PAIRED-CONTRACT AUDITS FOR TEMPORALLY VALID TOOL-AGENT LEADERBOARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-agent leaderboards guide system selection, but a score measured against one interface snapshot need not preserve the decision after the interface changes. We introduce PACTBENCH (Paired Audit of Contract Transitions), an audit protocol that turns this temporal-validity question into a controlled comparison. PACTBENCH pairs contracts that preserve a task's goal, initial state, and validator; commits the later contract before execution; and reevaluates a predeclared policy panel under a shared observation boundary, backbone, and budget. Its rank-transition signature records score-band width, pairwise reversals, tie-aware rank agreement, and the winning set. The protocol makes the information boundary explicit: ordinary policies may interact with the revised tool only through the declared runtime channel, while the updated static contract and validator remain sealed. We give a task-blocked estimator, a reporting contract for external execution surfaces, and an audit record that separates public contract evidence from execution evidence. This design makes temporal rank preservation a concrete requirement for benchmark-supported tool-agent selection.

1 INTRODUCTION

Tool-use methods and agent benchmarks motivate systematic comparisons of interactive capabilities (Schick et al., 2023; Yao et al., 2023; Liu et al., 2024). ToolBench (Qin et al., 2024), API-Bank (Li et al., 2023), WebArena (Zhou et al., 2024), and AgentBench (Liu et al., 2024) assess broad capability against a particular tool or environment configuration. Such a score is useful for a decision made at that configuration. A distinct question arises when a public interface evolves: does the shortlist selected from the earlier contract remain the appropriate shortlist when the task goal and success criterion remain fixed but the contract changes?

Interface evolution changes client-facing field names, response shapes, authentication flows, pagination, and error surfaces. These changes interact with the adaptation channels of tool agents: a policy may exploit runtime errors, structured endpoint responses, documentation retrieval, or a learned transition prior. A static leaderboard can therefore compress policies whose behavior diverges once a contract transition is introduced. This is a decision-validity problem, not merely another robustness score.

PACTBENCH provides an audit object for that problem: a paired contract transition together with a fixed policy panel and a precisely declared information boundary. The protocol does not treat arbitrary task variation as a transition. It admits a pair only when the task goal, saved initial state, and success validator are invariant. It commits the revised specification before policy execution, specifies what a policy can observe after the transition, and evaluates every policy under the same backbone, compute budget, task pair, and randomization plan. The resulting rank-transition signature exposes score compression, pairwise decision changes, tie-aware rank agreement, and changes in the winning set.

The paper makes four contributions:

- It formulates temporal rank validity for tool-agent leaderboards as a paired-contract audit with invariant task semantics and an explicit rank-transition signature.
- It separates direct release of a revised static specification from admissible runtime observations, producing a testable information boundary for adaptive policies.

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.

- It defines a task-blocked controlled estimator and a reporting contract that prevents trajectory episodes, seeds, or external interface pairs from being treated as interchangeable independent observations.
- It supplies an audit record and worked contract pair that make admission, release, state reset, scoring, and rank comparisons inspectable.

2 RELATED WORK

Tool and web-agent suites evaluate agents across diverse tasks and interfaces, including ToolBench (Qin et al., 2024), API-Bank (Li et al., 2023), Gorilla (Patil et al., 2024), Toolformer (Schick et al., 2023), WebArena (Zhou et al., 2024), Mind2Web (Deng et al., 2023), AgentBench (Liu et al., 2024), AppWorld (Trivedi et al., 2024), AndroidWorld (Rawles et al., 2025), OSWorld (Xie et al., 2024), and τ-bench (Yao et al., 2025). Some environments vary documentation or task parameters dynamically. PACTBENCH instead evaluates whether one fixed policy decision remains valid across a semantics-preserving versioned contract pair.

Non-stationarity and continual-adaptation research studies changing rewards, distributions, tasks, and simulators (Padakandla, 2021; Gama et al., 2014; Parisi et al., 2019; Kirkpatrick et al., 2017; Zenke et al., 2017; Tobin et al., 2017; Peng et al., 2018). Software-evolution research establishes the practical importance of client-visible interface changes (Espinha et al., 2015). PACTBENCH connects these perspectives by holding the task objective and validator fixed while making the interface transition, its release channel, and its effect on a predeclared ranking the evaluation object.

3 THE PACTBENCH AUDIT

3.1 ADMISSIBLE PAIRED CONTRACTS

Let a task instance be $x = (g, w_0, V)$, where g is a goal, w_0 is a serializable initial world state, and V is a deterministic validator. Let σ^{pre} and σ^{post} be the earlier and later contracts. A pair is admissible when

$$(g, w_0, V)_{\sigma^{\mathrm{pre}}} = (g, w_0, V)_{\sigma^{\mathrm{post}}},$$

and the contract difference changes a client-visible request, response, authentication, pagination, or error interaction without changing the success postcondition. This definition excludes a revision that changes what counts as a successful task completion.

The protocol uses six contract-change families: parameter rename, field nesting, pagination shift, authentication refresh, error-code remapping, and partial deprecation. A family label records the most specific client-visible difference used to construct the pair. A multi-change pair retains all applicable labels in its admission record; a study must state whether its sampling design uses one primary label or a multi-label composition.

3.2 INFORMATION BOUNDARY AND EXECUTION STATE

The audit distinguishes a static revised contract from observations produced while a policy interacts with the tool. Before execution, the admission record declares a release event that separates the pre-transition episode from every post-transition episode, together with a common action, token, and wall-clock budget. Before that event, every policy receives the same declared pre-transition materials. At the event, the canonical byte sequence for σ^{post} is sealed with its hash and is unavailable to ordinary policies. During a post-transition episode, a policy may receive only the tool observations that the evaluator declares for that environment. The validator and the sealed static contract remain outside the policy process.

This boundary makes the source of adaptation inspectable. Static Prompt, documentation retrieval, schema-oriented parsing, retry, and learned transition policies may differ in how they use their allowed observations, but the evaluator must not grant a policy an undeclared channel. A clairvoyant oracle is a separately named condition that reads σ^{post} directly before its first post-transition action; it is never pooled with ordinary policies in a rank comparison.

Table 1: Information-boundary contract. Each row is recorded for every evaluated environment and policy panel before execution.

Channel	Required rule
Pre-transition materials	The same versioned contract materials are released to every policy before episode 0.
Post-transition static contract	The canonical revised contract is sealed before execution and is released directly only to the declared oracle condition.
Runtime tool observation	Every policy receives only the responses, errors, and endpoint-specific observations generated by actions it is permitted to issue under the shared budget.
Introspection	An environment may expose a named query endpoint only when its request and response schema are included in the pre-transition materials; it is not a substitute for a serialized revised contract.
Validator	The validator is not a policy input and records success and unsafe state-changing actions after an episode ends.
State and reset	Each episode starts from the saved paired initial state. If policy memory is enabled, it spans only the post-transition episodes for one task–seed trajectory and is reset at the next task–seed trajectory; environment state never carries between episodes.

3.3 Controlled estimator and rank-transition signature

For policy a, task pair $t \in \mathcal{T}$, seed $z \in \mathcal{Z}$, and episode e, let $Y_{a,t,z,e} \in \{0,1\}$ be the validator outcome. Episode $e = 0$ is evaluated against σ^{pre} and post-transition episodes are indexed by $e \in \{1, \dots, E\}$. A failed or timed-out execution has $Y_{a,t,z,e} = 0$ and retains its termination reason in the execution record. Define

$$\bar{Y}^{\mathrm{pre}}_{a,t,z} = Y_{a,t,z,0}, \qquad \bar{Y}^{\mathrm{post}}_{a,t,z} = \frac{1}{E}\sum_{e=1}^{E} Y_{a,t,z,e}, \tag{1}$$

$$\widehat{s}^{\,v}_a = \frac{1}{|\mathcal{T}||\mathcal{Z}|}\sum_{t\in\mathcal{T}}\sum_{z\in\mathcal{Z}} \bar{Y}^{v}_{a,t,z}, \qquad v \in \{\mathrm{pre}, \mathrm{post}\}. \tag{2}$$

The trajectory average is nested within the task–seed unit; it is not an additional independent sample. A paired contrast is

$$\widehat{d}^{\,v}_{a,b} = \frac{1}{|\mathcal{T}||\mathcal{Z}|}\sum_{t,z}\left(\bar{Y}^{v}_{a,t,z} - \bar{Y}^{v}_{b,t,z}\right).$$

Confidence intervals resample complete task blocks, retaining every policy, seed, and episode associated with a sampled task; seeds and episodes are never resampled as independent units. Randomization tests, when reported, permute paired policy labels only within those complete task blocks. The audit report states the number of task pairs, seeds, post-transition episodes, confidence level, bootstrap replicates, multiplicity adjustment, and termination-reason counts.

For a predeclared panel $\mathcal{P}$, scores induce a descending weak order R_v. Policies tie exactly when their aggregate scores are equal and receive their shared midrank. Let $\mathcal{U}$ be the policy pairs untied at both versions. When $\mathcal{U} = \varnothing$, the reversal rate is reported as Not Applicable rather than as a numeric zero. Otherwise, the reversal rate and signature are

$$q_{\mathrm{rev}} = \frac{1}{|\mathcal{U}|}\sum_{\{a,b\}\in\mathcal{U}} \mathbf{1}\left[\widehat{d}^{\,\mathrm{pre}}_{a,b}\widehat{d}^{\,\mathrm{post}}_{a,b} < 0\right], \qquad \Gamma(\mathcal{P}) = \left(W_{\mathrm{pre}}, W_{\mathrm{post}}, q_{\mathrm{rev}}, \tau_b, A^{\star}_{\mathrm{pre}}, A^{\star}_{\mathrm{post}}\right), \tag{3}$$

where $W_v = \max_{a\in\mathcal{P}} \widehat{s}^{\,v}_a - \min_{a\in\mathcal{P}} \widehat{s}^{\,v}_a$, $A^{\star}_v = \{a : \widehat{s}^{\,v}_a = \max_{b\in\mathcal{P}} \widehat{s}^{\,v}_b\}$, and τ_b is Kendall's tie-adjusted correlation between R_{pre} and R_{post}. The winning object is a set, so a tie is reported rather than resolved by an unstated tie-break. The report gives the complete ranking or tied ranking used in every τ_b calculation.

3.4 External execution surfaces

An external interface evaluation is a distinct estimand, not an additional row that can silently be pooled with the controlled estimator. Its record identifies the version pair, task instance, policy, seed, state-reset rule, successful executions, attempted executions, and the external observation channel. If the unit is an interface-version pair, uncertainty resamples version pairs as blocks. If a policy is evaluated repeatedly on one pair, those repetitions remain nested within that pair. A report comparing controlled and external rankings states both estimands, their matching rule, and whether the comparison is post-transition-only or a true paired pre/post contrast.

4 Audit record and reporting contract

Each evaluated row is accompanied by four linked records. The admission record preserves the two contracts, the task invariant, the validator version, the contract hashes, and the taxonomy labels. The execution record preserves policy and model identifiers, released channels, budget, seed, reset rule, and per-episode outcomes. The aggregation record gives denominators, the estimator, resampling blocks, and every correction used for multiple comparisons. The figure record maps each plotted point and interval to the corresponding aggregation output. These records make a reported table, rank transition, or trajectory traceable without conflating public contract evidence with execution evidence.

The protocol also requires a negative-control declaration before scoring. A control must specify which release channel or task–policy alignment it removes, what remains observable, and the same task-blocked estimator used for the main comparison. This separates contract identity checks, information withholding checks, and policy adaptation checks; none is a proxy for the others.

5 Discussion

PACTBench changes the question a tool-agent benchmark answers. Alongside measuring capability under a snapshot, it asks whether the ordering induced by that snapshot remains decision-relevant when a semantics-preserving public contract changes. The answer is auditable because the task invariant, sealed transition, information boundary, estimator, and ranking rule are all explicit.

The protocol also supports sharper experimental interpretation. A score width describes compression or separation, a reversal identifies a changed pairwise choice, Kendall's τ_b summarizes the full tied ordering, and the winning set records the decision consequence. Reporting these objects together prevents a small aggregate correlation from masking a decisive crossing, or a trajectory repetition from inflating the independent sample size.

6 Conclusion

Paired-contract auditing provides a principled way to evaluate temporal validity in tool-agent leaderboards. PACTBench couples invariant task pairing, a sealed revised contract, a declared runtime information boundary, task-blocked estimation, and a tie-aware rank-transition signature. The resulting audit makes it possible to determine whether a static benchmark decision survives the interface evolution that deployed tool agents must confront.

AI Use Statement

Generative AI was used only for manuscript writing, editing, and presentation. The authors reviewed all edits and remain responsible for the final content.

References

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://doi.org/10.52202/075280-1220.

Tiago Espinha, Andy Zaidman, and Hans-Gerhard Gross. Web API growing pains: Loosely coupled yet strongly tied. *Journal of Systems and Software*, 100:27–43, 2015. URL https://doi.org/10.1016/j.jss.2014.10.014.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014. URL https://doi.org/10.1145/2523813.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114 (13):3521–3526, 2017. URL https://doi.org/10.1073/pnas.1611835114.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116, 2023. URL https://aclanthology.org/2023.emnlp-main.187/.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 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.

Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys*, 54(6):1–25, 2021. URL https://doi.org/10.1145/3459991.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. URL https://doi.org/10.1016/j.neunet.2019.01.012.

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. URL https://doi.org/10.52202/079017-4020.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, pp. 3803–3810. IEEE, 2018. URL https://doi.org/10.1109/ICRA.2018.8460528.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 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.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/01a83bc2f2732a58e6aa731e659e7101-Abstract-Conference.html.

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. URL https://doi.org/10.52202/075280-2997.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30. IEEE, 2017. URL https://doi.org/10.1109/IROS.2017.8202133.

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, 2024. URL https://aclanthology.org/2024.acl-long.850/.

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*, 2024. URL https://doi.org/10.52202/079017-1650.

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://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R 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.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 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.

A WORKED CONTRACT PAIR

Consider a ticketing task whose goal is to create a high-priority ticket for a specified assignee. In the earlier contract, the request body is `{"issue_id": 42, "priority": "high", "assignee": "alice"}`. In the later contract, the same task postcondition is evaluated from the same initial state, while the request body is `{"ticket_id": 42, "metadata": {"priority": "high"}, "owner": "alice"}`. The pair combines an identifier rename, field nesting, and an assignee-field rename. The validator checks the created ticket rather than the request syntax, so the task goal and success criterion are unchanged.

The admission record for this pair stores both canonical contract bytes, their hashes, the initial-state identifier, the validator identifier, the field-level diff, and the declared runtime error and introspection channels. An ordinary policy can act and observe only through the boundary in Table 1; the oracle condition receives the later contract directly.

B MINIMUM REPORT FOR A RANK TRANSITION

A rank-transition result reports the complete policy panel, all ties, the pre- and post-transition estimates, each pairwise contrast used to identify a reversal, and the task-blocked uncertainty procedure. It also reports the release matrix from Table 1, the task-pair admission criteria, and the policy-memory/reset convention. These fields allow a reviewer to determine whether a claimed rank change is attributable to the contract transition rather than to a semantic task change, an undeclared observation channel, model-family confounding, or an invalid independent unit.