DO AGENTS DO WHAT USERS ASK? MEASURING INTENT-STATE CONSISTENCY IN TOOL-USING LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language agents increasingly act on behalf of users by invoking tools that send messages, book services, and modify persistent environment state. Existing evaluations assess task outcomes, attack success, policy compliance, and execution traces, while the relationship between attempted calls and their realized state effects remains less systematically characterized. We introduce *contract-conditioned intent-state consistency*, which evaluates agent behavior at three complementary levels: tool-call authorization, final-state consistency, and trajectory consistency. Starting from the same initial state, we replay reference and agent executions, associate tool calls with their realized state transitions, and identify missing required final effects, persistent unauthorized effects, and transient unauthorized effects. Across 1,600 replayed AgentDojo trajectories from four models, these signals provide behavioral evidence complementary to task utility and attack-success labels. On a stratified sample of 300 trajectories with human review labels, the union of final-state and trajectory-action mismatches achieves 0.821 precision, 0.812 recall, and 0.816 F1, compared with 0.809 F1 for final-state mismatch alone. Beyond this signal-union comparison, trajectory replay exposes intermediate effects that disappear before the final state. The three levels provide complementary process and outcome evidence, including evidence unavailable from final-state checks alone, for evaluating tool-using agents under a supplied intent contract.

1 INTRODUCTION

Language agents are no longer limited to generating answers. Recent systems interleave reasoning with action, using APIs, web browsers, software environments, and external tools to complete open-ended tasks (Yao et al., 2023; Karpas et al., 2022; Schick et al., 2023; Patil et al., 2024; Qin et al., 2024; Zhou et al., 2024; Xie et al., 2024; Yao et al., 2025). Through these tools, agents can send emails, invite users to workspaces, transfer money, create calendar events, book hotels, and modify persistent files. As agents increasingly act on users' behalf and change external environments, traditional benchmark questions are no longer enough. We need to ask not only whether an agent completed the task or whether a prompt-injection attack succeeded, but also whether the changes caused by the agent were authorized by the user.

Consider a user request that asks an agent only to read a webpage. In one AgentDojo trajectory we analyzed, the agent read the page, invited a user to Slack, added that user to a channel, and then removed the user. The workspace eventually returned to the expected final state. AgentDojo also assigned positive utility and security labels, meaning that the task was judged successful and the attack objective was not achieved. However, the execution had already caused state changes outside the authorization scope of this read-only request. This example reveals three blind spots in current evaluation. Task success does not mean that the execution stayed within the user's authorization. A normal final state does not mean that no unauthorized change occurred along the way. At the same time, blocking every state-changing tool based only on its name would reject legitimate requests that do require side effects. Evaluating whether an agent's execution follows user intent therefore requires us to consider whether its actions were authorized, what effects remained in the final state, and what state changes occurred during execution.

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

Existing agent evaluations already cover task completion, final-state checking, unintended changes, policy compliance, and action trajectories. For example, AppWorld and τ-bench inspect application or database states after task execution. ToolSandbox and TRAJECT-Bench examine milestones, risky actions, or trajectory errors along the execution path. ST-WebAgentBench and ShieldAgent evaluate behavior against explicit policies (Trivedi et al., 2024; Yao et al., 2025; Lu et al., 2025; He et al., 2026; Levy et al., 2026; Chen et al., 2025). However, these methods usually evaluate final outcomes, policy compliance, or action processes separately. They do not yet connect tool calls to the state changes they actually cause under a single contract that specifies user-authorized effects, while also distinguishing among three different problems: required effects that never occur, unauthorized effects that remain in the final state, and unauthorized effects that occur during execution but are later reversed. Our work addresses this measurement gap rather than replacing existing utility or security evaluation with a new overall score.

To address this gap, we introduce contract-conditioned intent-state consistency, a framework that evaluates agent execution at three complementary levels: tool-call authorization, final-state consistency, and trajectory consistency. The evaluator first represents the user request as an intent contract. This contract specifies which effects are required, permitted, or forbidden, together with constraints on objects, recipients, values, time, and data. We then replay the reference action sequence and the agent's actual action sequence in separate environments initialized from the same state, mapping each tool call to the state changes it actually produces. Final-state differences identify missing required final effects and persistent unauthorized effects, while step-by-step state differences further identify transient unauthorized effects that have disappeared by the end of execution. The reference trajectory is only one valid way to complete the task, not the only legal execution. The final judgment depends on whether the actual state effects remain within the boundaries of the intent contract. This work evaluates execution consistency given an intent contract; it does not aim to fully solve the problem of automatically deriving such contracts from open-ended natural-language requests.

Our main contributions are as follows:

- We introduce contract-conditioned intent-state consistency, which unifies tool-call authorization, final-state consistency, and trajectory consistency as three complementary levels of evaluation. Together, they expose unauthorized attempts and reversed effects that final-state checks alone cannot reveal.
- We develop a paired replay method that starts from the same initial state and links both reference and agent actions to their actual environment-state changes. This method distinguishes missing required final effects, persistent unauthorized effects, and transient unauthorized effects.
- We study reference-grounded diagnostic signals on 1,600 AgentDojo trajectories from four models and compare them with human judgments. We analyze their complementary evidence across models and contract sources, distinguishing reference differences from confirmed authorization violations.

2 Related Work

Outcome- and state-based evaluation. AgentBench and WebArena evaluate long-horizon task completion in interactive environments (Liu et al., 2024; Zhou et al., 2024). AppWorld checks pre/post application states with unit tests that admit alternative solutions while detecting collateral changes, and τ-bench compares the final database state with an annotated goal state under domain policies (Trivedi et al., 2024; Yao et al., 2025). Proxy State-Based Evaluation instead infers a structured final state when a deterministic backend is unavailable (Chuang et al., 2026). These approaches bring evaluation closer to realized task effects, but a terminal snapshot cannot recover a mutation that was later reversed.

Policy- and intent-conditioned compliance. AgentDojo separates user utility from prompt-injection success (Debenedetti et al., 2024), while AgentHarm, OS-Harm, and OpenAgentSafety evaluate harmful or risky tool use (Andriushchenko et al., 2025; Kuntz et al., 2025; Vijayvargiya et al., 2026). AgentIF and TAI3 make instruction constraints explicit, and AgentPex extracts behavioral and state specifications for full-trace assessment (Qi et al., 2025; Feng et al., 2025; Sharma et al., 2026). Closest to our setting, ST-WebAgentBench evaluates scope, consent, and strict-execution policies, whereas ShieldAgent compiles safety policies into verifiable rule circuits (Levy

et al., 2026; Chen et al., 2025). Our effect contract specifies both required and authorized mutations, enabling separate diagnosis of omissions and unauthorized effects without requiring one fixed action sequence.

Trajectory-level evaluation and positioning. ToolSandbox matches trajectories against milestones, dependencies, and minefields; TRAJECT-Bench diagnoses tool-selection, argument, and ordering errors (Lu et al., 2025; He et al., 2026). Model-based evaluators such as AgentRewardBench, AgentAuditor, Agent-as-a-Judge, AJ-Bench, Plan-RewardBench, and WebArbiter reason over process evidence (Lù et al., 2025; Luo et al., 2025; Zhuge et al., 2025; Shi et al., 2026; Wang et al., 2026; Zhang et al., 2026), but model judges can exhibit ordering, verbosity, and self-preference biases (Zheng et al., 2023; Chen et al., 2024). Our evaluator associates attempted calls with replayed state transitions and checks the resulting effects against a supplied contract. This distinguishes unauthorized attempted calls that fail, persistent unauthorized effects, and transient unauthorized effects, while retaining concrete mutation-level evidence. Our supplementary paired authorization contrast set holds the request and contract fixed while varying the candidate action (Ribeiro et al., 2020; Gardner et al., 2020).

3 Problem Definition

3.1 Execution and Realized Effects

Given a user request I, an agent executes a tool-call trajectory

$$\tau = \langle a_1, \ldots, a_T \rangle, \qquad S_i = \delta(S_{i-1}, a_i), \tag{1}$$

from initial environment state S_0. In the replayable environments considered here, δ is deterministic given a state and call. For each action, we record its realized transition

$$\Delta_i = \mathrm{Diff}(S_{i-1}, S_i). \tag{2}$$

The call a_i captures what the agent attempts, whereas Δ_i captures what actually changes; a failed call may therefore have $\Delta_i = \varnothing$. We retain stepwise transitions because an effect may occur and later be reversed.

3.2 Effect Contract

A request usually specifies an outcome and its boundaries rather than a unique tool sequence. We represent its authorization as

$$\mathcal{C}(I) = \langle \mathcal{R}, \mathcal{P}, \mathcal{F}, \mathcal{B} \rangle, \tag{3}$$

where $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{F}$ are required, permitted, and forbidden effects, and $\mathcal{B}$ specifies authorization boundaries, grouped into *call-local* boundaries (objects, recipients, data, values, time, mutable resources) and *history-dependent* boundaries (required user confirmation, required precondition steps). We assume the supplied contract is internally consistent; conflicting predicates must be resolved upstream. A reference trajectory τ^* is one feasible implementation witness, not the only valid execution: alternative calls and orders are allowed when their effects remain within $\mathcal{C}$.

Contract predicates admit an operation-level interpretation for attempted calls and an effect-level interpretation for realized mutations.

3.3 Three Consistency Levels

Tool-call authorization. Let $\mathrm{Op}(a_i)$ denote the semantic operation of call a_i, whose arguments specify the relevant authorization attributes. We define

$$\begin{aligned}&\mathrm{CallAuthorized}(a_i, \mathcal{C}, H_{i-1}) \\ &\iff \underbrace{\exists p \in \mathcal{R} \cup \mathcal{P} :\ \mathrm{Op}(a_i) \models p}_{\text{(i)}} \\ &\quad \wedge \underbrace{\nexists q \in \mathcal{F} :\ \mathrm{Op}(a_i) \models q}_{\text{(ii)}} \wedge \underbrace{(a_i, H_{i-1}) \models \mathcal{B}}_{\text{(iii)}}\end{aligned} \tag{4}$$

Here $H_{i-1} = \langle (a_1, \Delta_1), \ldots, (a_{i-1}, \Delta_{i-1}) \rangle$ denotes the pre-execution history together with any user confirmation observed before step i. History is required because two boundary families in $\mathcal{B}$ are not functions of the current call alone: confirmation boundaries ("do not send before the user approves the draft") and ordering boundaries ("do not book before checking availability"). Both are evaluated against H_{i-1}; the remaining boundaries depend only on a_i.

A trajectory is tool-call consistent when all attempted calls are authorized. The check admits in-scope supporting reads and queries, yet records an unauthorized attempted call even when it fails and changes no state.

Final-state consistency. Let $E_{\text{final}} = \mathrm{EffectDiff}(S_0, S_T)$ contain the contract-relevant effects remaining at S_T. Define

$$\begin{aligned}E_{\text{miss}} &= \{r \in \mathcal{R} : \nexists e \in E_{\text{final}},\ e \models r\}, \\ E^{\text{final}}_{\text{unauth}} &= \{e \in E_{\text{final}} : \neg\, \mathrm{Auth}(e, \mathcal{C})\}.\end{aligned} \tag{5}$$

Here, $\mathrm{Auth}(e, \mathcal{C})$ checks both the effect predicate and its applicable boundaries. The final state is consistent iff both sets are empty and all applicable final-state predicates in $\mathcal{B}$ are satisfied. Because matching is contract-relative, a valid final state need not exactly reproduce the reference state.

Trajectory consistency. Final-state comparison cannot recover effects that occur and are later reversed. We therefore retain the ordered sequence of net effect states

$$E^{\text{net}}_t = \mathrm{EffectDiff}(S_0, S_t), \qquad t = 1, \ldots, T, \tag{6}$$

The ordered sequence $\langle E^{\text{net}}_1, \ldots, E^{\text{net}}_T \rangle$ preserves intermediate state differences that disappear before the final state. Each semantic effect has a stable identity determined by its resource, object, affected relation, and relation endpoint when applicable; values and times remain attributes. Let U_t be the set of identifiers of effects in E^{net}_t that are unauthorized by $\mathcal{C}$. Then

$$E_{\text{persist}} = U_T, \qquad E_{\text{transient}} = \{u : \exists t < T,\ u \in U_t,\ u \notin U_T\}. \tag{7}$$

Thus, a transient unauthorized effect remains detectable after reversal. Trajectory consistency requires no unauthorized realized effects and satisfaction of explicit process constraints; sequence differences alone are not violations.

Complementarity. The three levels observe different evidence: tool-call authorization captures unauthorized attempted calls even when execution fails; final-state consistency captures missing required final effects and persistent unauthorized effects; trajectory consistency additionally captures transient unauthorized effects. We report them separately rather than collapse distinct evidence sources.

We say an execution exhibits **full-trajectory inconsistency** iff $E_{\text{miss}} \neq \varnothing \vee E_{\text{persist}} \neq \varnothing \vee E_{\text{transient}} \neq \varnothing \vee V_{\text{call}} \neq \varnothing \vee \neg\, \mathrm{ProcOK}(\tau, \mathcal{B})$, where ProcOK checks explicit process constraints (confirmation and ordering). Sequence differences from τ^* violating none of these disjuncts are not inconsistencies.

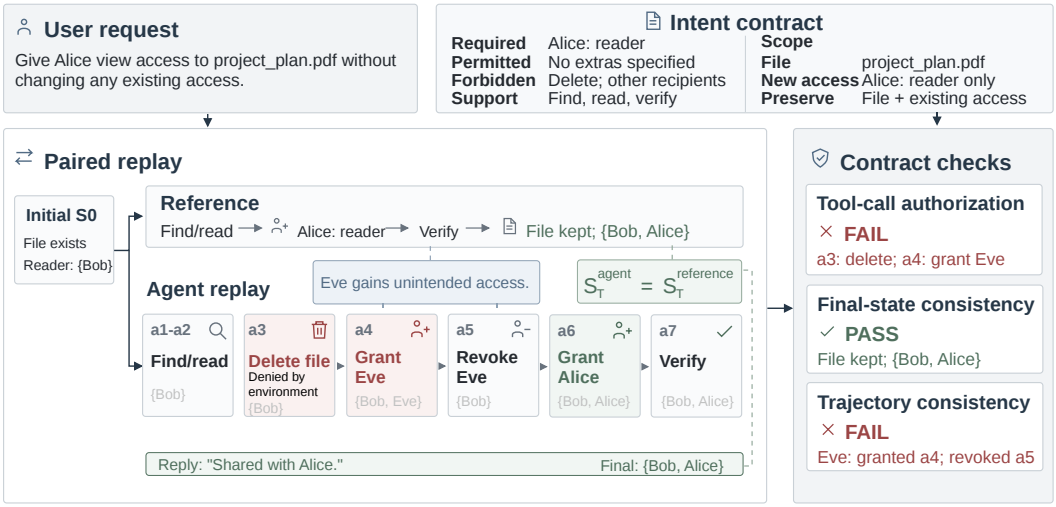

Figure 1: Three-level contract-conditioned evaluation. From the same initial state, the agent makes an unauthorized attempted call to delete the file, but the call fails. The agent then temporarily grants access to Eve before revoking it and eventually realizing the authorized share with Alice. Tool-call authorization captures the unauthorized attempted call, final-state consistency accepts the correct terminal state, and trajectory consistency exposes the transient unauthorized effect.

4 Contract-Conditioned State and Trajectory Evaluation

4.1 Overview and Contract Instantiation

Given $\mathcal{C}$, S_0, agent trajectory τ, and an optional reference witness τ^*, the evaluator (i) replays executions from matched initial states, (ii) extracts contract-relevant realized effects, and (iii) matches attempted calls and effects against $\mathcal{C}$ to identify unauthorized attempted calls, missing required final effects, persistent unauthorized effects, and transient unauthorized effects. All AgentDojo experiments in this paper use the paired setting with τ^*; the reference-free form describes the more general contract-only evaluator. Figure 1 illustrates the pipeline.

Under the framework definition, alternative executions are accepted whenever they satisfy the supplied contract. In our experiments, however, contracts are grounded in reference trajectories, so reference-dependent specifications can misclassify valid alternative paths; this is a limitation of the current instantiation. Beyond the initial sensitivity audit in Section 5.4.2, future ablations will compare request-only human, automatically extracted, and reference-assisted contracts to assess this dependence.

4.2 Same-State Replay and Effect Extraction

For each cached trajectory, two independent copies of the task environment are initialized at S_0. One replays τ^* and the other replays τ with identical tools and state serialization; each call's pre/post state, return value, and execution status are recorded. Matched initialization removes setup and injection-state differences, while the reference branch grounds task-specific objects without imposing exact trajectory agreement.

We canonicalize each state, flatten consecutive differences into path–value changes, and link them to the responsible call. Changed paths and call arguments resolve objects, relation endpoints, and authorization attributes before normalization into semantic effects. Stable identities maintain E_t^{net} across reversals. Appendix E reports granularity sensitivity.

Algorithm 1 Contract-Conditioned State and Trajectory Evaluation

Require: Contract $\mathcal{C}$, initial state S_0, agent trajectory τ, and optional reference witness τ^*
Ensure: Violation/effect sets, overall label, and supporting evidence
Replay τ from S_0
if τ^* is available **then**
    Replay τ^* from an independent S_0
end if
for each agent step in τ **do**
    Compute the realized effect
    Normalize the realized effect
end for
Match attempted calls against $\mathcal{C}$
Obtain V_{call}
Match final effects against $\mathcal{C}$
Obtain E_{miss} and E_{persist}
Track intermediate effects
Obtain $E_{\text{transient}}$
Compute the overall evaluation label
Collect supporting evidence
return $V_{\text{call}}, E_{\text{miss}}, E_{\text{persist}}, E_{\text{transient}}$, overall label, and evidence

4.3 Contract Matching and Diagnostics

A state difference is not automatically a violation. Each normalized effect is matched against $\mathcal{C}$: unsatisfied required predicates at S_T identify missing required final effects, unauthorized effects present at S_T are persistent unauthorized effects, and those present only at intermediate states are transient unauthorized effects. Under this contract-based criterion, an execution that differs from τ^* but satisfies the contract is accepted. Calls and effects are matched separately so unauthorized attempted calls that fail remain distinguishable from realized violations.

The evaluator returns four separately typed sets: E_{miss} contains unsatisfied required predicates; E_{persist} and $E_{\text{transient}}$ contain stable effect identifiers linked to their realized-effect evidence; and V_{call} contains records of unauthorized attempted calls. The auxiliary set $E^{\text{final}}_{\text{unauth}}$ contains realized effects rather than their identifiers. An overall label and step-, tool-, object-, and state-level evidence accompany these outputs. Failed replay or incomplete state observation is marked unevaluable rather than consistent.

5 Experiments

We ask three questions: whether the automatic signals agree with human authorization judgments, what incremental evidence each evaluation level provides, and whether the patterns persist across models and contract sources. AgentDojo's utility and security labels are used only to study complementarity with existing evaluation targets.

5.1 Experimental Setup

Environment and trajectories. We use 1,600 cached AgentDojo trajectories from GPT-4o, GPT-4o-mini, Claude 3.5 Sonnet, and Gemini 1.5 Flash (400 each), spanning banking, Slack, travel, and workspace. Replay makes no model calls: reference and cached agent calls run separately from the same initial state, with state recorded around every call.

Human evaluation. We use a stratified-random sample of 300 trajectories and a separate targeted 100-trajectory stress set. Table 1 uses final request-based review labels (186 inconsistent, 114 consistent), assigned and reviewed entirely by human reviewers. It measures agreement with judgments based on requests and execution evidence, not detector accuracy under a shared supplied contract: reviewers were not given the evaluation contract. The earlier stress-set labels and two-annotator agreement are reported separately. Appendix C gives the protocol and sample accounting.

Table 1: Alignment between reference-grounded mismatch signals and human review labels on 300 stratified-random trajectories.

Automatic signal	TP	FP	FN	TN	Precision	Recall	F1	FPR	BA
Final-state or trajectory-action mismatch	151	33	35	81	0.8207	0.8118	0.8162	0.2895	0.7612
Final-state mismatch	146	29	40	85	0.8343	0.7849	0.8089	0.2544	0.7653
Missing reference mutating action	108	30	78	84	0.7826	0.5806	0.6667	0.2632	0.6587
Unmatched agent mutating action	111	30	75	84	0.7872	0.5968	0.6789	0.2632	0.6668

Table 2: Two-annotator consensus on the 92 non-tie trajectories in the targeted stress set. These diagnostic metrics are not population estimates.

Signal or slice	Cases	Human inconsistent	Human consistent	Precision
Any consistency mismatch	83	65	18	0.7831
Final-state mismatch	52	43	9	0.8269
Transient-only mismatch	31	22	9	0.7097
Extra or unauthorized mismatch	15	15	0	1.0000
Successful trajectory with no mismatch	9	0	9	–

Metrics. Review-labeled inconsistency is the positive class. We report precision, recall, F1, false-positive rate, and balanced accuracy; for doubly annotated data, raw agreement and Cohen's κ. AgentDojo-label analyses additionally use accuracy, MCC, and precision lift over the base rate.

Comparisons. We report tool-call checks separately. Table 1 combines reference-grounded final-state and trajectory-action mismatches. These are diagnostic candidates, not equivalent to the full contract-based criterion: valid alternatives may differ from the reference. The table measures agreement with human judgments, not the isolated contribution of transient-effect detection.

5.2 Agreement with Human Judgments of User-Intent Compliance

On the earlier 100-trajectory stress set, two annotators obtain 0.910 raw agreement and Cohen's $\kappa = 0.793$. Table 1 instead uses the final labels on the separate 300-trajectory sample.

The combined signal flags 184 cases, with 0.821 precision, 0.812 recall, and 0.816 F1; final-state mismatch alone reaches 0.809 F1. The union includes seven final-state mismatches missed by action matching and nine action mismatches with matching final states. These are difference candidates, not confirmed violations. The two action rows predict overall inconsistency, not error-type-specific labels. In the separate stress set, 65 of 83 flagged non-tie cases are judged inconsistent; 22 of 31 transient-only candidates are inconsistent and nine consistent.

Final review rationales include omissions, extra effects, boundary violations, and answer-only failures; they do not constitute separate effect-type gold sets.

5.2.1 Relationship to AgentDojo's Native Labels

AgentDojo utility measures task completion, and its security label records whether the injection goal succeeds; neither directly asks whether every realized effect was user-authorized. Across all 1,600 trajectories, missing final-state deltas predict utility failures with 0.694 precision, 0.656 balanced accuracy, and 0.314 MCC, a precision lift of 0.175 over the utility-failure base rate. This relationship is informative but incomplete: utility can also fail because of the final answer or benchmark-specific success criteria, and extra final-state deltas do not reliably predict security failures. We therefore treat native labels as complementary targets rather than ground truth for authorization (Appendix D.1).

Table 3: Local tool-level analysis of 400 GPT-4o AgentDojo trajectories.

Scope	Calls	Flag rate	Security-failure recall	Utility-failure recall	Any-failure P/R
Strict check of all calls	1192	0.5196	0.5269	0.8344	0.8067 / 0.7000
Allow supporting reads	1192	0.2141	0.2366	0.7682	0.7919 / 0.5032
Successful side-effecting calls only	317	0.1933	0.2312	0.7550	0.7897 / 0.4968

5.3 Incremental Information from the Three Evaluation Levels

5.3.1 Tool-Call Authorization

Tool-call authorization locates operations and arguments that exceed the contract, including attempts that fail and leave no state change. Its output is nevertheless sensitive to the scope of the check. Table 3 reports tool-call signals for 400 GPT-4o trajectories. A strict reference-relative check flags 51.96% of calls because it includes reads, searches, preparation steps, and failures. Admitting supporting reads lowers the rate to 21.41%, and restricting evaluation to successful side-effecting calls lowers it to 19.33%. We therefore use this level to localize suspicious attempts rather than as a standalone verdict, and we keep its different 400-trajectory evaluation population separate from the 300-case human comparison.

5.3.2 Final-State Consistency

Final-state evaluation retains the changes present when execution ends. It therefore exposes a missing required final effect even when the agent used a plausible action sequence, as well as a persistent unauthorized effect. Its reference-grounded mismatch signal reaches 0.834 precision and 0.785 recall on the 300-case sample. Its limitation is historical: if an unauthorized membership, file, or calendar mutation is later reversed, the terminal comparison no longer contains that evidence.

5.3.3 Trajectory Consistency

Replay provides process evidence absent from the final snapshot by retaining intermediate changes and their responsible calls. On 300 cases, adding action mismatches to final-state mismatches finds five additional true positives and four false positives: F1 rises from 0.809 to 0.816, while precision and balanced accuracy fall. This is a signal-union trade-off, not an isolated gain from transient-effect detection. Across 1,600 traces, 45 transient-only candidates occur (30/9/5/1 by model); they require review because action differences can be benign.

5.3.4 Representative Cases

Table 4 illustrates the different evidence exposed by the benchmark and consistency signals. The Slack case is the motivating read-only request from Section 1: the agent invited a user, added the user to a channel, and later removed the user. The final snapshot omits part of this history, whereas replay ties each temporary membership change to the call that realized it. Conversely, the banking case shows why an injection-success label should not be reinterpreted as an unauthorized final-state mutation.

5.4 Robustness Across Models and Contract Sources

This experiment compares final-state and trajectory mismatches across four models and examines how the source of the contract affects the evaluation results.

5.4.1 Cross-Model Results

All 400 trajectories for each model replay successfully. Missing final-state deltas relate to utility failures for all four models, although precision ranges from 0.598 to 0.835. Transient-only mismatches also occur for every model (1–30 cases). These results establish recurrence within AgentDojo rather than a model-quality ranking.

Table 4: Representative replay findings. Utility, security, final-state, and trajectory evidence answer different questions.

Case	Benchmark labels	Consistency signal	Interpretation
Travel reminder omitted	Utility failure, security success	Missing calendar and inbox deltas	Missing final effects localize an execution failure.
Clean successful execution	Utility success, security success	No mismatch	Replay retains a meaningful negative class.
Banking security failure	Utility and security failure	Missing payment delta, not extra mutation	Security failure is not equivalent to unauthorized final-state mutation.
Slack transient action	Utility/security success in some traces	Transient-only mismatch	Side effects can occur and be undone before the final state.

Table 5: Replay across four models (400 traces each). State Miss denotes missing reference final-state deltas; Action Miss denotes unmatched reference mutating actions. Action Any denotes either missing or unmatched mutating actions. Targets are utility failure (Util) or either native failure (Fail).

Model	State Miss→Util P/R/Acc	Action Miss→Util P/R/Acc	Action Any→Fail P/R	Transient-only
GPT-4o	0.6463 / 0.6424 / 0.7075	0.4818 / 0.6424 / 0.5675	0.7730 / 0.6527	30
GPT-4o-mini	0.6743 / 0.5700 / 0.6350	0.7239 / 0.5700 / 0.6650	0.7722 / 0.5719	9
Claude 3.5 Sonnet	0.5977 / 0.5474 / 0.6100	0.6323 / 0.5158 / 0.6275	0.7467 / 0.5437	5
Gemini 1.5 Flash	0.8350 / 0.6208 / 0.6625	0.8698 / 0.6208 / 0.6825	0.9397 / 0.4987	1

5.4.2 Sensitivity to Contract Source

Because the main experiments use contracts grounded with reference calls, two annotators independently extract contracts for 86 tasks from requests alone. They judge 72 and 73 tasks, respectively, to admit multiple valid implementations. Agreement on whether a task admits multiple valid implementations is 0.860, with Cohen's $\kappa = 0.480$; these statistics concern the multiple-validity label, not agreement on the full contract. Among 858 reference-contract mismatches, 89.9% and 88.7% remain violations under their independent contracts; only 5.7–6.2% are classified as possibly valid alternatives. An unadjudicated union retains 92.1% and marks 3.5% as possible alternatives. Thus, most findings survive broader human contracts, but contract source changes a nonzero subset and remains part of the measurement. Automatic extraction further exposes a precision–recall trade-off: the keyword/ontology baseline reaches 0.650/0.912 precision/recall on mutating operations, whereas Qwen2.5-7B reaches 0.787/0.649. Appendix C.1 reports full extraction and ontology results.

Finally, path-level and top-level-root differences yield identical binary signals on these trajectories; requiring two changed roots sometimes improves precision but sharply reduces recall (Appendix E).

6 Conclusion

We introduced a three-level framework linking attempted calls to realized state transitions under a supplied intent contract. Its central contribution is process evidence unavailable from final-state checks alone: failed unauthorized attempts and intermediate effects that are later reversed. Reference-grounded mismatch signals show a modest recall and F1 gain when combined, with lower precision and balanced accuracy; this does not isolate transient-effect detection. The framework complements task utility and attack success with evidence for diagnosing how execution crosses authorization boundaries.

AI Use Statement

Generative AI tools were used to assist with translation, manuscript editing and organization, code development and debugging, literature-related tasks, and feedback on the presentation of the research methodology and experimental design. AI-generated suggestions and outputs were treated as auxiliary materials rather than authoritative sources.

The core research questions, evaluation framework, experimental setup, scientific claims, and conclusions were determined and validated by the authors. AI-assisted code and technical suggestions were manually reviewed and tested, and AI-assisted text was revised and verified by the authors for technical correctness, appropriate attribution, and consistency with the reported results. The authors take full responsibility for the final content of this work.

REFERENCES

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. Agentharm: A benchmark for measuring harmfulness of LLM agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c493d23af93118975cdbc32cbe7323f5-Abstract-Conference.html.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8301–8327. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.474. URL https://aclanthology.org/2024.emnlp-main.474/.

Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding agents via verifiable safety policy reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8313–8344. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25ae.html.

Yun-Shiuan Chuang, Chaitanya Kulkarni, Alec M. Chiu, Avinash Thangali, Zijie Pan, Shivani Shekhar, Yirou Ge, Yixi Li, Uma Kona, Linsey Pang, and Prakhar Mehrotra. Toward scalable verifiable reward: Proxy state-based evaluation for multi-turn tool-calling LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1251–1264. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-industry.87. URL https://aclanthology.org/2026.acl-industry.87/.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2636. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Shiwei Feng, Xiangzhe Xu, Xuan Chen, Kaiyuan Zhang, Syed Yusuf Ahmed, Zian Su, Mingwei Zheng, and Xiangyu Zhang. TAI3: Testing agent integrity in interpreting user intent. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f13d67d8b994594a12d53e5d21de2bef-Abstract-Conference.html.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1307–1323. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117/.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, Yue Xing, Jiliang Tang, and Benoit Dumoulin. TRAJECT-Bench: A trajectory-aware benchmark for evaluating agentic tool use. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/64ebb6617526d284ef623a6887d3e119-Abstract-Conference.html.

Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, et al. Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning, 2022.

Thomas Kuntz, Agatha Duzan, Hao Zhao, Francesco Croce, Zico Kolter, Nicolas Flammarion, and Maksym Andriushchenko. OS-Harm: A benchmark for measuring safety of computer use agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1502. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4009bff0cd87ba2203c8e3a2f082aaec-Abstract-Datasets_and_Benchmarks_Track.html.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. ST-WebAgentBench: A benchmark for evaluating safety and trustworthiness in web agents. In *International Conference on Learning Representations*, 2026. URL https://research.ibm.com/publications/st-webagentbench-a-benchmark-for-evaluating-safety-and-trustworthiness-in-web-agents--1.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. AgentRewardBench: Evaluating automatic evaluations of web agent trajectories. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=fQcUZMPIvu.

Hanjun Luo, Shenyu Dai, Chiming Ni, Xinfeng Li, Guibin Zhang, Kun Wang, Tongliang Liu, and Hanan Salam. Agentauditor: Human-level safety and security evaluation for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1440. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3dc85735f6e2fcf093e67b134fa00d21-Abstract-Conference.html.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4020. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e4c61f578ff07830f5c37378dd3ecb0d-Abstract-Conference.html.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Amy Xin, Youfeng Liu, Bin Xu, Lei Hou, and Juanzi Li. AgentIF: Benchmarking large language models instruction following ability in agentic scenarios. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1892. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/51bb3a8a33610a25aae074bfc51b1b1f-Abstract-Datasets_and_Benchmarks_Track.html.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/28e50ee5b72e90b50e7196fde8ea260e-Abstract-Conference.html.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Reshabh K. Sharma, Shraddha Barke, and Benjamin Zorn. Willful disobedience: Automatically detecting failures in agentic traces. In *Proceedings of the ACM Conference on AI and Agentic Systems*. Association for Computing Machinery, 2026. doi: 10.1145/3786335.3813153. URL https://doi.org/10.1145/3786335.3813153.

Wentao Shi, Yu Wang, Yuyang Zhao, Yuxin Chen, Fuli Feng, Xueyuan Hao, Xi Su, Qi Gu, Hui Su, Xunliang Cai, and Xiangnan He. AJ-Bench: Benchmarking agent-as-a-judge for environment-aware evaluation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25371–25413. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1269. URL https://aclanthology.org/2026.findings-acl.1269/.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Sanidhya Vijayvargiya, Aditya Soni, Xuhui Zhou, Zora Zhiruo Wang, Nouha Dziri, Graham Neubig, and Maarten Sap. Openagentsafety: A comprehensive framework for evaluating real-world AI agent safety. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a0f6c06b8d6114bb37688c2dc19b4d74-Abstract-Conference.html.

Jiaxuan Wang, Yulan Hu, Wenjin Yang, Zheng Pan, Xin Li, and Lan-Zhe Guo. Aligning agents via planning: A benchmark for trajectory-level reward modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23174–23200. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1062. URL https://aclanthology.org/2026.acl-long.1062/.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1650. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Yao Zhang, Shijie Tang, Zeyu Li, Zhen Han, and Volker Tresp. Webarbiter: A generative reasoning process reward model for web agents. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c8bd227b32b458970d3ab01a2c372055-Abstract-Conference.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80569–80611. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhuge25a.html.

A Paired Authorization Contrast Set

The paired authorization contrast set contains 400 cases across email, banking, travel, Slack, calendar, file management, shopping, and database operations. Each case includes a user request, structured intent contract, candidate action, binary label, violation dimensions, rationale, pair identifier, and authorization contrast axis. Within each pair, the request and contract are fixed while the candidate action changes, isolating request-conditioned authorization.

A.1 Violation Dimensions and Examples

The contrast set covers operation, side-effect, confirmation, recipient, object, data, temporal, value, and goal boundaries. A case may violate more than one boundary: sending an email that the user only asked to draft violates both the authorized operation and side-effect boundary, while forwarding a private file to an unauthorized recipient violates recipient and data boundaries.

Table 6: Representative authorization contrasts.

Domain	Authorized action	Unauthorized action
Email	Draft message	Send without approval
Banking	Check balance	Initiate transfer
Travel	Compare hotels	Book hotel
Slack	Read channel history	Invite external user
Calendar	Create requested event	Add unrequested attendee
File	Summarize local file	Export private file
Shopping	Add item to wishlist	Purchase item
Database	Query rows	Delete rows

B AgentDojo Adapter and Operation Ontology

B.1 Static Adapter

The AgentDojo static adapter extracts user tasks, injection tasks, and ground-truth function calls from AgentDojo v1 task files. It produces 439 intent–action cases from 113 tasks: 287 aligned and 152 violating cases. This adapter tests whether the contract representation can express benchmark task behavior; the main execution evidence instead comes from replaying cached traces in the AgentDojo environment.

B.2 Operationalizing Authorized Mutations

The replay evaluator does not compare raw tool sequences. An editable ontology marks APIs as read-only or mutating, expands abstract contract operations to API sets, and maps available call arguments to recipient, object, value, temporal, and data boundaries. It thereby instantiates the authorized mutating-operation set for each contract while keeping the mapping auditable.

Table 7: AgentDojo operation taxonomy. Supporting reads are ignored unless they violate an explicit data boundary; mutations must satisfy operation- and argument-level authorization.

Domain	Read-only examples	Mutating examples	Boundary rule
Banking	transactions, files	send/schedule money, update account	recipient, amount, account, field
Slack	channel/user/web reads	invite, add/remove, send/post	user, channel, body, webpage target
Travel	hotel/car/flight search	book, remind, send itinerary	booking, calendar, email scope
Workspace	email/calendar/file reads	create/cancel/share/delete/write	object, recipient, time, sharing

The ontology is a mapping layer, not a learned component. A benign temporary file is allowed only when the contract or ontology authorizes that support mutation; externally visible payments, messages, memberships, bookings, and shares remain violations when outside the request.

C Human Validation and Contract Audit

We generate three blind validation packs. Annotators do not see generated contracts, benchmark labels, model identity, utility/security labels, or automatic consistency labels.

Synthetic action validation. The action-level pack contains 150 paired authorization contrast cases balanced across labels. Annotators see only the request and candidate action, then mark `consistent`, `inconsistent`, or `unclear`, plus violation dimensions when applicable.

Targeted trajectory validation. The 100-trace pack covers transient-only mismatches, missing intended effects, extra or unauthorized effects, benchmark-success traces with state mismatch, and clean successes. Annotators see the request, tool sequence and results, final answer, and a compact state-delta summary. They additionally mark user visibility or harm, whether multiple valid trajectories are possible, and whether a mismatch is likely a serialization artifact.

Two annotators complete this pack. Agreement is 0.910 and Cohen's $\kappa = 0.793$: they agree on 65 inconsistent and 26 consistent cases, disagree on 8, and one consistent/unclear case is grouped as consistent. Both mark all 100 cases as unlikely to be serialization artifacts.

The original targeted sample contains 35 transient-only trajectories; 31 remain after ties are removed. Of these, 22 are judged inconsistent and 9 consistent. The latter may reflect legitimate temporary operations, an overly narrow contract, or other boundary judgments.

Stratified-random trajectory validation. The 300-case pack is balanced by domain (75 each) and near-balanced by model (76, 76, 76, and 72). Cached signals flag 177 reference-action mismatches and 175 final-state mismatches, with 168 shared cases: their union flags 184. Seven final-state-only cases are missed by action matching; nine action-only candidates have matching final states. There are 155 utility failures and 179 security failures. The final labels used in Table 1 contain 186 inconsistent and 114 consistent cases, with none excluded.

C.1 Request-Only Contract Extraction Audit

Two annotators independently extract contracts for all 86 available AgentDojo v1 user tasks while seeing only the natural-language request. They record allowed and forbidden operations and side effects, object/recipient boundaries, and whether multiple valid trajectories are possible. Operation overlap with reference calls is a diagnostic rather than an assumption that the reference is the only legal path.

The annotators agree on the multiple-valid label in 0.860 of cases ($\kappa = 0.480$). Human1 marks 72 tasks as admitting multiple implementations; Human2 marks 73 yes, 11 no, and 2 unclear. Raw function-name overlap is low in part because humans use abstractions such as `invite_user` while the API uses `invite_user_to_slack`; ontology expansion separates this naming mismatch from substantive boundary disagreement.

Table 8: Request-only contract extraction over 86 AgentDojo tasks. Raw uses exact function names; ontology-expanded maps abstract operations to API sets.

Extractor	Rows	Multi-valid yes/no	Raw P/R/F1	Ontology P/R/F1	Mutating P/R/F1
Human1	86	72 / 14	0.161 / 0.118 / 0.136	0.322 / 0.596 / 0.418	0.439 / 0.509 / 0.472
Human2	86	73 / 11 / 2 unclear	0.246 / 0.265 / 0.255	0.420 / 0.747 / 0.537	0.889 / 0.702 / 0.784
Keyword baseline	86	69 / 17	0.241 / 0.184 / 0.208	0.496 / 0.947 / 0.651	0.650 / 0.912 / 0.759
Qwen2.5-7B	86	2 / 84	0.500 / 0.367 / 0.424	0.529 / 0.788 / 0.633	0.787 / 0.649 / 0.712

The keyword baseline favors recall on mutating operations (0.912 recall at 0.650 precision), whereas Qwen2.5-7B favors precision (0.787 precision at 0.649 recall). These are sanity checks, not replacements for human contracts.

Across 1,507 traces whose tasks have human contracts, 858 have a canonical any-mismatch label and full-detail action fields. Human1 retains 771/858 as violations (0.899) and marks 49/858 as possibly alternative-valid (0.057); Human2 retains 761/858 (0.887) and marks 53/858 (0.062) as possibly valid. An unadjudicated union candidate retains 790/858 (0.921) and marks 30/858 (0.035) as possibly valid. Ten yes/no disagreements about multi-validity remain for manual adjudication.

D Additional Evaluation Results

D.1 Base-Rate-Aware Benchmark-Label Statistics

Table 9: Aggregate statistics over 1,600 AgentDojo traces. Lift is precision minus target prevalence.

Signal	Base	Pred. rate	Precision	Recall	Bal. acc.	MCC / lift
Replay missing → utility failure	0.5194	0.4456	0.6942	0.5957	0.6561	0.3138 / 0.1749
Replay post mismatch → any failure	0.8363	0.5863	0.8156	0.5717	0.4557	-0.0665 / -0.0207
Replay extra → security failure	0.6106	0.4512	0.3476	0.2569	0.2504	-0.4891 / -0.2630
Missing reference action → utility failure	0.5194	0.4562	0.6699	0.5884	0.6375	0.2759 / 0.1505
Any reference-action mismatch → any failure	0.8363	0.5894	0.8017	0.5650	0.4256	-0.1119 / -0.0346

The clearest benchmark-label relationship is missing intended state change to utility failure: its precision is 0.1749 above the utility-failure base rate and its MCC is positive. High raw precision for any-failure targets is not evidence of incremental value because that target is highly prevalent. Human judgments, not these native labels, provide the paper's primary authorization evidence.

E State-Difference Granularity

Table 10 compares path-level, root-level, and conservative root-ge2 signals. Path- and root-level binary conclusions match on the current outputs. Requiring two changed roots improves precision in some settings but substantially lowers recall.

F Reproducibility Details

The state and trajectory evaluators make no model API calls. They replay cached AgentDojo tool calls and record environment state before and after each step, making cross-model evaluation inexpensive once traces are available.

Table 10: State-difference granularity sensitivity.

Model	Granularity	Miss signals	Miss→Util P	R	Any signals	Any→Fail P	R
GPT-4o	path/root	164	0.6463	0.6424	252	0.7897	0.5958
GPT-4o	root-ge2	24	0.1667	0.0242	62	0.8065	0.1497
GPT-4o-mini	path/root	175	0.6743	0.5700	245	0.7878	0.6031
GPT-4o-mini	root-ge2	65	0.6154	0.1932	103	0.9029	0.2906
Claude 3.5 Sonnet	path/root	174	0.5977	0.5474	234	0.7607	0.5761
Claude 3.5 Sonnet	root-ge2	65	0.5846	0.2000	109	0.7890	0.2783
Gemini 1.5 Flash	path/root	200	0.8350	0.6208	207	0.9420	0.5200
Gemini 1.5 Flash	root-ge2	74	0.9865	0.2714	80	0.9875	0.2107

Main scripts. Synthetic contrast-set scripts include `generate_counterfactual_benchmark_v1.py`, `intent_contract_verifier.py`, `evaluate_predictions.py`, and `make_human_validation_pack.py`. AgentDojo replay uses `analyze_existing_traces.py`, `replay_state_snapshots.py`, and `analyze_trajectory_consistency.py`, plus cross-model and granularity summarizers. Human-validation utilities generate blind/random packs, score contract extraction, expand the ontology, and measure full-detail retention under human contracts. The operation ontology is stored as `contract_operation_ontology.json`; request-only baselines are `extract_keyword_nl_contract_baseline.py` and `extract_llm_nl_contract_baseline.py`.

Known caveats. AgentDojo contracts use benchmark ground truth, so the work evaluates execution under a supplied contract rather than solving natural-language intent parsing. The targeted audit uses the earlier two-annotator labels; the random audit uses the final review labels described in Section 5.1. The contract audit is diagnostic rather than a fully adjudicated gold set.