rows, identifiers expose nominal scale where applicable, such as GPT-OSS-120B and Qwen3-32B; providers control the serving infrastructure. All headline architecture and model-strength rows use the same evaluator implementation and requested evaluator: `gpt-5.4` (or `openai:gpt-5.4` in the ablation run metadata), public OpenAI snapshot `gpt-5.4-2026-03-05`, response model `openai/gpt-5.4`, max tokens 32k for semantic/comparator calls, timeout 300s, temperature null, and no exposed serving fingerprint. The original four architecture runs and the model-strength ablations use the configured medium reasoning setting. The Diff-then-Verify metadata retains that historical configuration label, but its official OpenAI helper removes `reasoning_effort` before transmission; all saved Diff-then-Verify comparator traces report zero reasoning tokens. The frozen run preserves this effective execution path, and Appendix D reports evaluator-route sensitivity bounds. The development-only model-family bias check in Section 5 re-scores fixed GPT-5.4-mini Direct JSON candidate records with `anthropic/claude-sonnet-4-6`, the pinned medium-reasoning Claude 4.6 model.

E.5 Partition Rules

Partition splits are created once per questionnaire from schema information only, validated deterministically, cached, and reused across development and test items. The splitter partitions top-level state keys: bare regular field ids and repeat-group ids, not repeat-child ids. A valid split must

be complete (every top-level key appears exactly once), dependency-safe (gate controllers stay with gated children, branch controllers with routed children, and repeat-count controllers with their repeat groups), semantically coherent, and roughly balanced. The splitter prompt asks for 3–8 partitions with 3–15 top-level keys where possible and asks it to merge singleton partitions into the closest semantic neighbor. At runtime, Split Agent merges only each worker’s assigned top-level keys, while Triage-Briefed Split preserves all unselected partitions and allows the global updater to write only selected top-level keys.

F Additional Results, Bootstrap Intervals, and Load-Normalized Analysis

F.1 Field-Transition Accuracy

Table 20 breaks field accuracy down by the prior-to-gold transition. Preserve decisions are easiest, while clear transitions are the least reliable transition type across all five systems. The denominators vary slightly across rows because repeated instances are aligned separately for each candidate record; Candidate-specific repeat alignment explains the released-run differences.

F.2 Strict Versus Lenient Scoring

Table 21 checks whether strict scoring alone is driving the main conclusion. Treating partial verdicts as correct improves changed-field F_1 and exact-record counts, but complete-record success still remains far below field-level accuracy.

F.3 Architecture-Family Solvability

Table 22 asks whether any of the five systems solves each held-out item. Solvability falls sharply with history, repeats, high load, mixed revisions, clearing, or silent-state repair.

F.4 Paired Bootstrap Intervals

We compute paired item-level bootstrap confidence intervals for the held-out architecture comparison. Each bootstrap replicate samples the 144 held-out items with replacement and recomputes the metric on the sampled item set. For changed-field F_1 , all changed-field counts within a sampled item are kept together before recomputing strict micro F_1 . The intervals below use 10,000 bootstrap samples with seed 20260710. The intervals do not reliably separate Direct JSON, Update Tool, Diff-then-Verify, and Triage-Briefed Split Agent on exact-

Table 17: Observed inference accounting for released held-out runs. Original-run costs are provider reported; Diff-then-Verify uses a bundled pricing estimate.

Run family	Hosted provider route	Calls	Tokens	USD cost
Direct JSON architecture row	OpenAI-compatible generation and evaluation	565	3.35M	\$6.02
Update Tool architecture row	OpenAI-compatible generation and evaluation	644	3.97M	\$4.86
Diff-then-Verify architecture row	OpenAI-compatible generation and evaluation	784	5.39M	\$4.18 estimated
Split Agent architecture row	OpenAI-compatible generation and evaluation	1,511	5.69M	\$6.59
Triage-Briefed Split architecture row	OpenAI-compatible generation and evaluation	1,450	6.70M	\$7.47
Five-system architecture subtotal	OpenAI-compatible generation and evaluation	4,954	25.11M	\$29.12 (\$24.94 reported + \$4.18 estimated)
Claude Opus 4.7 Direct JSON ablation	Anthropic generation; OpenAI-compatible evaluation	526	3.55M	unavailable; 526 calls missing cost metadata
GPT-OSS-120B Direct JSON ablation	Groq generation; OpenAI-compatible evaluation	513	3.05M	unavailable; 513 calls missing cost metadata
Qwen3-32B Direct JSON ablation	Groq generation; OpenAI-compatible evaluation	519	3.18M	unavailable; 519 calls missing cost metadata

Table 18: Generator metadata for the Direct JSON comparison. Released directories include run IDs, candidates, prompt builders, evaluator traces, and response metadata.

Setting	GPT-5.4-mini	Claude Opus 4.7	GPT-OSS-120B	Qwen3-32B
Requested generator	gpt-5.4-mini	anthropic:claude-opus-4-7	openai/gpt-oss-120b	qwen/qwen3-32b
Public snapshot / pinned ID	gpt-5.4-mini-2026-03-17	claude-opus-4-7	openai/gpt-oss-120b	qwen/qwen3-32b
Parameter count	not disclosed by provider	not disclosed by provider	120B-class from model ID	32B-class from model ID
Trace-resolved ID	gpt-5.4-mini	claude-opus-4-7	openai/gpt-oss-120b	qwen/qwen3-32b
Response model name	openai/gpt-5.4-mini	claude-opus-4-7	openai/gpt-oss-120b	qwen/qwen3-32b
Response provider / route	OpenAI-compatible	Anthropic	Groq OpenAI-compatible	Groq OpenAI-compatible
Reasoning setting	none	medium	medium	default
Max tokens / timeout	32k / 300s	32k / 300s	32k / 300s	32k / 300s
Temperature	null	null	null	null
Serving fingerprint(s)	not exposed	not exposed	fp_87b3c396db	fp_5cf921caa2
Run batch size	20	8	8	8
Run window	2026-05-25 14:48–14:52 +02:00	2026-05-25 15:16–16:12 UTC	2026-05-25 15:25–15:43 UTC	2026-05-25 15:43–16:05 UTC

record correctness. Diff-then-Verify is reliably ahead of Triage-Briefed Split Agent on changed-field F_1 and collateral edits, but not on exact-record correctness, and is reliably ahead of Split Agent on all three metrics. Direct JSON is reliably ahead of Split Agent on exact-record correctness and collateral edits, and ahead of Triage-Briefed Split Agent on exact-record correctness. Update Tool is reliably ahead of Split Agent on all three metrics. Triage-Briefed Split Agent is reliably ahead of Split Agent on changed-field F_1 and collateral edits.

F.5 Load-Normalized Exact-Record Analysis

To assess the role of record size in exact-record failure, we compute two leave-one-item-out independent-field nulls. For item i and architecture a , let n_{ia} be the number of evaluated field checks, g_{ia} the gold-changed field checks, and p_{ia} the gold-preserved field checks in that architecture’s evaluator view. The field-count null is

$$q_{ia}^{\text{field}} = (\hat{p}_{a,-i})^{n_{ia}},$$

where $\hat{p}_{a,-i}$ is strict all-field accuracy estimated without item i . The changed/preserve null is

$$q_{ia}^{\text{cp}} = (\hat{p}_{a,-i}^{\text{chg}})^{g_{ia}} (\hat{p}_{a,-i}^{\text{pres}})^{p_{ia}}.$$

For architecture-family solvability, we report $1 - \prod_a (1 - q_{ia})$. This descriptive independence approximation provides a load baseline for the observed architecture-family solvability. Confidence intervals use 10,000 item-bootstrap samples within each load bucket with seed 20260525.

G Released Artifacts and CONFORMBENCH STUDIO

The released artifact bundle contains the public item packets, machine-readable schemas, prior records, gold resulting records, scorer, baseline candidate records, baseline prompt templates and source prompt builders, tool traces where applicable, evaluator decision traces, model/run metadata, pseudonymized annotation summaries, adjudication records, and the released audit-slice manifest, together with scripts for reproducing the benchmark statistics, baseline-system results, and robustness analyses reported in the paper. The benchmark data, schemas, documentation, and audit artifacts are released under the Creative Commons Attribution 4.0 International license (CC BY 4.0); the scorer, baseline runners, scripts, and CONFORMBENCH STUDIO code are released under the MIT

Table 19: Direct JSON model-strength ablation on the held-out 144-item test split. All rows use the same Direct JSON prompt template and GPT-5.4 evaluator with medium reasoning. GPT-OSS-120B and Qwen3-32B are served through Groq; Qwen uses the provider-default reasoning setting.

Generator	Reasoning	Fully correct turns	All-field accuracy	Changed P/R/ F_1 (%)	Preservation	Collateral edits
GPT-5.4-mini	none	48/144 (33.3%)	96.2%	84.5/80.2/82.3	98.6%	120
Claude Opus 4.7	medium	53/144 (36.8%)	97.9%	85.4/91.1/88.2	98.4%	141
GPT-OSS-120B	medium	51/144 (35.4%)	97.4%	88.0/85.1/86.5	99.1%	77
Qwen3-32B	default	29/144 (20.1%)	94.2%	74.1/75.4/74.8	97.4%	226

Table 20: Field-level transition accuracy on the held-out 144-item test split. Transitions are derived from the prior-to-gold relation within each run’s aligned leaf-field view.

Architecture	Set	Change	Clear	Preserve
Direct JSON	647/828 (78.1%)	299/343 (87.2%)	39/58 (67.2%)	8471/8597 (98.5%)
Update Tool	688/823 (83.6%)	296/348 (85.1%)	34/56 (60.7%)	8480/8597 (98.6%)
Diff-then-Verify	708/823 (86.0%)	294/348 (84.5%)	38/56 (67.9%)	8490/8593 (98.8%)
Split Agent	670/823 (81.4%)	281/348 (80.7%)	38/56 (67.9%)	8367/8607 (97.2%)
Triage-Briefed Split Agent	686/817 (84.0%)	300/352 (85.2%)	32/59 (54.2%)	8459/8596 (98.4%)

Table 21: Effect of partial verdicts on held-out system scores. Strict counts treat *partial* as non-correct; lenient counts treat *partial* as correct. The first five rows are the architecture comparison, and the last three rows are the Direct JSON model-strength ablation.

System row	Partial fields all / changed	Exact records strict / lenient	Changed F_1 (%) strict / lenient
Direct JSON	35 / 35	48 / 52	82.3 / 85.2
Update Tool	43 / 40	43 / 53	85.1 / 88.4
Diff-then-Verify	40 / 37	44 / 51	86.0 / 89.1
Split Agent	65 / 58	26 / 33	78.5 / 83.1
Triage-Briefed Split Agent	47 / 42	36 / 49	83.8 / 87.2
Claude Opus 4.7	34 / 30	53 / 66	88.2 / 90.6
GPT-OSS-120B	47 / 46	51 / 63	86.5 / 90.3
Qwen3-32B	65 / 62	29 / 40	74.8 / 79.7

Table 22: Architecture-family solvability by item shape. An item is solved if any GPT-5.4-mini architecture yields an exact record.

Shape slice	Items	Solved by ≥ 1 arch.
No history required	77	49/77 (63.6%)
History required	67	20/67 (29.9%)
No repeated group	46	34/46 (73.9%)
Single repeated group	55	24/55 (43.6%)
Multiple repeated groups	43	11/43 (25.6%)
Low load	40	30/40 (75.0%)
High or scale load	36	6/36 (16.7%)
Mixed revision operation	62	20/62 (32.3%)
Clear transition present	33	12/33 (36.4%)
S4 silent mismatch	30	11/30 (36.7%)

License. The public forms and domain standards listed in Appendix A.1 served as schema and provenance references for synthetic benchmark construction; the release excludes the source documents. CONFORMBENCH STUDIO is an inspection interface for the same artifacts. It lets users inspect public packets, schema structure, prior/gold/candidate records, field diffs, evidence spans, repeat alignment, scoring verdicts, comparator traces, and aggregate reports. The versioned public artifact release accompanies the published paper.

Table 23: Per-system bootstrap estimates on the held-out architecture comparison. Intervals are 95% item-level bootstrap confidence intervals.

Architecture	Exact record rate	Changed-field F_1	Collateral edits
Direct JSON	33.3% [25.7%, 41.0%]	82.3% [74.8%, 87.6%]	120 [85, 157]
Update Tool	29.9% [22.2%, 37.5%]	85.1% [82.1%, 87.7%]	90 [59, 125]
Split Agent	18.1% [11.8%, 24.3%]	78.5% [74.9%, 81.8%]	217 [169, 272]
Triage-Briefed Split Agent	25.0% [18.1%, 31.9%]	83.8% [79.9%, 87.1%]	120 [89, 154]
Diff-then-Verify	30.6% [23.6%, 38.2%]	86.0% [82.4%, 89.0%]	93 [65, 126]

Table 24: Paired bootstrap differences for architecture comparisons on the held-out test split. Differences are row architecture A minus architecture B. Intervals that include zero do not support ranking the two systems on that metric.

Comparison	Exact record rate	Changed-field F_1	Collateral edits
Direct JSON – Update Tool	+3.5 pp [-3.5, +10.4]	-2.8 pp [-11.0, +2.9]	+30 [-7, +68]
Direct JSON – Split Agent	+15.3 pp [+8.3, +22.2]	+3.7 pp [-6.1, +10.4]	-97 [-140, -56]
Direct JSON – Triage-Briefed Split Agent	+8.3 pp [+1.4, +15.3]	-1.5 pp [-10.6, +4.5]	+0 [-33, +34]
Direct JSON – Diff-then-Verify	+2.8 pp [-4.2, +9.0]	-3.7 pp [-12.5, +2.2]	+27 [-3, +58]
Update Tool – Split Agent	+11.8 pp [+4.9, +18.8]	+6.6 pp [+3.5, +9.8]	-127 [-177, -82]
Update Tool – Triage-Briefed Split Agent	+4.9 pp [-2.1, +11.8]	+1.3 pp [-1.2, +4.0]	-30 [-61, +1]
Update Tool – Diff-then-Verify	-0.7 pp [-6.9, +5.6]	-0.9 pp [-3.2, +1.5]	-3 [-28, +26]
Split Agent – Triage-Briefed Split Agent	-6.9 pp [-14.6, +0.7]	-5.3 pp [-7.8, -2.8]	+97 [+52, +144]
Split Agent – Diff-then-Verify	-12.5 pp [-19.4, -5.6]	-7.5 pp [-9.9, -5.1]	+124 [+80, +171]
Triage-Briefed Split Agent – Diff-then-Verify	-5.6 pp [-11.8, +0.7]	-2.2 pp [-4.0, -0.5]	+27 [+4, +51]

Table 25: Load-normalized architecture-family solvability with 95% item-bootstrap intervals. Rate columns are percentages; expected rates use the descriptive family approximation above.

Load	Items	Mean fields	Mean changed	Observed solved	Field-count null	Changed/preserve null	Errors/100 fields
Low	40	67.9	2.4	75.0 [60.0, 87.5]	48.2 [37.6, 58.8]	74.9 [68.1, 81.3]	1.16 [0.80, 1.61]
Medium	68	58.0	6.1	48.5 [36.8, 60.3]	56.0 [49.0, 62.9]	58.3 [53.3, 63.3]	3.89 [3.07, 4.85]
High	23	76.3	14.6	13.0 [0.0, 26.1]	45.0 [31.0, 59.0]	22.6 [15.1, 30.4]	6.02 [4.20, 8.32]
Scale	13	108.7	29.3	23.1 [0.0, 46.2]	43.2 [22.5, 63.1]	8.7 [2.4, 18.0]	5.08 [3.12, 7.95]
High+scale	36	88.0	19.9	16.7 [5.6, 30.6]	44.4 [32.9, 55.9]	17.5 [11.8, 23.8]	5.60 [4.20, 7.37]