

Cross-Domain Evaluation of Semantic Trace Summaries: A Leakage-Controlled Study of 4,448 τ -Bench Trajectories

George Weale
Columbia University (2026)
george.weale@columbia.edu

Abstract—External state matters for tool-using agents, but a plausible state-aware summary is not automatically a useful evaluator. We test whether 14 generic features—including state-changing-call burden, repeated and failed writes, confirmation, read-after-write closure, argument footprint, and handoffs—add out-of-sample information beyond ordinary interaction traces. The outcome is the released τ -bench task reward, not a label generated by our rule. The cohort contains 4,448 public trajectories from four agents, 278 tasks, three state-changing domains, and four trials per agent-task pair. Every prediction holds out both its task fold and its agent model; uncertainty resamples tasks, not individual trajectories. An interaction-trace model obtains Brier score 0.1312. Adding the semantic summary worsens it to 0.1369, a paired difference of +0.0057 (task-cluster bootstrap 95% interval -0.0002 to $+0.0117$), where positive is worse. Excluding 56 infrastructure or missing-reward rows gives the same null result. A hashed tool-name comparator also receives no reliable gain from the semantic features. Post-hoc nested calibration narrows the gap to +0.0033 but worsens both absolute Brier scores. Across all 12 model-domain cells, six favor semantics and seven intervals exclude zero, split three for semantics and four for the trace. The global null therefore masks opposed transport, not a general semantic gain. The finding is negative but useful: generic trace morphology does not substitute for an independently authored, task-conditioned state contract.

Index Terms—agent evaluation, tool use, state change, cross-fitting, calibration, negative result

I. THE QUESTION IN PLAIN LANGUAGE

Suppose an agent says it updated a reservation. A conventional trace records the conversation, tool calls, and errors. A more semantic trace can also say: there were two writes; one was repeated; the user confirmed before the write; and the agent read the reservation afterward. Those facts are easier to audit than raw text. The tempting next claim is that they make success easier to predict.

This paper tests that claim without letting the proposed representation define its own answer. Our question is:

Do compact, task-agnostic summaries of state-changing traces improve prediction of independently graded task success when both the task and agent model are new to the predictor?

The answer for the studied τ -bench cohort is no. This matters because state-based evaluation is already central to

agent benchmarks. τ -bench compares final database state and policy behavior with an annotated goal [1]; AppWorld uses database-state unit tests and collateral-damage checks [3]; and ToolSandbox evaluates milestones over stateful trajectories [4]. The open problem is not whether state matters. It is which representation generalizes beyond the tasks and systems that inspired it.

This study replaces an earlier circular simulation. That simulator generated latent labels using violation-over-uncertainty precedence, then evaluated a rule implementing the same precedence. A disjoint random seed protected only against row reuse; it did not create an independent target. The simulation is therefore removed rather than repackaged as evidence.

II. INDEPENDENT DATA, NARROW SCOPE

A. Released outcomes

τ -bench is a simulated customer-service benchmark with an LLM user, domain policy, API tools, task-specific evaluation criteria, and database state [1]. We use the public trajectory files distributed through the official leaderboard repository [2]. The label is each trajectory’s released binary reward. It is computed by the benchmark, not by this paper.

The cohort combines four standard v1.0.1 submissions: Claude Sonnet 4.5, GLM-5, GPT-5.2 high, and Qwen3.5-397B. All use the GPT-5.2 user simulator and four trials per task. We retain the airline, retail, and telecom domains because they contain state-changing customer-service workflows. Banking knowledge is excluded because its primary difficulty is retrieval.

There are 50 airline tasks, 114 retail tasks, and 114 telecom tasks. Crossing 278 tasks with four agents and four trials yields 4,448 trajectories. The raw pass rates range from 72.0% to 97.8% across agent-domain cells (Fig. 1). Fifty-six GLM rows have no numeric reward because of recorded infrastructure termination; the submission states that these count as failures, so the primary analysis follows that convention. A sensitivity analysis removes them.

B. Source-integrity gate

Inclusion required public trajectories, common benchmark and user-simulator settings, four trials, consistent embedded

a Released benchmark outcomes

Claude Sonnet 4.5	72.0%	72.4%	84.9%
GLM-5	82.5%	73.7%	86.8%
GPT-5.2 high	83.0%	81.6%	89.7%
Qwen3.5-397B	81.5%	84.4%	97.8%
	Airline	Retail	Telecom

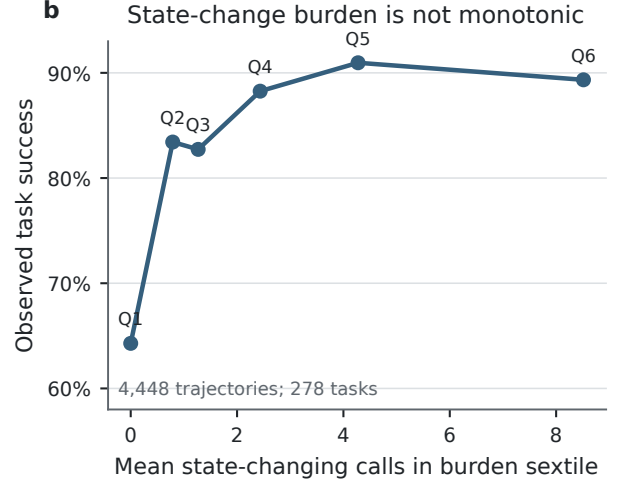


Fig. 1. The real evaluation cohort. (a) Released pass rates for every included agent-domain cell. (b) Descriptive success by sextile of state-changing-call burden. More writes are not monotonically associated with failure, illustrating why a learned and leakage-controlled comparison is needed.

model metadata, and raw rewards that reproduce the submission’s reported pass-1 rate. Every selected file is pinned by URL and SHA-256. Two candidates failed this gate before fitting: a linked Gemini 3 Flash airline file did not reproduce the posted pass rate, and two GPT-5.2-none files embedded a different reasoning setting and inconsistent outcomes. This selection rule concerns artifact identity, not whether a run favors our method. The complete audit is released with the analysis.

The trajectories represent benchmark interactions, not deployed customers. The released feature table omits conversation text, policies, tool arguments, and synthetic customer records; it contains anonymous identifiers, numeric features, and outcomes only.

III. FROM A TRACE TO TESTABLE FEATURES

A. What the models may see

Feature extraction reads only ordered messages, tool-call names, tool-call arguments as structural counts, tool errors, and termination. It never reads task evaluation criteria, expected actions, database checks, natural-language assertions, reward breakdowns, or policy text. Changing those forbidden fields leaves the feature vector unchanged in the executable tests.

The *interaction trace* uses domain indicators, message and turn counts, tool-call burden, tool-error fraction, empty-action status, and abnormal termination. The *semantic summary* adds 14 generic quantities:

- whether and how often the trace invokes state-changing tools;
- unique, repeated, and failed writes;
- whether a later read addresses the same lexical resource as a successful write;
- whether the last user message explicitly confirms a write;

- retry-after-error, argument footprint, resource breadth, final-action type, handoff count, and unclassified-tool fraction.

A transparent verb-prefix lexicon classifies tools as reads, writes, handoffs, or other. This is deliberately compact and domain-agnostic. It is fixed in the artifact but exploratory, not preregistered.

Two comparators test whether raw action identity is more useful than the semantic compression. A 32-bin SHA-256 hashing transform forms a fixed tool-name bag. The *full* representation joins that bag and the semantic summary. All learned models use standardized ridge logistic regression with a fixed penalty of 2.0; no outcome-driven feature selection or penalty search is performed.

B. Holding out both ways leakage can enter

Repeated trials make a random row split unsafe: three runs of task 42 could teach the model how to score the fourth. Model behavior can leak similarly: a predictor can learn that one agent is terse or error-prone and appear to generalize.

We hash each domain-task identifier into one of eight folds. For every fold and every agent, the test block contains that fold from that agent. Training excludes *all* trajectories from the task fold and *all* trajectories from the held-out agent. Thus every trajectory is predicted once per method by a fit that has seen neither its task nor its model. Domain remains observable; the target is a new task and model within the three studied domains, not a new domain.

The primary score is the Brier score

$$B = \frac{1}{N} \sum_{i=1}^N (p_i - y_i)^2, \quad (1)$$

which evaluates probability accuracy rather than thresholded labels [5]. The primary contrast is $D = B_{\text{semantic}} - B_{\text{trace}}$; negative favors the semantic summary. We resample the 278

TABLE I
OUT-OF-MODEL, OUT-OF-TASK PREDICTIONS

Representation	Brier ↓	AUROC ↑	Log loss ↓
Domain prevalence	0.1394	0.542	0.451
Interaction trace	0.1312	0.663	0.433
Trace + semantic	0.1369	0.664	0.473
Trace + tool bag	0.1305	0.699	0.429
Full	0.1328	0.735	0.468

domain-task clusters 10,000 times and keep all 16 agent-trial observations together [7]. AUROC and log loss are secondary diagnostics.

IV. RESULTS

The interaction trace reaches Brier 0.1312. Adding semantic features reaches 0.1369 (Table I). The paired difference is +0.0057 with a 95% task-cluster interval of -0.0002 to $+0.0117$ (Fig. 2). The point estimate is adverse and the interval contains no defensible improvement. Removing all 56 missing-reward rows gives +0.0010 (-0.0045 to $+0.0065$), the same substantive result.

The semantic increment is heterogeneous but not reliably helpful. Its descriptive Brier change is +0.0107 for Claude, +0.0141 for GLM, -0.0024 for GPT-5.2 high, and +0.0005 for Qwen. By domain it improves retail descriptively (-0.0176) but worsens airline ($+0.0198$) and telecom ($+0.0229$). These strata are explanatory diagnostics, not separately powered confirmatory tests.

The hashed tool bag has the best Brier score, 0.1305, but differs from the trace by only -0.0007 (-0.0044 to $+0.0031$). Adding semantics to it increases AUROC from 0.699 to 0.735 yet worsens Brier by +0.0023 (-0.0046 to $+0.0094$) and log loss from 0.429 to 0.468. The full model therefore ranks some cases differently without producing better probabilities on the declared primary metric.

A. Post-hoc calibration exposes opposed transport

The primary models share a fixed ridge penalty but can still differ in calibration. Temperature-style post-hoc calibration is a standard way to separate ranking from probability scaling [6]. As a secondary analysis, we fit a logistic intercept and slope to the logit of each frozen probability. For every test trajectory, its calibrator excludes both its agent model and its task fold. This produces 64 audited fits: two representations, four held models, and eight held task folds.

The frozen comparison retains its primary interval: +0.0057 (-0.0002 to $+0.0117$), from the declared seed 20260803. Recalibration narrows the gap to +0.0033 (95% task-cluster interval -0.0001 to $+0.0066$), but it does not reverse the null. Absolute Brier worsens from 0.1312 to 0.1351 for the trace and from 0.1369 to 0.1384 for trace plus semantics. Thus calibration is not a hidden rescue for the generic semantic summary.

Figure 4 maps every model-domain cell rather than searching for a favorable subgroup. After recalibration, six of 12 cells

favor semantic features. Seven cell intervals exclude zero: three favor semantics and four favor the trace. Retail improves for Claude, GPT, and Qwen, while telecom worsens for Claude and GLM. Qwen also has trace-favoring intervals in airline and telecom. The aggregate null therefore masks opposed transport, not a stable semantic gain. This map and calibrator were selected after the primary analysis and are descriptive secondary evidence.

V. WHAT THE NEGATIVE RESULT TEACHES

The result does not imply that external state is unimportant. It separates two ideas that are easy to conflate.

First, a *trace summary* says how a run looks: how many writes occurred, whether they repeated, and whether a read followed. Second, a *state contract* says what the run was supposed to do: which reservation could change, which fields should equal what values, and which side effects were forbidden. The first can be computed without understanding the task. The second cannot.

Task success in τ -bench depends on target-specific values and policy conditions. Two traces can each contain one confirmed write and one verification read, while one changes the correct order and the other changes the wrong one. The generic summary deliberately discards that distinction. Its failure to travel is therefore informative: a compact evaluator should not be called semantic merely because it labels tools as reads and writes.

The next defensible experiment is harder. Independent authors should translate task instructions into typed preconditions, target entities, permitted equivalences, required postconditions, and forbidden side effects *before* viewing a run. A separate blinded adjudication should label success, failure, or insufficient evidence. Evaluation should measure both agreement and diagnostic localization, with task and environment held out. That design could test state contracts; this paper tests only generic trace morphology.

VI. LIMITATIONS AND REPRODUCIBILITY

This is retrospective benchmark research. The four agents were selected by a source-integrity gate, not randomly sampled from all agents. The three domains share a benchmark family and user simulator. The lexical classifier can misclassify tools, confirmation is a surface cue, and lexical resource overlap is not entity resolution. Feature design was frozen before fitting but was not publicly preregistered. The cluster bootstrap describes task variation conditional on the released trajectories; it does not include model reruns, future task revisions, or new domains. Database and policy rewards are more independent than labels generated by our rule, but they are still benchmark oracles rather than human production outcomes.

The local package releases the complete numeric cohort, cross-fitted predictions, split inventory, task-cluster intervals, source-integrity audit, secondary calibrator parameters, exhaustive cell maps, and figures. Twelve raw files are pinned by URL and SHA-256 but not copied into the repository. The downloader verifies every hash, and

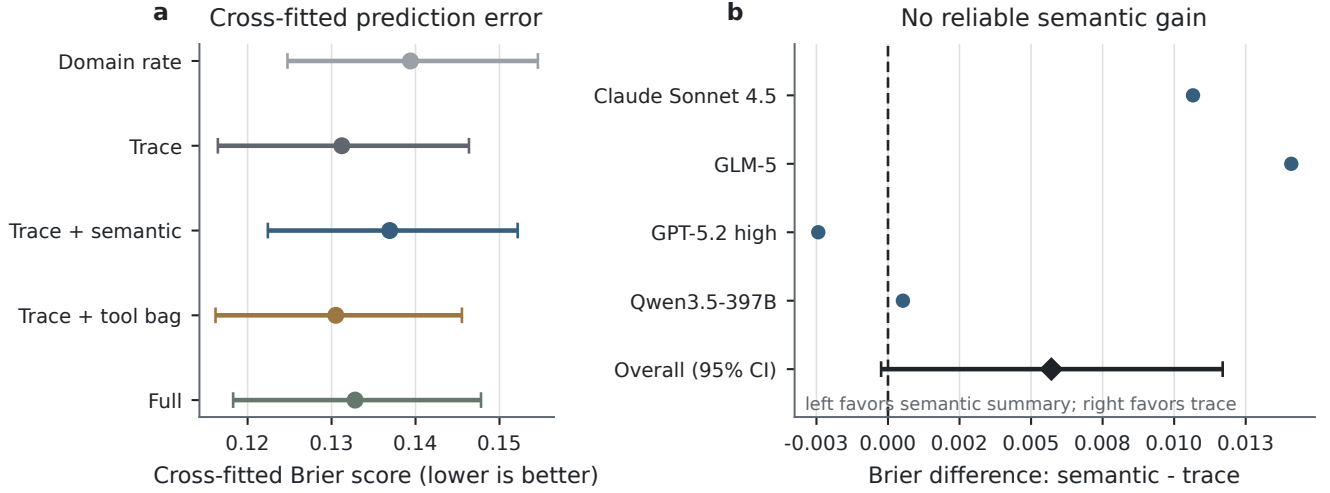


Fig. 2. Primary cross-fitted result. (a) Brier scores with task-cluster bootstrap 95% intervals. (b) Semantic-minus-trace differences by held-out agent and overall. Negative favors semantic features. The aggregate interval crosses zero and the point estimate is worse.

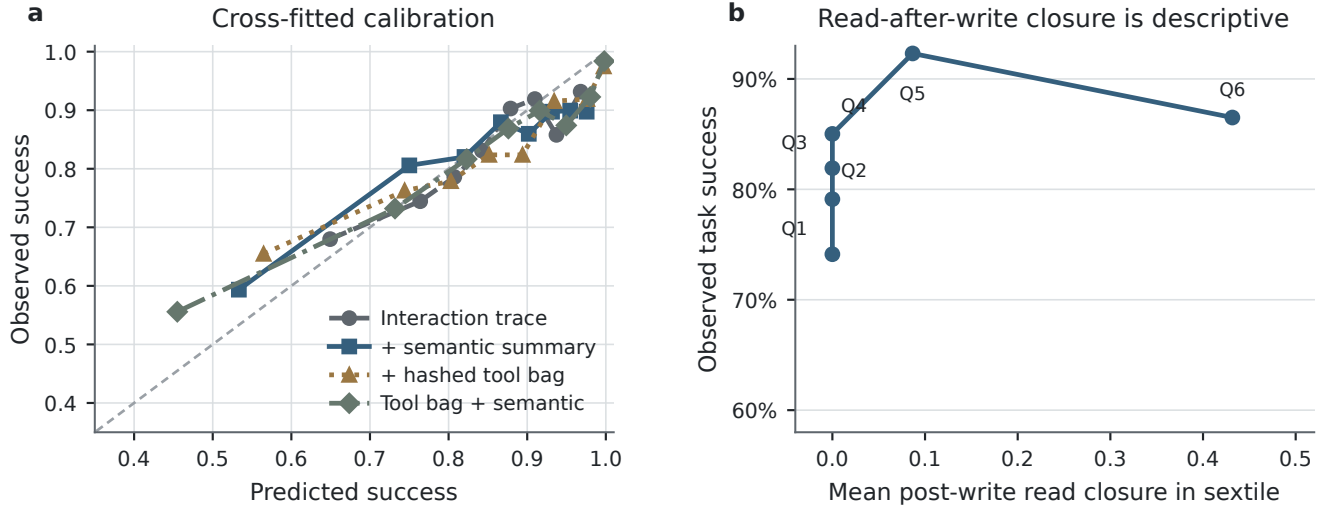


Fig. 3. Why ranking alone is insufficient. (a) Equal-count calibration bins from cross-fitted probabilities; the diagonal is ideal calibration. (b) descriptive success by read-after-write closure sextile. Closure correlates with outcomes nonlinearly but does not yield an incremental calibrated gain once transported to a held-out task and model.

the primary analysis refuses count or pass-rate mismatches. Regression suites check feature isolation, model/task separation, prediction inventory, privacy-reduced exports, the negative result, manifest hashes, and zero model/fold overlap in nested calibration. The secondary command is `python -B analyze_transport_calibration.py`.

VII. CONCLUSION

On 4,448 independently graded τ -bench trajectories, a generic semantic trace summary does not improve calibrated success prediction over an ordinary interaction trace when both tasks and agent models are held out. The null survives removal of infrastructure failures, comparison with a hashed tool bag, and leakage-controlled post-hoc recalibration. The

exhaustive cell map shows opposed model-domain effects rather than a general semantic gain. The useful research direction is not a richer list of generic trace counts. It is an independently authored, task-conditioned state contract with blinded adjudication and leakage-controlled evaluation.

REFERENCES

- [1] S. Yao, N. Shinn, P. Razavi, and K. Narasimhan, “ τ -bench: A benchmark for tool-agent-user interaction in real-world domains,” in *ICLR*, 2025.
- [2] Sierra Research, “tau2-bench: Code, task data, and public leaderboard submissions,” GitHub repository, commit 165848b9a78ccad7ca7fe7c89c688b58e6501219, accessed Aug. 3, 2026.
- [3] H. Trivedi et al., “AppWorld: A controllable world of apps and people for benchmarking interactive coding agents,” in *ACL*, 2024, pp. 16022–16076.

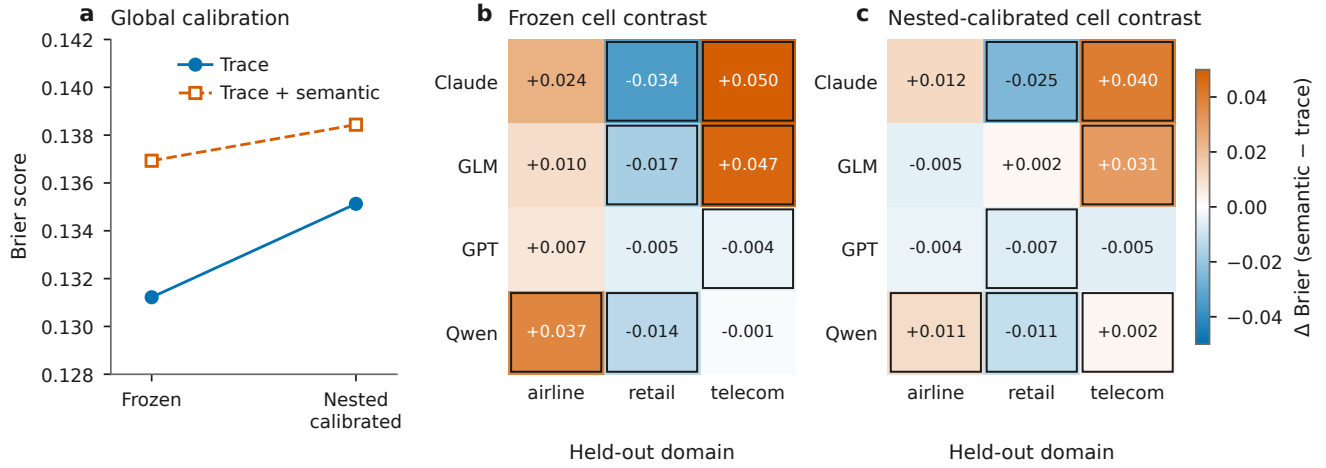


Fig. 4. Post-hoc calibration and exhaustive transport map. (a) Nested out-of-model, out-of-fold calibration narrows but does not reverse the global Brier gap; lower is better. (b–c) Frozen and recalibrated semantic-minus-trace Brier for all four models and three domains. Negative cells favor semantics; outlined cells have task-cluster intervals that exclude zero.

- [4] J. Lu et al., “ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities,” in *Findings of NAACL*, 2025.
- [5] G. W. Brier, “Verification of forecasts expressed in terms of probability,” *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.
- [6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proc. ICML*, 2017, pp. 1321–1330.
- [7] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman and Hall, 1993.