PROBING TASK-LEVEL AGENT RELIABILITY THROUGH ROLLOUT AND REPLAY

Anonymous authors
Paper under double-blind review

Anonymous code: [https://anonymous.4open.science/r/agent-reliability-3353]

ABSTRACT

An agent's execution record can be available before an external evaluator returns a task outcome. We ask whether the acting model's own unlabeled reasoning-and-action traces reveal how reliably it handles a fixed task. Our rollout–replay protocol collects repeated natural executions, replays their recorded histories through the original model, and reads intrinsic statistics at action and post-observation events. Discovery batch A fixes the measurement rules before evaluation on new execution seeds in batch B. Across benchmarks, event-aligned readouts rank task reliability more accurately than matched whole-trajectory statistics. Level separates groups whose sampled executions consistently succeed or fail, while Dispersion distinguishes higher- from lower-success groups with mixed outcomes. Scores from A also rank outcomes from independent B executions; comparator gains persist before explicit outcome disclosure and on held-out MCPMark. Cross-model replay shows that reader capability alone does not replace compatibility with the acting model. These results support task-level reliability measurement from unlabeled interaction traces. They do not establish calibrated success probabilities or a benefit from using the scores to change agent behavior.

"Nature uses only the longest threads to weave her patterns,
so each small piece of her fabric reveals the organization of the entire tapestry."

— Richard Feynman

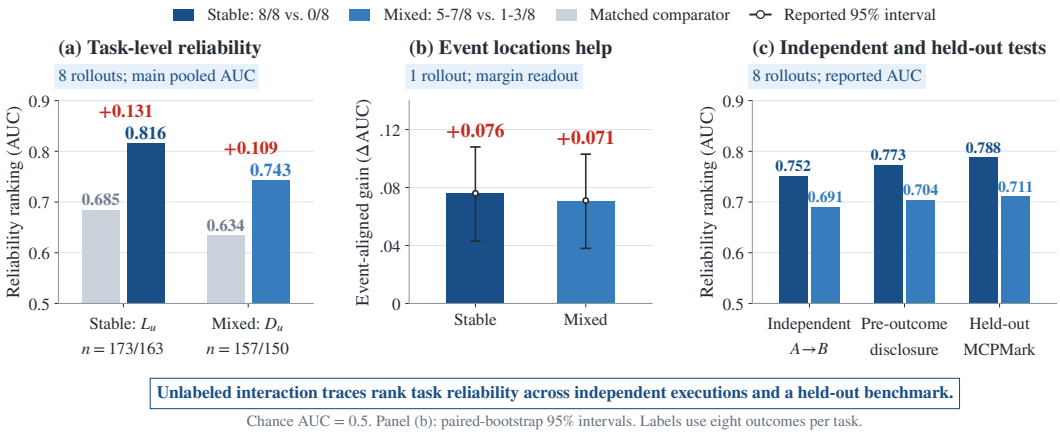

Figure 1: Task-level reliability from unlabeled interaction traces. (a) At eight scoring rollouts, Level L_u ranks 8/8-success versus 0/8-success groups and Dispersion D_u ranks 5–7/8 versus 1–3/8 groups; gray bars show the matched whole-trajectory comparator and n lists positive/negative task counts. (b) At one scoring rollout, event-aligned margin improves on whole-trajectory margin; whiskers show paired-bootstrap 95% confidence intervals. (c) Ranking persists for independent A-to-B executions, pre-disclosure readouts, and held-out MCPMark. Bars show Stable L_u and Mixed D_u readouts (comparators omitted for compactness; see Table 2). Labels always use eight outcomes; the scores are not calibrated success probabilities.

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.

1 INTRODUCTION

Reasoning systems increasingly spend computation on more than one proposed answer. Self-consistency samples reasoning chains and aggregates their conclusions (Wang et al., 2023); trained verifiers select among candidate solutions (Cobbe et al., 2021); process supervision evaluates intermediate steps (Lightman et al., 2024). Tool-using agents add actions, observations, and changing environment state to that execution record (Yao et al., 2023; Schick et al., 2023). An agent may submit a patch before the tests finish or stop after tool calls whose effects require a separate check. Its trace then records how it acted, but the official outcome may still be unavailable. Can the acting model's own record already reveal which tasks it handles more reliably?

This question differs from estimating whether one answer is correct. Models can report confidence, while semantic uncertainty measures variation among answers grouped by meaning (Kadavath et al., 2022; Kuhn et al., 2023; Farquhar et al., 2024). Learned verifiers inspect reasoning with correctness supervision and additional training. Agents can also revise later attempts using feedback (Shinn et al., 2023). Before requesting feedback or retaining experience, a system may need to compare a fixed agent configuration's reliability across tasks. Repeated-run evaluation exposes reliability gaps (Yao et al., 2025); we ask whether unlabeled traces anticipate these outcomes without training a predictor or changing the execution policy.

A fixed agent–task tuple can reach different states across executions, while token positions need not identify comparable interactions. We therefore read intrinsic signals at action commitments and post-observation updates, then aggregate rollout summaries across independently reset executions. Level L_u describes typical readout value; Dispersion D_u describes execution-to-execution variation. We evaluate them separately.

Natural rollouts exclude official outcomes from measurement. Teacher-forced replay reads the original model's distributions at fixed model-visible prefixes, without replacing actions or collecting observations. Discovery batch A freezes measurement choices before B evaluation; a separate A-to-B analysis ranks fresh outcomes of the same units.

Figure 1 summarizes the evidence. Eight-rollout Level and Dispersion reach pooled AUC 0.816/0.743 on stable/mixed groups, compared with 0.685/0.634 for the matched whole-trajectory comparator. Event alignment also improves one-rollout ranking, and the association persists across execution batches, pre-disclosure restrictions, and held-out MCPMark. Stable and mixed comparisons use different observed groups, rather than identical labels scored by alternative aggregations.

We contribute an event-aligned, fixed-replay protocol and evidence that task-level rankings persist beyond recorded attempts. Budget and model comparisons show diminishing sampling returns, an advantage for self replay, and ranking gains alongside task performance. These findings support actor-specific measurement; causal mechanisms and downstream value require separate evaluation, including matched-budget compute-allocation and experience-retention studies.

2 RELATED WORK

Answer confidence and repeated sampling. Language models can expose answer-correctness information through elicited confidence (Kadavath et al., 2022; Lin et al., 2022; Tian et al., 2023). Semantic uncertainty groups sampled answers by meaning rather than treating lexical variation as disagreement (Kuhn et al., 2023); semantic entropy extends this approach to detecting confabulations (Farquhar et al., 2024). LM-Polygraph organizes token- and sequence-level uncertainty estimators (Fadeeva et al., 2023). These studies motivate intrinsic readouts, but an agent's reliability concerns the outcome of an interaction with a changing environment. Two traces may use different actions and still succeed, or agree on an unsuccessful plan. Dispersion of event readouts consequently measures variation in execution statistics; it is not semantic entropy or a vote over final answers.

Self-consistency aggregates reasoning paths (Wang et al., 2023); outcome verifiers and process supervision assess candidate solutions (Cobbe et al., 2021; Lightman et al., 2024). Adaptive test-time computation allocates effort by problem (Snell et al., 2025). Our budget measures a fixed agent–task tuple through independent natural executions, without selecting an answer or training a verifier. Improved allocation requires separate validation, as does probability calibration (Guo et al., 2017).

Model-internal knowledge and readout specificity. Contrast-consistent search extracts candidate truth directions from hidden representations without correctness labels (Burns et al., 2023). Subsequent analysis shows that consistency objectives can also select features unrelated to truth, making the interpretation of a discovered direction an identification problem (Farquhar et al., 2023). We use token-distribution statistics from fixed replay and evaluate them against observed execution outcomes. Cross-model replay asks whether a reader transfers to another acting model's traces; this distinguishes model-specific ranking information from a claim that a readout directly identifies the model's knowledge or internal failure mechanism.

Interactive agents and reliability evaluation. ReAct interleaves reasoning with actions and observations (Yao et al., 2023); Toolformer learns when and how to call external tools (Schick et al., 2023). Interaction boundaries thus provide useful measurement locations, although those methods do not establish the value of event-aligned reliability readouts. AgentBench evaluates agents across environments (Liu et al., 2024), WebArena assesses functional web-task completion (Zhou et al., 2024), and SWE-bench evaluates repository changes against software tests (Jimenez et al., 2024). The benchmarks in our evaluation cover function calling, multi-app state changes, and interactive science tasks (Patil et al., 2025; Trivedi et al., 2024; Wang et al., 2022), with MCPMark providing a held-out tool-use setting (Wu et al., 2025). τ-bench explicitly examines repeated-run reliability in tool–agent–user interaction (Yao et al., 2025). We complement outcome-based evaluation by asking whether unlabeled traces can rank agent–task units before their official outcomes enter the analysis.

Learning from agent experience. Reflexion stores verbal feedback (Shinn et al., 2023), Voyager accumulates executable skills (Wang et al., 2024), and ExpeL extracts reusable lessons from task experience (Zhao et al., 2024). Self-Rewarding Language Models use model-generated rewards during training (Yuan et al., 2024); the Darwin Gödel Machine searches over agent-code modifications (Zhang et al., 2025). A survey organizes adaptation by the components, timing, and feedback involved (Gao et al., 2026). These systems make experience selection a practical downstream question. Our protocol isolates the measurement that might inform such a decision; it neither updates agent state nor measures gains from retaining experience.

3 Problem Setup

Task-specific reliability. Let x denote a task instance with its prescribed initial environment state, m the acting model revision, and c an agent configuration specifying the prompt, scaffold, tools, decoding rule, context policy, and execution budget. The unit of analysis is $u = (m, c, x)$. An external evaluator maps a completed execution τ to an outcome $r(\tau)$. The population reliability of this fixed agent–task tuple is

$$P_u = \mathbb{E}_{\tau \sim \pi_{m,c}(\cdot|x)}[r(\tau)]. \tag{1}$$

Our question is whether the agent's own unlabeled interaction trajectories contain information that ranks these task-specific reliabilities. The representation we construct is neither a calibrated estimate of P_u nor an outcome predictor trained on evaluator labels.

Repeated natural executions. For each u, we independently reset the task and collect K ordinary executions,

$$\mathcal{T}_u = \{\tau_1, \ldots, \tau_K\}, \qquad \tau_i \sim \pi_{m,c}(\cdot \mid x). \tag{2}$$

The agent acts under its original policy, without confidence elicitation or behavior modification for measurement. Each trace records generated tokens, actions, observations, and the model-visible context at each generation point. Evaluator outputs and reference answers are held outside the measurement trace. If the environment itself discloses an outcome to the agent, we mark the first disclosure so that earlier events can be analyzed separately.

Interaction coordinates and task representation. An agent rollout alternates variable-length generation with actions and observations. Absolute token positions need not correspond across executions, whereas action commitments and post-observation updates identify comparable interaction locations. We use these events to index endogenous readouts from fixed-trajectory replay. For a chosen readout and within-rollout reduction, each τ_i yields a summary s_i. Across executions, the

task-level representation is

$$R_u = (L_u, D_u), \qquad L_u = \frac{1}{K}\sum_{i=1}^{K} s_i, \qquad D_u = \mathrm{Dispersion}(\{s_1, \ldots, s_K\}). \tag{3}$$

L_u describes the typical level of the intrinsic readout; D_u describes its variation across independent executions. The choice of readout, event mask, within-rollout reduction, dispersion functional, and directions is determined on discovery data and frozen before evaluation. We examine L_u and D_u separately; a combined scalar is not part of the core method.

Outcomes used only for evaluation. For a collected batch, its observed group outcome is

$$\widehat{P}_u = \frac{1}{K}\sum_{i=1}^{K} r(\tau_i). \tag{4}$$

We compute the intrinsic representation before joining it to official outcomes. A score and an observed outcome computed from the same execution batch are not statistically independent, even though the outcome labels never enter score construction. An independent-batch analysis can instead compare measurements from one batch with outcomes of fresh executions of the same u; that analysis is separate from the method illustration.

4 METHOD

Figure 2 separates event selection from aggregation: natural rollouts supply histories (a), replay reads interaction events (b), and rollout summaries form Level L_u and Dispersion D_u (c). Discovery batch A freezes the rules before B outcome evaluation (d).

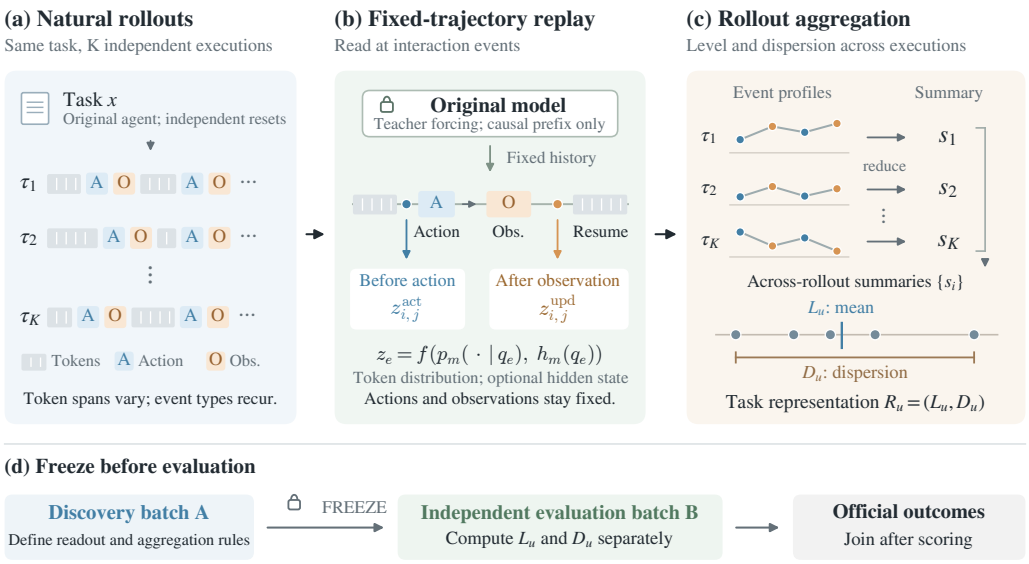

Figure 2: Event-aligned rollout–replay protocol. (a) Independent natural executions share event types despite varying token lengths. (b) Fixed replay reads original model-visible prefixes before actions and after observations. (c) Rollout summaries form Level L_u and Dispersion D_u, evaluated separately. (d) Batch A freezes measurement rules; official B outcomes enter only after scoring.

4.1 WHERE TO READ? INTERACTION EVENTS

Natural rollouts. For each fixed $u = (m, c, x)$, the original policy generates $\mathcal{T}_u$ (Equation 2). We preserve model-visible histories at every generation point and exclude official evaluator outputs.

We align action and observation event types rather than token indices; the jth action need not share content or purpose across rollouts (Figure 2a).

Fixed replay at event boundaries. Let $e^{\mathrm{act}}_{i,j}$ denote the boundary immediately before rollout τ_i's jth recorded action, and $e^{\mathrm{upd}}_{i,j}$ the point where generation resumes after the associated observation, when a continuation exists. These locations read the model before an action commitment and after new environment information enters its context.

Teacher-forced replay through the original model revision fixes the recorded tokens, actions, observations, and context policy. It neither resamples actions nor invokes tools. A readout uses only the prefix visible to the original agent at that event. This requires preserving the tokenizer, message template, token boundaries, and any history truncation or compression; invalid reconstructions and unavailable readouts count as missing events (Figure 2b).

Endogenous event readouts. For an eligible event e with causal prefix q_e, the model supplies a next-token distribution $p_m(\cdot \mid q_e)$ and, where access permits, a hidden-state readout $h_m(q_e)$. An event measurement is

$$z^q_{i,j} = f\Big(p_m(\cdot \mid q_{e^q_{i,j}}), h_m(q_{e^q_{i,j}})\Big), \qquad q \in \{\mathrm{act}, \mathrm{upd}\}. \tag{5}$$

The reported coordinate ablation examines log probability, entropy, margin, and token rank, standard ingredients in language-model uncertainty estimation (Fadeeva et al., 2023). A chosen-token measurement scores the recorded token, without sampling a replacement. Each readout definition fixes its event and token masks, probability convention, within-event reduction, and missingness rule. The protocol also allows hidden-state measurements, but the reported coordinate comparisons evaluate token-distribution statistics.

4.2 How to aggregate? Repeated executions

From event profiles to rollout summaries. A fixed reduction of eligible $z^q_{i,j}$ values yields one summary s_i for each rollout. Across executions, $\mathcal{S}_u = \{s_1, \ldots, s_K\}$ forms an empirical distribution of summaries, rather than token-aligned observations. Changing the readout, event mask, or reduction defines a different candidate measurement.

Level and dispersion. Equation 3 defines mean Level L_u and across-rollout Dispersion D_u (Figure 2c), which we evaluate separately. The dispersion functional and ranking direction belong to the measurement definition: greater variation need not imply lower reliability. The aggregation ablation compares standard deviation with other distribution summaries.

Freeze before evaluation. Discovery batch A fixes the readout, masks, reductions, aggregation functionals, missingness rules, and score directions. The frozen procedure computes L_u and D_u from unlabeled B rollouts before evaluation joins them to official outcomes (Figure 2d). This tests ranking under new execution seeds. We separately compare A measurements against outcomes from fresh B executions to address the shared-batch dependence described in Section 3.

5 Results

Experimental setup. We evaluate eight backbones: Gemma 3 IT (1B, 4B, 12B, 27B) (Gemma Team, 2025) and Qwen3.5 (2B, 4B, 9B, 27B) (Qwen Team, 2026) (Table 3). Unless otherwise stated, main experiments use Qwen3.5-27B as both the acting and replay model. Natural rollouts use sampling temperature $T = 1.5$ to encourage diversity; other inference settings follow each backbone's official configuration. Labels use $K_{\mathrm{label}} = 8$ outcomes: stable-task positives/negatives have 8/8 and 0/8 successes; mixed-task positives/negatives have 5–7/8 and 1–3/8. Groups with 4/8 are excluded; these labels do not identify population reliabilities. Scores use $K_{\mathrm{score}} \in \{1, 2, 4, 8\}$ unlabeled executions. Batch B contributes 173/163 positive/negative stable groups and 157/150 mixed groups across BFCL (Patil et al., 2025), AppWorld (Trivedi et al., 2024), ScienceWorld (Wang et al., 2022), and OdysseyArena (Odyssey) (Yan et al., 2026); held-out MCPMark (Wu et al., 2025) contributes 42/38 and 35/36, respectively. Pooled AUCs combine task groups across benchmarks. Discovery batch A on the main benchmarks fixes measurement choices before B evaluation. MCPMark is used only for held-out evaluation, excluded from readout, aggregation, and direction selection. Appendix A, Table 4, details support and exclusions.

5.1 INTERACTION EVENTS CONCENTRATE RANKING INFORMATION

Reading at interaction events improves ranking even when the rollout budget stays fixed (Table 1). The single-rollout margin gains are +0.076 stable AUC (95% CI [0.043, 0.108]) and +0.071 mixed AUC ([0.038, 0.103]); all four intrinsic readouts benefit from event alignment (Table 6). The gains across readouts suggest that measurement location matters beyond the statistic chosen. Whole-trajectory averaging may dilute interaction-boundary information by mixing action-relevant uncertainty with routine text generation.

Action boundaries favor stable ranking, whereas observation updates favor mixed ranking; combining them performs best in both regimes. The pattern fits two complementary sources of information: committing to an action and responding to new environment evidence. The ablation establishes predictive value, however, rather than proving that either event type captures a specific internal computation.

Table 1: Task-level AUC on batch B. The single-rollout rows compare margin readout coordinates. The matched comparator applies the same 8-rollout aggregation (L_u for stable, D_u for mixed) to whole-trajectory margin.

Score	Scoring budget	Stable AUC	Mixed AUC
Model-call count	8	0.611	0.602
Whole-trajectory margin	1	0.623	0.571
All-event margin	1	0.699	0.642
Matched whole-trajectory comparator	8	0.685	0.634
Event-aligned L_u / D_u	8	**0.816**	**0.743**

Level and dispersion answer different ranking questions. At eight executions, the event-aligned scores exceed the matched whole-trajectory comparator by 0.131 stable AUC and 0.109 mixed AUC; paired intervals exclude zero ([0.096, 0.164] and [0.071, 0.145]). The aggregation ablation explains why a single confidence summary is insufficient here. Mean outperforms standard deviation on stable groups (0.816 versus 0.681), but the ordering reverses on mixed groups (0.696 versus 0.743; Table 7). Level separates consistently successful from consistently unsuccessful observed executions, while dispersion helps distinguish groups whose outcomes vary. Treating variation as uniformly bad would miss this second use: both mixed classes contain failures, and the ranking direction must be established empirically. Different targets and task support preclude a paired comparison of these AUCs.

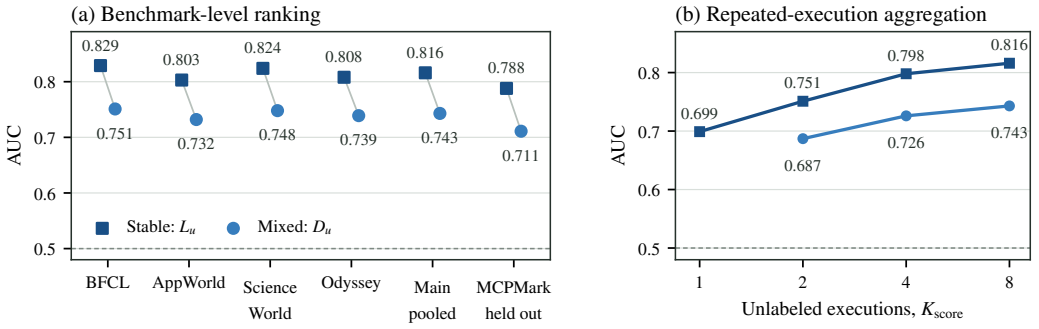

Figure 3: Ranking across benchmarks and scoring budgets. (a) Benchmark-level and pooled AUCs; MCPMark is held out. (b) AUC by the number of unlabeled executions used to construct the score. Stable tasks use L_u, mixed tasks use D_u, and the dashed lines mark chance. Dispersion requires at least two executions. Labels use eight outcomes at every scoring budget.

Repeated executions give diminishing ranking returns. Four executions recover 72%/70% of the stable/mixed improvement from two to eight, at half the tabulated replay cost of eight (Figure 3b;

Table 8). The smaller increments after four executions suggest diminishing returns to estimating the distribution of rollout summaries. Eight still improves on two by 0.065/0.056 AUC, with paired intervals [0.036, 0.094]/[0.027, 0.084]. This is a measurement-efficiency result: extra executions improve the ranking of a fixed policy, while adaptive test-time methods use computation to improve its answers (Snell et al., 2025). The curve does not determine whether scoring, retrying, or requesting an evaluator is the best use of another execution's cost.

5.2 Ranking persists beyond the trajectories used for scoring

The benchmark-level results reduce the possibility that pooled ranking comes only from differences between environments. Each main benchmark shows positive discrimination in both regimes (Figure 3a). MCPMark further tests measurement choices on a held-out tool-use setting: its gains over the matched comparator remain 0.127/0.094 AUC (Table 2). Absolute AUC is lower than in the main pool, suggesting that ranking transfers more readily than its precise ranking quality. Benchmark coverage establishes breadth across these settings, rather than unrestricted transfer to new agents or tools.

The independent A-to-B comparison asks a stricter question. Repeated-run reliability evaluation, as in τ-bench (Yao et al., 2025), concerns behavior over fresh attempts; reproducing a score–outcome association within one batch does not by itself address that target. A scores rank fresh B outcomes with AUC 0.752/0.691, and the setting-demeaned correlation remains 0.61 (95% CI [0.55, 0.67]). Persistence within each main benchmark (Table 11) supports information about recurring task-level behavior beyond the recorded attempts. Yet the 0.064/0.052 AUC loss relative to same-batch scoring shows that some of the association depends on the realized executions. A-to-B therefore provides the more relevant estimate for anticipating another batch of the same agent–task units; it still does not test unseen tasks, new configurations, or calibrated success probabilities.

5.3 The signal does not require an explicitly disclosed outcome

Pre-disclosure scoring retains an advantage of 0.111/0.083 AUC over its corresponding comparator, despite losing some absolute ranking quality (Table 2). Explicitly reading the returned result therefore cannot fully explain the main signal. Earlier observations can still expose partial progress or failed actions, which are legitimate model-visible information but prevent interpreting this control as a removal of all environmental evidence.

Clean-execution and length-matching restrictions address a different alternative: that the score mainly tracks broken tools, truncation, or the amount of recorded text. Positive gains persist after these restrictions (Table 15), supporting information beyond those coarse proxies. Matching can also change task support and comparator quality; a larger restricted ΔAUC consequently need not mean better absolute ranking. These controls narrow simple explanations rather than isolating a unique mechanism.

Table 2: AUC and gain over the matched whole-trajectory comparator. Each row uses its corresponding evaluation restriction.

	Stable			Mixed		
Setting	Ours	Comp.	Δ	Ours	Comp.	Δ
Main	0.816	0.685	+0.131	0.743	0.634	+0.109
Pre-outcome disclosure	0.773	0.662	+0.111	0.704	0.621	+0.083
Clean executions	0.808	0.681	+0.127	0.732	0.631	+0.101
Held-out MCPMark	0.788	0.661	+0.127	0.711	0.617	+0.094

Paired intervals also support the held-out MCPMark gains ([0.074, 0.177] stable; [0.041, 0.145] mixed; Table 16). The available results give point-estimate consistency, rather than separate significance claims, for restrictions without paired intervals.

5.4 A MORE CAPABLE READER DOES NOT REPLACE A MATCHED READER

Cross-model replay keeps a recorded trajectory fixed and changes the model that reads it (Figure 4). Self replay has the highest AUC point estimate in every column, in both families and regimes. On gemma-3-1b-it traces, for example, gemma-3-27b-it reaches stable AUC 0.621 versus self replay's 0.704. Moving upward in the reader scale ordering thus does not uniformly improve another model's ranking. With actions and observations fixed within each column, this tests readout compatibility rather than how readers would act.

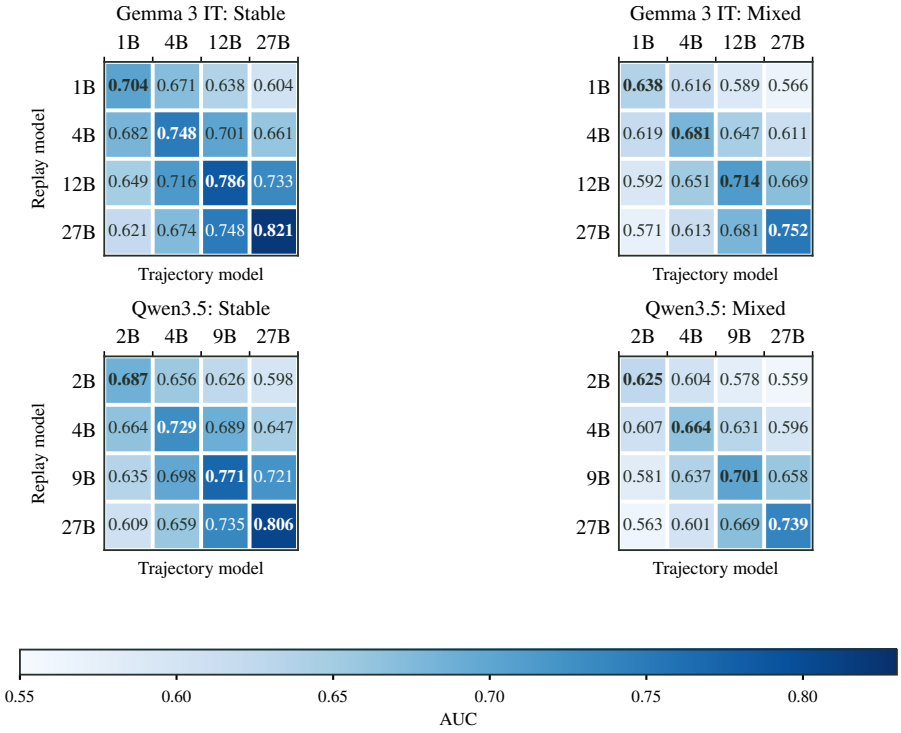

Figure 4: Cross-model replay for Gemma 3 IT and Qwen3.5. Axis labels give parameter sizes within each family. Columns denote trajectory models and rows denote replay models; diagonal cells are self replay. Darker blue indicates higher AUC.

Self replay exceeds all cross-model pairs by 0.088/0.076 AUC on average (95% CIs [0.064, 0.112]/[0.053, 0.099]); the gap is smaller for adjacent than distant models (Table 14). This graded loss supports readout dependence on the actor–reader pairing. Token-distribution shift, representation differences, and replay compatibility remain possible explanations, while above-chance cross replay indicates partial transfer.

Capability and observability move together, but are separable. Task success and self-replay AUC rise with scale within each family (Table 3), but the replay matrices show that the best actor's reader need not best rank another actor. Success rates count completed tasks; AUCs rank selected outcome groups. Actor changes can alter trajectories and group membership, preventing attribution to parameter scale alone. Transferred scores require validation for each actor–reader pair, beyond reader task performance.

Table 3: Task performance and reliability prediction across model scales.

Checkpoint	Success Rate	Stable L_u AUC	Mixed D_u AUC
gemma-3-1b-it	0.41	0.704	0.638
gemma-3-4b-it	0.49	0.748	0.681
gemma-3-12b-it	0.57	0.786	0.714
gemma-3-27b-it	0.64	0.821	0.752
Qwen3.5-2B	0.39	0.687	0.625
Qwen3.5-4B	0.47	0.729	0.664
Qwen3.5-9B	0.55	0.771	0.701
Qwen3.5-27B	0.62	0.806	0.739

6 Discussion

A measurement of recurring behavior. The experiments support a combined interpretation: event selection improves the relevance of each reading, repeated executions estimate its distribution, and matching the reader to the actor preserves more of the ranking signal. Independent-batch persistence shows that the representation describes behavior beyond its sampled traces. These results support task-level observability under the tested configurations; predictive association alone cannot identify a truth direction or internal failure detector (Farquhar et al., 2023).

The observed groups bound the conclusion. Eight outcomes define each label. An 8/8 group can fail on its next attempt, and a 0/8 group can succeed; neither identifies a population probability. Excluding 4/8 groups leaves the center of the reliability range untested. The strongest transfer analysis uses fresh executions of the same units, not unseen tasks or configurations. Future evaluation should retain intermediate groups and distinguish task transfer from execution-seed transfer. Probability calibration requires its own evaluation (Guo et al., 2017), including how an operating threshold behaves as the task mix changes.

A ranking does not yet specify a decision rule. Retries, external evaluation, and abstention have different costs. Test-time allocation studies assess how computation changes solution quality (Snell et al., 2025); our budget curve measures how computation changes ranking quality. Connecting them requires an intervention that charges for score construction and compares downstream task success at fixed total cost. The same constraint applies to experience retention: a high-ranked trace may be dependable but offer little new information, whereas an unsuccessful trace may contain a useful corrective lesson. Ranking reliability and selecting useful experience are separate objectives.

7 Conclusion

Event-aligned replay extracts task-level ranking information from unlabeled agent executions. Level and Dispersion describe different outcome regimes, and independent-batch persistence shows that their signal extends beyond the attempts used for scoring. Outcome-visibility controls narrow simple explanations; cross-model replay shows that reader capability alone does not replace compatibility with the acting model. These results support reliability measurement under the tested configurations. Extending it to calibrated probabilities, unseen tasks, or improved compute and experience-selection decisions requires separate evaluation.

Ethics Statement

Generative AI tools were used only for language polishing and manuscript clarity, not for research ideas, experiments, result analysis, or scientific conclusions. The authors reviewed and take full responsibility for all content.

REFERENCES

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. LM-Polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 446–461, 2023. URL https://aclanthology.org/2023.emnlp-demo.41/.

Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. Challenges with unsupervised LLM knowledge discovery. *arXiv preprint arXiv:2312.10029*, 2023. URL https://arxiv.org/abs/2312.10029.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=CTr3bovS5F.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06770.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2302.09664.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=8s8K2UZGTZ.

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://arxiv.org/abs/2308.03688.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *International Conference on Machine Learning*, pp. 48371–48392, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Qwen Team. Qwen3.5: Towards native multimodal agents. Official model release, 2026. URL https://qwen.ai/blog?id=qwen3.5.

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*, 2023. URL https://proceedings.neurips.cc/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, 2023. URL https://aclanthology.org/2023.emnlp-main.330/.

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/.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2305.16291.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côt'e, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022. URL https://aclanthology.org/2022.emnlp-main.775/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Zijian Wu, Xiangyan Liu, Xinyuan Zhang, Lingjun Chen, Fanqing Meng, Lingxiao Du, Yiran Zhao, Fanshi Zhang, Yaoqi Ye, Jiawei Wang, Zirui Wang, Jinjie Ni, Yufan Yang, Arvin Xu, and Michael Qizhe Shieh. MCPMark: A benchmark for stress-testing realistic and comprehensive MCP use. *arXiv preprint arXiv:2509.24002*, 2025. URL https://arxiv.org/abs/2509.24002.

Hang Yan, Fangzhi Xu, Qiushi Sun, Jinyang Wu, Zixian Huang, Muye Huang, Jingyang Gong, Zichen Ding, Kanzhi Cheng, Yian Wang, Xinyu Che, Zeyi Sun, Jian Zhang, Zhangyue Yin, Haoran Luo, Ben Kao, and Qika Lin. OdysseyArena: Benchmarking large language models for long-horizon, active and inductive interactions. *arXiv preprint arXiv:2602.05843*, 2026. URL https://arxiv.org/abs/2602.05843.

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://openreview.net/forum?id=roNSXZpUDN.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/yuan24d.html.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Tjarko Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025. URL https://arxiv.org/abs/2505.22954.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

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://arxiv.org/abs/2307.13854.

A Additional Results

This appendix reports per-benchmark support, the full readout and aggregation ablations, cross-model replay matrices, and robustness comparisons. The paired-effect figure summarizes the reported differences; the tables retain the exact reported values. The replay visualization and model-scale summary appear in the main text as Figure 4 and Table 3.

A.1 Per-benchmark task-group support

Table 4 reports the complete positive, negative, and excluded task-group counts for each benchmark and execution batch.

Table 4: Task-group support across benchmarks and independent execution batches. Stable n_+/n_- are the all-success/all-failure counts; mixed n_+/n_- are the mixed-high/mixed-low counts. Other groups have 4/8 successes. MCPMark counts are included for completeness; neither batch is used for readout, aggregation, or direction selection.

Batch	Benchmark	Complete $K = 8$	All success	Mixed high	Mixed low	All failure	Other / excluded
A	BFCL	180	49	39	37	43	12
A	AppWorld	176	41	43	40	41	11
A	ScienceWorld	172	52	35	34	41	10
A	Odyssey	168	37	41	40	40	10
A	MCPMark	164	44	36	36	37	11
B	BFCL	178	46	40	38	42	12
B	AppWorld	174	42	41	39	41	11
B	ScienceWorld	170	49	36	34	40	11
B	Odyssey	166	36	40	39	40	11
B	MCPMark	162	42	35	36	38	11

A.2 Benchmark-level ranking results

These are the benchmark-level AUCs summarized in Figure 3.

Table 5: Task-level reliability ranking across benchmarks. Stable-task results use Level L_u; mixed-task results use Dispersion D_u.

Benchmark	Stable L_u AUC	Mixed D_u AUC
BFCL	0.829	0.751
AppWorld	0.803	0.732
ScienceWorld	0.824	0.748
Odyssey	0.808	0.739
Main pooled	**0.816**	**0.743**
MCPMark (held out)	0.788	0.711

A.3 READOUT AND EXECUTION ABLATIONS

These tables compare event coordinates and intrinsic measurements, task-level aggregation statistics, and the number of executions used to construct the score. The full method comparison gives the individual AUCs behind the concise summary in Table 1.

Table 6: Single-rollout readout-coordinate ablation ($K_{\text{score}} = 1$, $K_{\text{label}} = 8$). For each intrinsic measurement, Δ is measured relative to the corresponding whole-trajectory readout.

Readout	Coordinate	Stable AUC	Δ	Mixed AUC	Δ
Log probability	Whole trajectory	0.612	–	0.558	–
	Action boundary	0.661	+0.049	0.594	+0.036
	Observation update	0.653	+0.041	0.608	+0.050
	All interaction events	**0.681**	**+0.069**	**0.619**	**+0.061**
Entropy	Whole trajectory	0.601	–	0.551	–
	Action boundary	0.649	+0.048	0.589	+0.038
	Observation update	0.643	+0.042	0.602	+0.051
	All interaction events	**0.671**	**+0.070**	**0.614**	**+0.063**
Margin	Whole trajectory	0.623	–	0.571	–
	Action boundary	0.681	+0.058	0.611	+0.040
	Observation update	0.674	+0.051	0.624	+0.053
	All interaction events	**0.699**	**+0.076**	**0.642**	**+0.071**
Token rank	Whole trajectory	0.606	–	0.554	–
	Action boundary	0.655	+0.049	0.592	+0.038
	Observation update	0.648	+0.042	0.606	+0.052
	All interaction events	**0.676**	**+0.070**	**0.617**	**+0.063**

A.4 AGGREGATION AND SCORING BUDGET

The two targets favor different aggregation statistics; replay cost grows with the number of scored executions.

Table 7: Comparison of task-level aggregation statistics over $K = 8$ event-aligned rollout readouts.

Statistic over $\{s_i\}_{i=1}^{8}$	Stable AUC	Mixed AUC
Mean / Level L_u	**0.816**	0.696
Median	0.782	0.681
Standard deviation / Dispersion D_u	0.681	**0.743**
Interquartile range	0.668	0.721
Range	0.651	0.704

Table 8: Effect of the number of executions used to construct the task-level reliability representation. Relative cost counts replayed executions; it does not measure wall-clock time. Stable-task results use Level L_u; mixed-task results use Dispersion D_u.

K_{score}	Stable L_u AUC	Δ vs. $K = 1$	Mixed D_u AUC	Δ vs. $K = 2$	Relative replay cost
1	0.699	–	N/A	N/A	1×
2	0.751	+0.052	0.687	–	2×
4	0.798	+0.099	0.726	+0.039	4×
8	**0.816**	**+0.117**	**0.743**	**+0.056**	8×

Table 9: Main results. Single-rollout experiments evaluate the readout coordinate of intrinsic signals, while repeated-rollout experiments evaluate task-level reliability statistics across executions.

Method	K_{score}	Stable AUC	Mixed AUC
Execution statistics			
Number of model calls	8	0.611	0.602
Generated token count	8	0.594	0.587
Trajectory length	8	0.603	0.591
Single-rollout intrinsic measurements			
Whole-trajectory log probability	1	0.612	0.558
Whole-trajectory entropy	1	0.601	0.551
Whole-trajectory token rank	1	0.606	0.554
Whole-trajectory margin	1	0.623	0.571
Interaction-aligned margin			
Action-boundary margin	1	0.681	0.611
Observation-update margin	1	0.674	0.624
All-event margin	1	<u>0.699</u>	<u>0.642</u>
Repeated-rollout statistics			
Level L_u	8	**0.816**	0.696
Dispersion D_u	8	0.681	**0.743**

A.5 INDEPENDENT-BATCH DETAILS

The benchmark-level estimates complement the pooled A-to-B result below.

Table 10: Independent-batch persistence. Spearman correlations are setting-demeaned.

Evaluation	Stable AUC	Mixed AUC	Setting-demeaned Spearman ρ	95% CI
$(L_u^B, D_u^B) \leftrightarrow \widehat{P}_u^B$	0.816	0.743	0.68	[0.63,0.73]
$(L_u^A, D_u^A) \rightarrow \widehat{P}_u^B$	0.752	0.691	0.61	[0.55,0.67]

Table 11: Independent A→B persistence across the main benchmarks.

Benchmark	Stable L_u AUC	Mixed D_u AUC	Spearman ρ
BFCL	0.766	0.702	0.64
AppWorld	0.738	0.677	0.57
ScienceWorld	0.759	0.698	0.62
Odyssey	0.744	0.685	0.59
Main pooled	0.752	0.691	0.61

A.6 CROSS-MODEL REPLAY

These tables retain the exact matrices visualized in main-text Figure 4. Diagonal entries are self-replay results; off-diagonal entries use a different replay model. The final table separates adjacent and distant cross-model comparisons.

Table 12: Cross-model replay for Gemma 3 IT. Size labels identify the checkpoints in Table 3. Columns are trajectory models and rows are replay models; bold cells denote self replay.

	Stable L_u AUC				Mixed D_u AUC			
Replay	1B	4B	12B	27B	1B	4B	12B	27B
1B	**0.704**	0.671	0.638	0.604	**0.638**	0.616	0.589	0.566
4B	0.682	**0.748**	0.701	0.661	0.619	**0.681**	0.647	0.611
12B	0.649	0.716	**0.786**	0.733	0.592	0.651	**0.714**	0.669
27B	0.621	0.674	0.748	**0.821**	0.571	0.613	0.681	**0.752**

Table 13: Cross-model replay for Qwen3.5. Size labels identify the checkpoints in Table 3. Columns are trajectory models and rows are replay models; bold cells denote self replay.

	Stable L_u AUC				Mixed D_u AUC			
Replay	2B	4B	9B	27B	2B	4B	9B	27B
2B	**0.687**	0.656	0.626	0.598	**0.625**	0.604	0.578	0.559
4B	0.664	**0.729**	0.689	0.647	0.607	**0.664**	0.631	0.596
9B	0.635	0.698	**0.771**	0.721	0.581	0.637	**0.701**	0.658
27B	0.609	0.659	0.735	**0.806**	0.563	0.601	0.669	**0.739**

Table 14: Replay-model specificity across model families.

Family	Comparison	Stable ΔAUC	Mixed ΔAUC
Gemma 3 IT	Self − all cross	+0.090	+0.078
Gemma 3 IT	Self − adjacent cross	+0.056	+0.049
Gemma 3 IT	Self − distant cross	+0.124	+0.106
Qwen3.5	Self − all cross	+0.087	+0.075
Qwen3.5	Self − adjacent cross	+0.054	+0.048
Qwen3.5	Self − distant cross	+0.119	+0.103
Pooled	Self − all cross	**+0.088**	**+0.076**
Pooled	Self − adjacent cross	+0.055	+0.049
Pooled	Self − distant cross	+0.121	+0.104

A.7 ROBUSTNESS AND EFFECT SIZES

Figure 5 shows the paired AUC differences and their reported bootstrap intervals. Table 15 gives the individual matched-comparator results across evaluation restrictions.

Table 15: Robustness under alternative evaluation restrictions.

	Stable			Mixed		
Setting	Ours	Comparator	Δ	Ours	Comparator	Δ
Main	0.816	0.685	+0.131	0.743	0.634	+0.109
Pre-outcome disclosure	0.773	0.662	+0.111	0.704	0.621	+0.083
Clean executions	0.808	0.681	+0.127	0.732	0.631	+0.101
Call matched	0.792	0.613	+0.179	0.721	0.601	+0.120
Length matched	0.786	0.601	+0.185	0.714	0.596	+0.118
Observation-length matched	0.779	0.608	+0.171	0.708	0.590	+0.118
No truncation	0.804	0.678	+0.126	0.729	0.628	+0.101
No tool errors	0.801	0.676	+0.125	0.727	0.625	+0.102
Held-out MCPMark	0.788	0.661	+0.127	0.711	0.617	+0.094

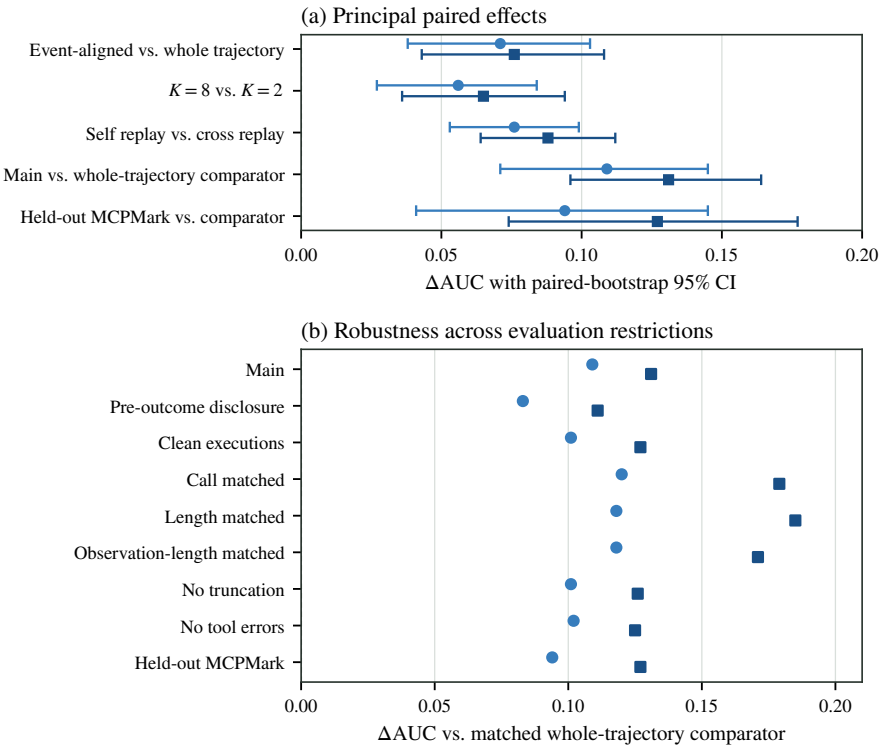

Figure 5: Paired AUC effects. Dark-blue squares denote stable-task effects and light-blue circles denote mixed-task effects. Panel (a) shows principal differences with paired-bootstrap 95% confidence intervals. Panel (b) shows differences between the proposed score and the matched whole-trajectory comparator across evaluation restrictions.

Table 16: Principal effect sizes with paired-bootstrap 95% confidence intervals.

Comparison	Stable ΔAUC	Stable 95% CI	Mixed ΔAUC	Mixed 95% CI
Event-aligned vs. whole trajectory ($K_{\text{score}} = 1$)	+0.076	[0.043, 0.108]	+0.071	[0.038, 0.103]
$K = 8$ vs. $K = 2$	+0.065	[0.036, 0.094]	+0.056	[0.027, 0.084]
Self replay vs. all cross replay	+0.088	[0.064, 0.112]	+0.076	[0.053, 0.099]
Main vs. matched comparator	+0.131	[0.096, 0.164]	+0.109	[0.071, 0.145]
Held-out MCPMark vs. comparator	+0.127	[0.074, 0.177]	+0.094	[0.041, 0.145]