Interactive Evaluation Requires a Design Science

Keyang Xuan [1 2 *] **Peiyang Song** [3 4 *] **Pan Lu** [5] **Pengrui Han** [2] **Wenkai Li** [4] **Zhenyu Zhang** [5] **Zexue He** [5] **Wenyue Hua** [6] **Manling Li** [7] **Jiaxuan You** [2] **Adrian Weller** [8] **Yizhong Wang** [1 †] **Jiaxin Pei** [1 5 †]

Abstract

AI evaluation is undergoing a structural change. Large language models (LLMs) are increasingly deployed as systems that act over time through tools, environments, users, and other agents, while many evaluation practices still inherit assumptions from response-centered benchmarks (e.g., fixed inputs, isolated outputs, and outcome judgments that can be made from a single response). The field has begun to build interactive benchmarks, but the resulting landscape is fragmented: benchmarks differ in what interaction artifacts they admit, how trajectories are scored, and what claims their results support. This position paper argues that interactive evaluation should be treated as a principled evaluation paradigm, not merely a new family of agent benchmarks. Simply adopting previous evaluation paradigms does not suffice. We define evaluation as an autonomous mapping from evidence to judgments, and show that interactive evaluation changes both sides of this mapping: the evidence becomes interaction-generated trajectories, while the evaluation procedure must assess process, recoverability, coordination, robustness, and system-level performance. Building on this definition, we propose a two-axis taxonomy, derive design principles and reporting standards, examine representative scenarios, and analyze how longstanding evaluation challenges reappear at the trajectory level.

[1]University of Texas Austin, Austin, TX, USA [2]University of Illinois Urbana-Champaign, Urbana, IL, USA [3]California Institute of Technology, Pasadena, CA, USA [4]Carnegie Mellon University, Pittsburgh, PA, USA [5]Stanford University, Stanford, CA, USA [6]Microsoft Research, Redmond, WA, USA [7]Northwestern University, Evanston, IL, USA [8]University of Cambridge, Cambridge, UK. Correspondence to: Keyang Xuan <keyangx@utexas.edu>, Peiyang Song <psong2@andrew.cmu.edu>.

Accepted to the 1st Workshop on Combining Theory and Benchmarks, CTB@ICML 2026, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

AI evaluation is undergoing a visible transition. For much of modern NLP, benchmark design was organized around response-centered evaluation: models received fixed instances and were judged by the quality of standalone final outputs, rather than by behavior unfolding through interaction. As Figure 1 illustrates, benchmark design has increasingly expanded toward executable, grounded, and interactive settings. This shift reflects a broader change in what large language models (LLMs) are expected to do: they are increasingly evaluated not only as standalone generators (Gruver et al., 2023), but as systems acting through tools (Qin et al., 2023), interfaces (Deng et al., 2023b), environments (Zhang et al., 2023), users (Chalamalasetti et al., 2023; Lee et al., 2022), and other agents (Chen et al., 2023; Li et al., 2023). Across web navigation (Zhou et al., 2023a), tool use (Guo et al., 2024), coding (Jimenez et al., 2023), formal mathematics (Collins et al., 2025) and multi-agent coordination (Emde et al., 2026), evaluation object is shifting from an single response to behavior that unfolds through feedback, state, and consequence (Wang et al., 2023; Xi et al., 2025; Froger et al., 2025; Oktar et al., 2025). This is not a cosmetic change in benchmark format. It changes what evidence an evaluation must observe and what claim a score can support.

The question is therefore no longer whether interactive evaluation ought to matter. Recent benchmarks have already established its importance. The urgent question instead is how interactive evaluation should be designed so that it becomes interpretable, comparable, and scientifically useful.

Existing interactive evaluations vary in the artifacts they record, the substrates and environments they include, the extent to which later states depend on earlier actions, and the procedures by which trajectories become scores. Some primarily test long-horizon goal completion in grounded environments (Feng et al., 2026); others emphasize tool-user interaction (Yao et al., 2024; Lu et al., 2025; Ibrahim et al., 2025), process-level reward modeling (Wang et al., 2026), social interaction (Zhou et al., 2023b), or robustness under imperfect guidance (Fu et al., 2026). These differences are productive, but they are consequential to evaluation. A benchmark that records a trajectory but scores only final

success supports a different claim from one that measures recoverability, risk, coordination, or adaptation. Without a shared conceptual frame, these distinctions are easy to flatten into "agent evaluation," obscuring which claims are well supported and which remain systematically under-covered.

This design problem is sharpened by an uneven transition across the evaluation ecosystem. As Figure 1 suggests, task-driven and interactive evaluation appear more prominently in recent frontier-lab reports, while academic benchmark work still centers more on response-centered evaluation. We do not view this divergence as a sophistication gap, but as different optimization pressures: academic benchmarks often prioritize comparability, reproducibility, and scalable problem definition, while deployed systems require evidence about long-horizon interaction, tool use, robustness, and behavior under feedback. Different communities are therefore optimizing for different evidence and claims, making it important to ask not only what trajectories a benchmark records, but what evaluation program maps them to judgments.

Therefore, this paper argues that:

> **Position.** Interactive evaluation should be built as a design science for evaluating systems acting through trajectories. The field does not merely need more interactive benchmarks; it needs explicit principles for specifying what interaction artifacts enter evaluation and how an evaluation program maps those artifacts to judgments.

This paper develops that position from the perspective of evaluation itself. We first explain why response-centered evaluation was historically useful and why its assumptions become insufficient when systems act in closed loop. We then define evaluation as an autonomous program $E : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the admissible evidence available to the evaluator and E is the procedure that maps that evidence to judgments. Interactive evaluation changes both parts: $\mathcal{X}$ expands from final responses to interaction-generated trajectories, and E must assess not only final correctness but also process quality, recoverability, coordination, safety, efficiency, and robustness. This framing lets us build a taxonomy of interactive evaluation, use it to identify where current benchmarks concentrate and what they miss, and derive principles for designing future evaluations.

Our contributions are fourfold: **1)** We give a compact definition of interactive evaluation and clarify its boundary cases (Sec. 3). **2)** We propose a 2D taxonomy organized around evaluation inputs and evaluation programs, making benchmarks comparable without forcing them into one task domain (Sec. 4). **3)** We derive principles and a roadmap for benchmark design, reporting, and infrastructure (Sec. 5). **4)** We illustrate the framework in representative coding-agent and multi-agent social-system scenarios (App. D), then discuss risks that arise when classic evaluation problems–overfitting, gaming, leakage, brittleness, and reproducibility–become trajectory-level problems (Sec. 6).

We therefore invite the community to treat interactive evaluation as a design science: one that specifies what trajectory artifacts count as evidence, how those artifacts are mapped to judgments, and what claims the resulting scores can legitimately support. The goal is not simply to evaluate harder tasks, but to evaluate interactive systems in ways that are interpretable, comparable, and scientifically useful.

2. Rethinking Evaluation Beyond Response-centered Evaluation

Response-centered evaluation is not a mistake to be discarded. It became dominant because it solved real methodological problems. Fixed datasets and standardized task instances made model comparison scalable; single-output scoring made results legible; and many AI tasks could plausibly be represented as input-output mappings, including classification (Bowman et al., 2015; Warstadt et al., 2019), question answering (Rajpurkar et al., 2016), translation (Goyal et al., 2022), summarization (Gliwa et al., 2019; Kryściński et al., 2022), and broad capability probing (Hendrycks et al., 2020; Srivastava et al., 2023). In those settings, most relevant evidence is provided in the instance, the system's response is the natural unit of assessment, and later evaluation conditions do not depend on earlier model behavior.

Why the Old Assumptions Worked. The response-centered paradigm matched a particular view of AI systems: a model receives an input x, produces an output y, and evaluation asks whether y has the desired relation to a reference, rubric, or judge. Its strength was not only convenience. It offered essential values including comparability, aggregation, and repeatability. Interactive evaluation should supplement response-centered evaluation where interaction is constitutive of the capability being measured; it should not turn every evaluation into an expensive simulation by default.

Why Interaction Breaks the Fit. The fit breaks when the evaluated system acts over time. A web action can reveal or hide future opportunities; a tool call can modify persistent state; a user reply can change after clarification; another agent can adapt strategically; and an error can become recoverable rather than terminal. In these cases the evidence needed for judgment is not contained in the initial prompt or the final answer. It is generated through the trajectory. Several latest benchmarks (Zhou et al., 2023a; Xie et al., 2024; Trivedi et al., 2024; Wang et al., 2023; Lu et al., 2025) make this visible by requiring systems to operate through executable environments, tools, or conversational feedback.

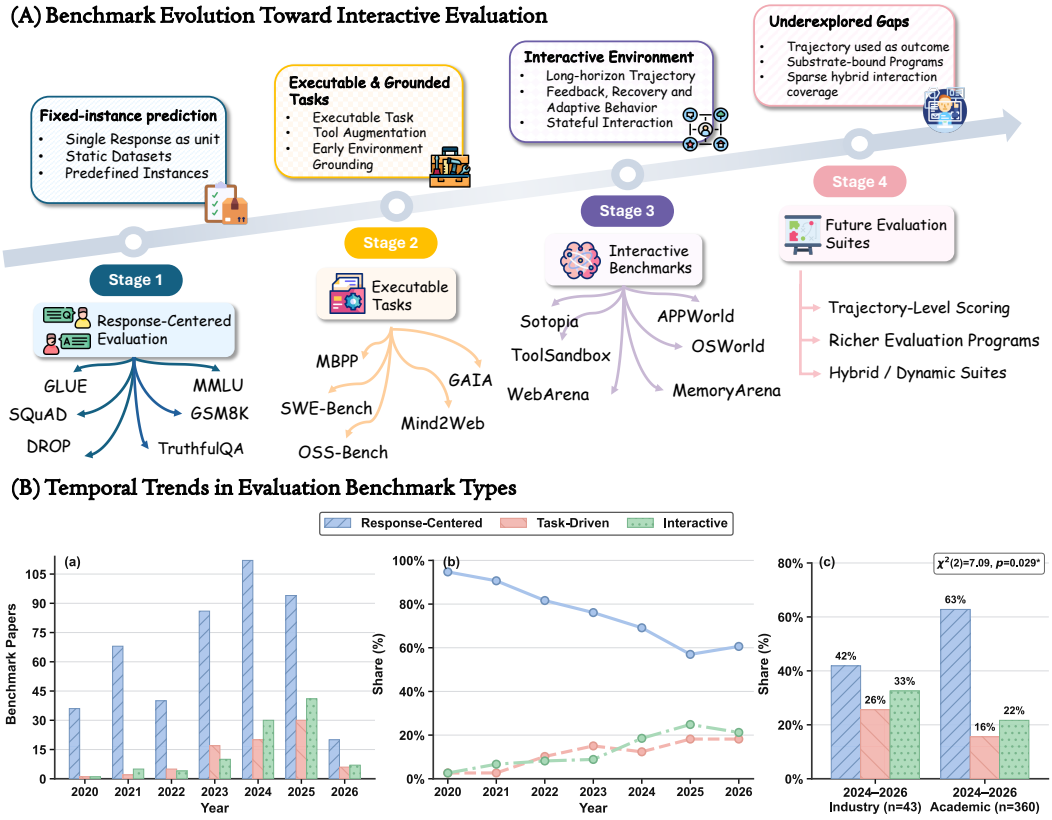

Figure 1. **The rise of interactive evaluation motivates a design-science view.** (A) Benchmarks have evolved from fixed-instance response evaluation to executable, grounded, and interactive settings, revealing gaps in trajectory-level scoring and hybrid/dynamic evaluation (B) Empirical patterns support this shift: academic benchmark papers show growth in task-driven and interactive evaluation by count (a) and share (b), while frontier-lab public materials show a more balanced evaluation-type mix than the academic corpus (c). Benchmark collection details are provided in Appendix C, with representative benchmarks in Appendix E.

Why Interaction Itself Must Be Evaluated. Interaction is not merely a path toward an answer; it is often the capability of interest. A coding agent that passes tests by making a brittle patch has not demonstrated the same competence as one that isolates the fault, preserves interfaces, and recovers from failing tests. A social agent that achieves a local objective by confusing a counterpart has not demonstrated the same competence as one that coordinates transparently. Once process changes the meaning of success, outcome-only measurement becomes under-specified: interactive evaluation must ask how evidence was gathered, which actions changed the state, whether mistakes were detected, and what costs or risks were incurred.

A Minimal Notion of Interaction. We use *interaction* in a consequential sense. A setting is interactive when the system operates in an external loop involving tools, environments, users, or other agents; when what it encounters next depends at least partly on earlier behavior; and when that dependence matters for evaluation. Multiple turns alone are insufficient. A scripted dialogue whose later prompts are fixed in advance may be sequential, but it is not interactive in the evaluative sense used here. Conversely, a short tool-use task may be interactive if the tool result changes the subsequent evidence, state, or scoring conditions.

3. Definition and Scope of Interactive Evaluation

An evaluation can be viewed as an autonomous program

$$E : \mathcal{X} \to \mathcal{Y},$$

where $\mathcal{X}$ is the domain of artifacts accepted as evidence and $\mathcal{Y}$ is the space of evaluative outputs, such as scores, rankings, pass/fail decisions, diagnostic reports, or qualitative judgments. This framing is intentionally simple but inevitable. Any scalable evaluation must decide what artifacts can be submitted to the evaluator and what procedure maps those artifacts to claims.

Definition. Interactive evaluation is evaluation in which the admissible evidence $\mathcal{X}$ includes trajectories generated by consequential interaction, and the evaluation program E maps those trajectories to judgments about system-level performance.

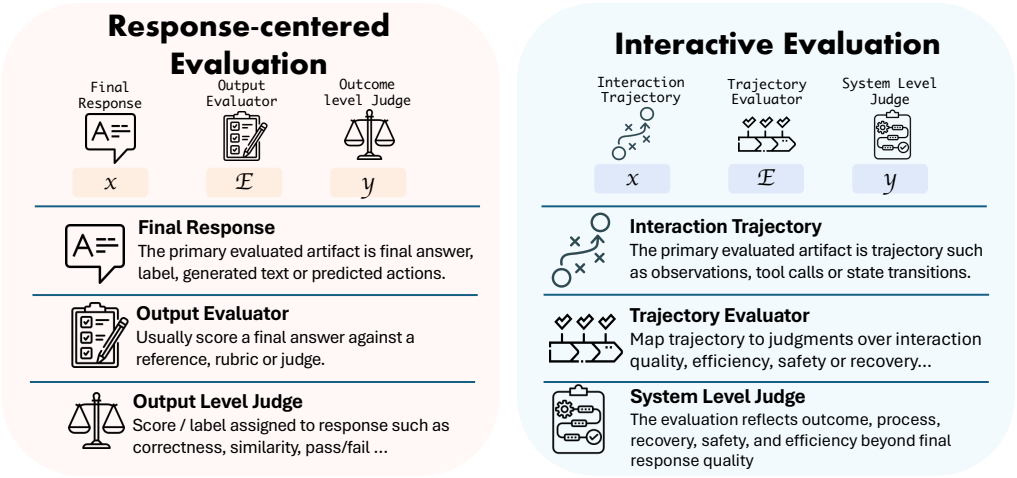

Figure 2. Response-centered evaluation judges final responses; interactive evaluation judges trajectories as evidence of system-level performance and process.

What Changes in $\mathcal{X}$. In response-centered evaluation, the central artifact is a final answer, label, generated text, or prediction for a predefined instance. In interactive evaluation, the artifact is a trajectory: observations, actions, tool calls, state transitions, user or counterpart responses, intermediate artifacts, costs, constraints, and final outcomes. Such trajectory may rise in web environment (Zhou et al., 2023a), an operating system (Xie et al., 2024), a set of stateful apps (Trivedi et al., 2024), a tool-user simulation (Yao et al., 2024; Lu et al., 2025), or a social/multi-agent world (Zhou et al., 2023b; Zhu et al., 2025). What matters is not the substrate alone, but whether the recorded artifact preserves the action-dependent structure needed to judge performance.

What Changes in E. The evaluator also changes. A response-centered evaluator can often score a final answer against a reference, rubric, or judge. An interactive evaluator must decide which trajectory properties count: completion, progress, constraint satisfaction, efficient exploration, safe tool use, recoverability after error, cooperation, communication quality, or resilience under disruption. Thus E is not merely an answer checker; it is a trajectory-to-judgment procedure. It may combine executable tests, state checks, human or model judges, process annotations, penalties for unsafe actions, and aggregation across stochastic runs.

Boundary Cases. Boundary cases are helpful in positioning the scope precisely. This definition excludes three common false positives. First, multiple turns are not enough if the sequence is predetermined and earlier behavior does not affect later conditions. Second, tool calls are not enough if they are only hidden computation and do not change the evaluation evidence or state. Third, chain-of-thought or self-reflection is not enough by itself: internal reasoning may be valuable evidence when exposed under a protocol, but interaction requires an external loop whose continuation is partly action-dependent. The boundary is therefore evaluative rather than stylistic: a setting counts when judging the system requires evidence from consequential interaction.

4. Taxonomy of Interactive Evaluation

The definition above suggests that the main design problem in interactive evaluation is not simply whether a benchmark contains interaction, but whether its trajectory evidence is matched to an appropriate evaluation program. We therefore use the evaluation mapping $E : \mathcal{X} \to \mathcal{Y}$ as a diagnostic framework. Then interactive evaluations differ along two axes: what interaction-generated artifacts enter $\mathcal{X}$, and how E maps those artifacts to judgments. This two-axis view avoids a common confusion: task domain, substrate, metric, and judgment protocol are not separate top-level taxonomies. They are properties of either the input artifact or the evaluation program.

4.1. Axis 1: Evaluation Inputs

The first axis asks what interactive artifact is passed into evaluation. The central object is a trajectory, but trajectories differ in what they connect the system to.

Tools and Environments. Many current benchmarks evaluate agents in executable digital or tool-mediated settings. WebArena, Mind2Web, BrowseComp, OSWorld, AndroidWorld, and AppWorld test interaction with web pages, interfaces, operating systems, mobile environments, or stateful applications (Deng et al., 2023b; Zhou et al., 2023a; Wei et al., 2025; Xie et al., 2024; Rawles et al., 2024; Trivedi et al., 2024). These inputs expose action-dependent state: clicks, API calls, file edits, or app operations change what the agent can observe later.

Users. User-centered trajectories evaluate whether systems can interact effectively with people under incomplete, ambiguous, or evolving instructions. These evaluations focus not only on task completion, but also on whether systems can clarify user intent, maintain alignment with human goals, communicate uncertainty appropriately, and adapt as user preferences or requirements change over time. τ-bench, IN3, ToolSandbox, MINT, RealWebAssist, and AgentClinic represent this direction by making user feedback or simulated user behavior part of the trajectory (Yao et al., 2024; Chowa et al., 2026; Lu et al., 2025; Ye et al., 2026; Schmidgall et al., 2024). The artifact is therefore not just a task log; it includes how the system negotiates information, uncertainty, and coordination with users throughout the interaction.

Other Agents. Multi-agent trajectories evaluate coordination, competition, delegation, negotiation, and emergent behavior. SOTOPIA, MultiAgentBench, BattleAgentBench, Intellagent, MASEval, and CooperBench show that the relevant evidence may include messages, role assignments, joint plans, conflicts, and counterpart adaptation (Zhou et al., 2023b; Zhu et al., 2025; Wang et al., 2024; Levi & Kadar, 2025; Emde et al., 2026; Khatua et al., 2026).

Hybrid and Dynamic Systems. The most deployment-like evaluations will combine tools, users, agents, memory, and changing environments. MemoryArena (He et al., 2026) and large environment suites like AI Gamestore (Ying et al., 2026) and ARC-AGI-3 (Foundation, 2026) point toward persistent state and cross-session dependencies (Froger et al., 2025; Backlund & Petersson, 2025). This category remains comparatively underexplored, but it is central for evaluating systems that must remain reliable across time rather than only within a single task episode, and we anticipate that this direction will soon become vital to a wide range of real-world tasks.

4.2. Axis 2: Evaluation Programs

The second axis asks how trajectories are mapped to judgments. Several measurement logics recur, and strong benchmarks should state which ones they support.

Task Success. The most common program checks whether the final state satisfies a goal: a web task completed, a repository issue resolved, a mobile task performed, or an app state updated (Zhou et al., 2023a; Jimenez et al., 2023; Rawles et al., 2024; Trivedi et al., 2024). This is indispensable, but insufficient when two trajectories reach the same final state through very different risks or costs. This is the base-case evaluation where most principles from response-centered evaluation transfer, and will be supplemented by the interactive evaluation measures below.

Process Quality and Efficiency. Interactive settings make intermediate behavior evaluable. A benchmark may score tool choice, action economy, state exploration, code-edit locality, communication clarity, or unnecessary disruption (Wang et al., 2023; Lu et al., 2025; Yue et al., 2026; Li et al., 2026; Fan et al., 2026). These measures are important because poor processes often predict brittle deployment even when final success is achieved.

Recoverability and Robustness. A trajectory-level evaluator can test whether systems detect mistakes, revise plans, resist misleading guidance, and adapt to changing conditions (Debenedetti et al., 2024; Fu et al., 2026; Froger et al., 2025). This is one of the clearest advantages of interactive evaluation: failure is not merely an endpoint, but an event that can be observed, repaired, or amplified.

Safety, Alignment, and Social Competence. When systems interact with users or agents, evaluation must include norm-sensitive behavior, cooperation, honesty about uncertainty, and avoidance of manipulative or unsafe strategies (Zhou et al., 2023b; Zhang et al., 2024; Zhou et al., 2024; Khatua et al., 2026). These properties are often invisible in final-answer scoring but central to whether an interactive system should be trusted.

Taxonomy Claim. A benchmark is a point, or a region, in a two-dimensional design space: the interaction artifact it admits as evidence, and the trajectory-to-judgment program it implements. This view lets us compare benchmarks without pretending that all interactive tasks measure the same capability.

4.3. Putting It Together: The 2D taxonomy

Figure 3 maps a representative set of interactive evaluations into the 2D space defined above. We do not claim to provide an exhaustive census. Instead, to obtain a representative set, we prioritize works based on academic impact, such as citation counts; open-source adoption, such as GitHub stars; and use in frontier-model evaluation reports. This mapping provides a global view of the main focuses of existing interactive evaluation works, while also highlighting areas that remain underexplored. From mapping result we derive the following observations and insights:

Trajectory Evidence Remains Outcome-Centered. The most visible pattern is the concentration around Task Success and the relative sparsity of Recoverability and Robustness. This concentration indicates a mismatch between trajectory evidence and evaluation programs. Many works admit trajectories as evidence, but still evaluate them as final outcomes. As a result, interactive evaluation has of-

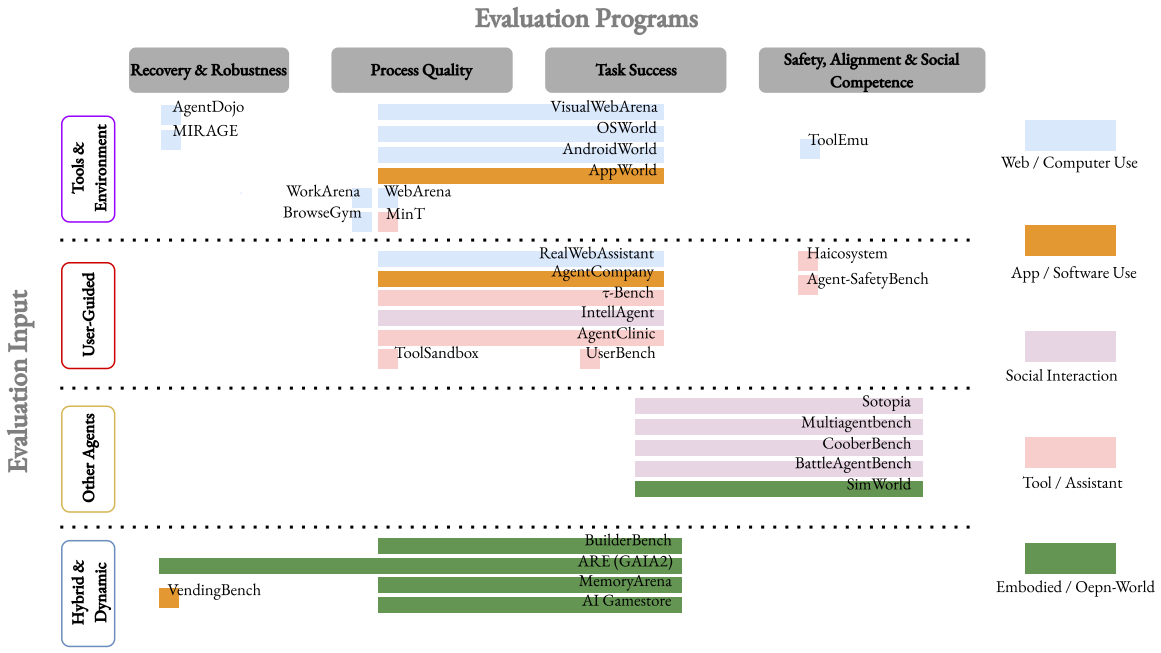

Figure 3. Mapping the current interactive evaluation landscape in a two-dimensional design space.

ten adopted trajectory recording without fully developing trajectory-level judgment. When trajectories contain actions, observations, state changes, and feedback, evaluation should also support judgments about process quality, cost, risk, and recoverability, rather than treating them only as evidence for a single success label.

Evaluation Programs Remain Substrate-Bound. Many interactive evaluations are organized around evaluation inputs rather than evaluation programs. Tools and Environments are usually evaluated through Task Success or Process Quality and Efficiency; Other Agents are more often evaluated through Safety, Alignment, and Social Competence; and Hybrid and Dynamic Systems remain sparse across programs. This suggests that current interactive evaluation design is often led by the evaluation substrate, with the evaluation program following from what is easiest or most natural to measure in that setting. As a result, fewer benchmarks begin from an explicit evaluative claim, such as recoverability under disruption, safety under persistent tool use, or robustness under changing user intent, and then design the input artifact, interaction protocol, and scoring procedure needed to support that claim.

Hybrid and Dynamic Systems Lack Robust Coverage. As LLM systems face more complex end-to-end tasks, evaluation must increasingly account for hybrid and dynamic interaction evidence. Yet our mapping shows that this category remains the sparest evaluation input category. Existing work in this region remains concentrated around Task Success and Process Quality and Efficiency, leaving recoverability, safety, alignment, and social competence comparatively underdeveloped. This creates a mismatch between current evaluation coverage and likely deployment risk. In systems with persistent state, cross-session dependencies, and hybrid interaction loops, errors may accumulate over time instead of ending with a single failed task. As systems move toward longer horizons and more persistent operation, interactive evaluation must make these conditions first-class objects of measurement rather than treating them as future extensions.

5. Principles and Roadmap for Interactive Evaluation

The taxonomy is useful only if it changes design practice. The gaps above reveal a recurring failure mode: interactive evaluations often record trajectories without specifying what claims those trajectories support. Evaluation programs then follow the interaction substrate rather than the intended capability claim. This is especially problematic in hybrid settings, where persistence, state, and cross-session dependencies change how success, failure, and risk should be interpreted. We therefore combine principles and roadmap: the choices that make benchmarks interpretable also define what the field should build next.

Specify the System and Trajectory Evidence. Interactive evaluations should specify the evaluated system, its accessible resources, the trajectory evidence, and the claims that evidence supports. Model identity alone is insufficient: tool wrappers, memory, retrieval, execution sandboxes, user simulators, and orchestration policies can all affect performance. Recording trajectories is also insufficient unless benchmarks state whether those traces support judgments about task success, process quality, recoverability, safety,

efficiency, coordination, or other system-level properties.

Specify the Interaction Protocol. Trajectory evidence is only interpretable relative to the protocol that generates it. Evaluation should therefore specify the conditions under which interaction unfolds, including the initial state distribution, allowed actions, observation space, counterpart behavior, stopping rules, randomness, persistence, and reset conditions. These details determine what opportunities the system had, what constraints it faced, and whether two trajectories are meaningfully comparable. Protocol documentation is the interactive analogue of dataset documentation: without it, scores may reflect hidden differences in interaction setup rather than differences in system capability.

Design for Perturbation and Repair. As evaluation tasks move toward more complex and dynamic environments, clean task completion becomes an insufficient test of interactive competence. Future benchmarks should therefore evaluate whether systems can remain effective when interaction conditions change, including ambiguity, misleading feedback, partial failure, state drift, and counterpart adaptation. These conditions should not be treated as adversarial add-ons; they are central to interactive evaluation because they reveal whether a system can detect problems, revise its strategy, recover from errors, and continue acting effectively under evolving conditions.

Separate Outcome, Process, and Risk. Interactive evaluations should distinguish what the system ultimately achieves from how it achieves it and what risks it creates along the way. A single scalar score may be useful for ranking systems, but it should not hide distinct evaluative claims. Interactive benchmarks should therefore report final success separately from trajectory-level properties such as action cost, unsafe behavior, recovery behavior whenever those dimensions matter. Aggregated scores can still be useful, but they should be treated as summaries of multiple reported dimensions rather than as the only evidence of system capability.

Build Shared Infrastructure without Freezing the Design Space. The field needs reusable environments, logging schemas, trajectory viewers, evaluation harnesses, and reporting templates. At the same time, standardization should preserve diversity in domains and protocols. A healthy roadmap moves from response-centered benchmarks, through executable and tool-augmented tasks, toward interactive suites that make protocol, state, and measurement logic first-class objects.

6. Risks and Open Issues in Interactive Evaluation

Interactive evaluation inherits many validity concerns from response-centered benchmarking, but it relocates them from isolated instances and outputs to action-dependent trajectories. It also introduces risks that arise only when evaluation unfolds through state, feedback, users, tools, or other agents. We therefore distinguish two classes of issues: longstanding evaluation risks that reappear at the trajectory level, and risks that are native to consequential interaction.

6.1. Longstanding Evaluation Risks at the Trajectory Level

Overfitting, Leakage, and Gaming. Static benchmarks can be memorized; interactive benchmarks can be policy-gamed. In response-centered evaluation, leakage often concerns exposure to test inputs or reference answers. In interactive evaluation, leakage can occur through environment state, public task templates, tool APIs, simulator regularities, predictable user models, or evaluator heuristics. The resulting failure mode is not only that a model knows the answer, but that a system learns how to behave strategically inside the benchmark. Agents may exploit simulator quirks, avoid meaningful exploration, optimize for superficial trajectory signals, or discover shortcuts that satisfy the scorer without demonstrating the intended competence. Mitigations should therefore operate at the trajectory level: held-out environments, procedurally generated tasks, private or refreshed evaluation suites, adversarial perturbations, and audits of suspiciously efficient or unnatural trajectories.

Distribution Shift and Benchmark Brittleness. Benchmarks have always risked measuring performance under narrow distribution. Interactive evaluation makes this problem more acute because small changes in interface, timing, initial state, tool behavior, or counterpart response can alter the trajectory itself. This sensitivity should not be dismissed as mere noise: deployment also involves shifting states, imperfect instructions, and changing counterparts. However, benchmarks must distinguish robustness failures from protocol artifacts. Reporting should include variance across seeds, environments, users, perturbations, and state initializations, and should identify whether failures reflect missing capability, brittle policies, or underspecified interaction rules.

Reproducibility and Comparability. Response-centered benchmarks gained much of their scientific value from shared instances, fixed scoring, and repeatable comparison. Interactive evaluation weakens these assumptions because runs may depend on stochastic environments, human behavior, long-context traces, and mutable state. Without standardized logs, environment versioning, replay mechanisms, and explicit protocol documentation, interactive scores can become difficult to interpret or reproduce, making benchmarks more realistic but less verifiable.

6.2. Risks Native to Interaction

Protocol Dependence and Standardization–Diversity Tradeoff. In interactive evaluation, the protocol is part of the measurement instrument. Observation spaces, stopping rules, persistence, randomness, counterpart behavior, and reset conditions shape what the system can discover and what failures mean. Benchmark should therefore document protocols with the same care that dataset benchmarks document data construction. At the same time, the field faces a second risk: premature convergence on a small set of protocols or environments. Shared logging schemas, reporting standards, and replay infrastructure are necessary for comparability, but excessive standardization can narrow the range of interaction patterns through which competence is defined. A useful ecosystem should standardize how evaluative claims are specified and reported while preserving diversity in environments, interaction substrates, counterpart models, and trajectory-level measurement programs.

Fidelity, Control, and Simulator Artifacts. Interactive evaluation must decide how much of deployment to reproduce or abstract away. High-fidelity environments can provide richer evidence about situated behavior, but they are expensive, noisy, and harder to control. Controlled simulators improve repeatability and comparison, but may reward strategies that exploit simulator artifacts rather than genuine interactive competence. There is no universal optimum between realism and control. Benchmarks should instead state which deployment conditions they model faithfully, which they deliberately abstract away, and which claims their level of fidelity can and cannot support.

Human and Counterpart Variance. When users, judges, or other agents participate in evaluation, the counterpart becomes part of the benchmark. Human counterparts vary in patience, expertise, preference, interpretation, and willingness to cooperate; simulated users scale more easily but may encode narrow or unrealistic behavior. Multi-agent settings add further variance because other agents may adapt strategically. Reporting should specify who or what plays the counterpart, how consistency is controlled, whether counterpart behavior is randomized or fixed, and whether conclusions depend on a particular user or agent model.

Cost, Accessibility, and Auditability. As interactive evaluations move toward long-horizon, stateful, human-in-the-loop, or multi-agent tasks, cost becomes a methodological constraint. Environment steps, repeated runs, human participation, trace storage, and judge evaluation can make benchmarks difficult for smaller academic groups, independent auditors, or public-interest evaluators to reproduce. The community will need lightweight diagnostic subsets, replayable traces, shared harnesses, and public reporting standards. Without such infrastructure, interactive evaluation may become impressive but inaccessible and unverifiable.

7. Scope and Implications of the Position

The risks above already address several broad alternative views to interactive evaluation: that it may be costly, protocol-dependent, difficult to reproduce, sensitive to simulator artifacts, and vulnerable to trajectory-level gaming. We discuss several additional alternative views in Appendix A. Our position is therefore not that every benchmark should become interactive, or that interactive evaluation always requires expensive high-fidelity simulation. Rather, the central question is when interaction is constitutive of the capability claim and what kinds of evaluative claims trajectory evidence can legitimately support. This section clarifies the scope of interactive evaluation from that perspective.

Interactive Evaluation Is Not Just Agent Evaluation. Our argument is not that every "agent" requires interactive evaluation. Agent benchmarks can remain response-centered evaluation when actions do not affect later conditions or only the final output is judged. Conversely, non-agent systems may also require interactive evaluation when their behavior unfolds through tools, web environments, users, or other external loops. The key question is therefore not what the system is called, but what evidence the evaluation needs in order to support its claim.

Trajectory-Level Evaluation Is Claim-dependent. We do not attempt to claim that every evaluation should become a high-fidelity simulation at the cost of scalability, or that task success should be replaced. Task success remains indispensable when the claim concerns final completion. The problem arises when a benchmark records trajectories but treats them only as evidence for a final success label. If the claim concerns process quality, recoverability, safety, efficiency, coordination, or robustness, the evaluation program must preserve and score the relevant trajectory evidence. The design challenge is to match the cost and fidelity of interactive evaluation to the claim.

8. Conclusion

In this position paper, we argue that interactive evaluation must be designed, not merely adopted. As AI systems increasingly act through consequential interactions, the field needs a systematic and unified framework for designing interactive evaluations that support comparison, reproducibility, and extension. We frame interactive evaluation as trajectory-based, system-level evaluation under action-dependent conditions, organized by two questions: what interaction artifacts enter evaluation, and how those artifacts are mapped to judgments. This framing clarifies why response-centered benchmarks remain useful but insufficient, why current interactive benchmarks should not be treated as interchangeable, and what the field must build next such as explicit protocols, richer trajectory measures, robustness tests, shared infrastructure, and reporting stan-

dards that make interactive scores interpretable. We therefore call on the community to design interactive evaluation before merely adopting it as the next benchmark format.

Acknowledgment

We thank Daniel Fried (CMU) and Katie M. Collins (Cambridge/MIT/Princeton) for discussions and helpful feedback on earlier versions of this paper.

References

Abdelnabi, S., Gomaa, A., Sivaprasad, S., Schonherr, L., and Fritz, M. Cooperation, competition, and maliciousness: Llm-stakeholders interactive negotiation. *Advances in Neural Information Processing Systems 37*, 2023. URL https://api.semanticscholar.org/CorpusID:263310628.

Ailem, M., Marazopoulou, K., Siska, C., and Bono, J. Examining the robustness of llm evaluation to the distributional assumptions of benchmarks. In *Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://api.semanticscholar.org/CorpusID:269430693.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C. J., Terry, M., Le, Q. V., and Sutton, C. Program synthesis with large language models. *ArXiv*, abs/2108.07732, 2021. URL https://api.semanticscholar.org/CorpusID:237142385.

Backlund, A. and Petersson, L. Vending-bench: A benchmark for long-term coherence of autonomous agents. *arXiv preprint arXiv:2502.15840*, 2025.

Balloccu, S., Schmidtová, P., Lango, M., and Dusek, O. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *ArXiv*, abs/2402.03927, 2024. URL https://api.semanticscholar.org/CorpusID:267499939.

Bowman, S., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 632–642, 2015.

Chalamalasetti, K., Götze, J., Hakimov, S., Madureira, B., Sadler, P., and Schlangen, D. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 11174–11219, 2023.

Chandran, N., Sitaram, S., Gupta, D., Sharma, R., Mittal, K., and Swaminathan, M. Private benchmarking to prevent contamination and improve comparative evaluation of llms. *ArXiv*, abs/2403.00393, 2024. URL https://api.semanticscholar.org/CorpusID:268201479.

Chen, W., Su, Y., Zuo, J., Yang, C., Yuan, C., Chan, C.-M., Yu, H., Lu, Y., Hung, Y.-H., Qian, C., et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2023.

Chowa, S. S., Alvi, R., Rahman, S. S., Rahman, M. A., Raiaan, M. A. K., Islam, M. R., Hussain, M., and Azam, S. From language to action: a review of large language models as autonomous agents and tool users. *Artificial Intelligence Review*, 2026.

Collins, K. M., Frieder, S., Bayer, J., Loader, J., Lim, J., Song, P., Zaiser, F., Zhou, L., Li, S., Looi, S.-Z., et al. Ai impact on human proof formalization workflows. In *The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025*, 2025.

Davidson, T. R., Veselovsky, V., Josifoski, M., Peyrard, M., Bosselut, A., Kosinski, M., and West, R. Evaluating language model agency through negotiations. *ArXiv*, abs/2401.04536, 2024. URL https://api.semanticscholar.org/CorpusID:266899922.

Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., and Tramèr, F. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in Neural Information Processing Systems*, 37:82895–82920, 2024.

Deng, C., Zhao, Y., Tang, X., Gerstein, M. B., and Cohan, A. Investigating data contamination in modern benchmarks for large language models. In *North American Chapter of the Association for Computational Linguistics*, 2023a. URL https://api.semanticscholar.org/CorpusID:265220695.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023b.

Emde, C., Rubinstein, A., Goel, A., Heakl, A., Yun, S., Oh, S. J., and Gubri, M. Maseval: Extending multi-agent evaluation from models to systems. *arXiv preprint arXiv:2603.08835*, 2026.

Fan, S., Ye, X., Huo, Y., Chen, Z.-Y., Guo, Y., Yang, S., Yang, W., Ye, S., Chen, J., Chen, H., Cong, X., and Lin,

Y. Agentprocessbench: Diagnosing step-level process quality in tool-using agents, 2026. URL https://arxiv.org/abs/2603.14465.

Feng, Y., Sun, J., Yang, Z., Ai, J., Li, C., Li, Z., Zhang, F., He, K., Ma, R., Lin, J., et al. Longcli-bench: A preliminary benchmark and study for long-horizon agentic programming in command-line interfaces. *arXiv preprint arXiv:2602.14337*, 2026.

Foundation, A. P. Arc-agi-3: A new challenge for frontier agentic intelligence, 2026. URL https://arxiv.org/abs/2603.24621.

Froger, R., Andrews, P., Bettini, M., Budhiraja, A., Cabral, R. S., Do, V., Garreau, E., Gaya, J.-B., Laurençon, H., Lecanu, M., et al. Are: Scaling up agent environments and evaluations. *arXiv preprint arXiv:2509.17158*, 2025.

Fu, Y., Qiu, R., Wang, X., Sansom, J., Prabhu, S. A., Tang, H., Kim, J., Sohn, S., and Lee, H. Beyond blind following: Evaluating robustness of llm agents under imperfect guidance. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6591–6618, 2026.

Gliwa, B., Mochol, I., Biesek, M., and Wawer, A. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, 2019.

Golchin, S. and Surdeanu, M. Time travel in llms: Tracing data contamination in large language models. *ArXiv*, abs/2308.08493, 2023. URL https://api.semanticscholar.org/CorpusID:260925501.

Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., and Fan, A. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022.

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36: 19622–19635, 2023.

Guo, Z., Cheng, S., Wang, H., Liang, S., Qin, Y., Li, P., Liu, Z., Sun, M., and Liu, Y. Stabletoolbench: Towards stable large-scale benchmarking on tool learning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11143–11156, 2024.

He, Z., Wang, Y., Zhi, C., Hu, Y., Chen, T.-P., Yin, L., Chen, Z., Wu, T. A., Ouyang, S., Wang, Z., et al. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. *arXiv preprint arXiv:2602.16313*, 2026.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hendrycks, D., Basart, S., Kadavath, S., Mazeika, M., Arora, A., Guo, E., Burns, C., Puranik, S., He, H., Song, D. X., and Steinhardt, J. Measuring coding challenge competence with apps. *ArXiv*, abs/2105.09938, 2021. URL https://api.semanticscholar.org/CorpusID:234790100.

Ibrahim, L., Huang, S., Ahmad, L., Bhatt, U., and Anderljung, M. Towards interactive evaluations for interaction harms in human-ai systems. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pp. 1302–1310, 2025.

Jacovi, A., Caciularu, A., Goldman, O., and Goldberg, Y. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:258741333.

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. Livecodebench: Holistic and contamination free evaluation of large language models for code. *ArXiv*, abs/2403.07974, 2024. URL https://api.semanticscholar.org/CorpusID:268379413.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

Kasai, J., Sakaguchi, K., Takahashi, Y., Bras, R. L., Asai, A., Yu, X. V., Radev, D. R., Smith, N. A., Choi, Y., and Inui, K. Realtime qa: What's the answer right now? *ArXiv*, abs/2207.13332, 2022. URL https://api.semanticscholar.org/CorpusID:251105205.

Khatua, A., Zhu, H., Tran, P., Prabhudesai, A., Sadrieh, F., Lieberwirth, J. K., Yu, X., Fu, Y., Ryan, M. J., Pei, J., et al. Cooperbench: Why coding agents cannot be your teammates yet. *arXiv preprint arXiv:2601.13295*, 2026.

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., Ma, Z., Thrush, T., Riedel, S., Talat, Z., Stenetorp, P., Jia, R., Bansal, M., Potts, C., and Williams, A. Dynabench: Rethinking benchmarking in nlp. *ArXiv*, abs/2104.14337, 2021. URL https://api.semanticscholar.org/CorpusID:233444226.

Kryściński, W., Rajani, N., Agarwal, D., Xiong, C., and Radev, D. Booksum: A collection of datasets for long-form narrative summarization. In *Findings of the association for computational linguistics: EMNLP 2022*, pp. 6536–6558, 2022.

Lee, M., Srivastava, M., Hardy, A., Thickstun, J., Durmus, E., Paranjape, A., Gerard-Ursin, I., Li, X. L., Ladhak, F., Rong, F., et al. Evaluating human-language model interaction. *arXiv preprint arXiv:2212.09746*, 2022.

Levi, E. and Kadar, I. Intellagent: A multi-agent framework for evaluating conversational ai systems. *arXiv preprint arXiv:2501.11067*, 2025.

Lewis, M., Yarats, D., Dauphin, Y., Parikh, D., and Batra, D. Deal or no deal? end-to-end learning of negotiation dialogues. *ArXiv*, abs/1706.05125, 2017. URL https://api.semanticscholar.org/CorpusID:2454882.

Li, D., Yao, Y., Tan, Z., Liu, H., and Guo, R. Toolprmbench: Evaluating and advancing process reward models for tool-using agents. *ArXiv*, abs/2601.12294, 2026. URL https://api.semanticscholar.org/CorpusID:284910432.

Li, J., Li, R., and Liu, Q. Beyond static datasets: A deep interaction approach to llm evaluation, 2023. URL https://arxiv.org/abs/2309.04369.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., R'e, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., Wang, J., Santhanam, K., Orr, L. J., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N. S., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T. F., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525:140 – 146, 2023. URL https://api.semanticscholar.org/CorpusID:253553585.

Lu, J., Holleis, T., Zhang, Y., Aumayer, B., Nan, F., Bai, H., Ma, S., Ma, S., Li, M., Yin, G., et al. 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, 2025.

Manheim, D. and Garrabrant, S. Categorizing variants of goodhart's law. *ArXiv*, abs/1803.04585, 2018. URL https://api.semanticscholar.org/CorpusID:4715794.

McCoy, R. T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *ArXiv*, abs/1902.01007, 2019. URL https://api.semanticscholar.org/CorpusID:59599752.

Oktar, K., Collins, K. M., Hernandez-Orallo, J., Coyle, D., Cave, S., Weller, A., and Sucholutsky, I. Identifying, evaluating, and mitigating risks of ai thought partnerships. *ACM AI Letters*, 2025.

Oren, Y., Meister, N., Chatterji, N. S., Ladhak, F., and Hashimoto, T. Proving test set contamination in black box language models. *ArXiv*, abs/2310.17623, 2023. URL https://api.semanticscholar.org/CorpusID:264490730.

Perez, E., Ringer, S., Lukošiūtė, K., Nguyen, K., Chen, E., Heiner, S., Pettit, C., Olsson, C., Kundu, S., Kadavath, S., Jones, A., Chen, A., Mann, B., Israel, B., Seethor, B., McKinnon, C., Olah, C., Yan, D., Amodei, D., Amodei, D., Drain, D., Li, D., Tran-Johnson, E., Khundadze, G. R., Kernion, J., Landis, J. M., Kerr, J., Mueller, J., Hyun, J., Landau, J. D., Ndousse, K., Goldberg, L., Lovitt, L., Lucas, M., Sellitto, M., Zhang, M., Kingsland, N., Elhage, N., Joseph, N., Mercado, N., DasSarma, N., Rausch, O., Larson, R., McCandlish, S., Johnston, S., Kravec, S., Showk, S. E., Lanham, T., Telleen-Lawton, T., Brown, T. B., Henighan, T., Hume, T., Bai, Y., Hatfield-Dodds, Z., Clark, J., Bowman, S., Askell, A., Grosse, R. C., Hernandez, D., Ganguli, D., Hubinger, E., Schiefer, N., and Kaplan, J. Discovering language model behaviors with model-written evaluations. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:254854519.

Qian, K., Wan, S., Tang, C., Wang, Y., Zhang, X., Chen, M., and Yu, Z. Varbench: Robust language model benchmarking through dynamic variable perturbation. *ArXiv*, abs/2406.17681, 2024. URL https://api.semanticscholar.org/CorpusID:270711329.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., et al. Toolllm: Facilitating

large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Rawles, C., Clinckemaillie, S., Chang, Y., Waltz, J., Lau, G., Fair, M., Li, A., Bishop, W., Li, W., Campbell-Ajala, F., et al. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024.

Ribeiro, M. T., Wu, T. S., Guestrin, C., and Singh, S. Beyond accuracy: Behavioral testing of nlp models with checklist. *ArXiv*, abs/2005.04118, 2020. URL https://api.semanticscholar.org/CorpusID:218551201.

Sainz, O., Campos, J. A., García-Ferrero, I., Etxaniz, J., de Lacalle, O. L., and Agirre, E. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:264555419.

Schmidgall, S., Ziaei, R., Harris, C., Reis, E., Jopling, J., and Moor, M. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv preprint arXiv:2405.07960*, 2024.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.

Trivedi, H., Khot, T., Hartmann, M., Manku, R., Dong, V., Li, E., Gupta, S., Sabharwal, A., and Balasubramanian, N. 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.

Wang, J., Hu, Y., Yang, W., Pan, Z., Li, X., and Guo, L.-Z. Aligning agents via planning: A benchmark for trajectory-level reward modeling. *arXiv preprint arXiv:2604.08178*, 2026.

Wang, W., Zhang, D., Feng, T., Wang, B., and Tang, J. Battleagentbench: A benchmark for evaluating cooperation and competition capabilities of language models in multi-agent systems. *arXiv preprint arXiv:2408.15971*, 2024.

Wang, X., Wang, Z., Liu, J., Chen, Y., Yuan, L., Peng, H., and Ji, H. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*, 2023.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Wei, J., Sun, Z., Papay, S., McKinney, S., Han, J., Fulford, I., Chung, H. W., Passos, A. T., Fedus, W., and Glaese, A. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Dey, S., Shubh-Agrawal, Sandha, S. S., Naidu, S., Hegde, C., LeCun, Y., Goldstein, T., Neiswanger, W., and Goldblum, M. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations*, 2024. URL https://api.semanticscholar.org/CorpusID:277955704.

Xi, Z., Ding, Y., Chen, W., Hong, B., Guo, H., Wang, J., Guo, X., Yang, D., Liao, C., He, W., et al. Agentgym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27914–27961, 2025.

Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T. J., Cheng, Z., Shin, D., Lei, F., et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Xu, C., Guan, S., Greene, D., and Kechadi, M.-T. Benchmark data contamination of large language models: A survey. *ArXiv*, abs/2406.04244, 2024. URL https://api.semanticscholar.org/CorpusID:270285708.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K. A., Yao, S., Narasimhan, K., and Press, O. Swe-agent: Agent-computer interfaces enable automated software engineering. *ArXiv*, abs/2405.15793, 2024. URL https://api.semanticscholar.org/CorpusID:270063685.

Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Ye, S., Shi, H., Shih, D., Yun, H., Roosta, T. G., and Shu, T. Realwebassist: A benchmark for long-horizon web assistance with real-world users. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34441–34449, 2026.

Ying, J., Cao, Y., Bai, Y., Sun, Q., Wang, B., Tang, W., Ding, Z., Yang, Y., Huang, X., and Yan, S. Automating dataset updates towards reliable and timely evaluation of large language models. *Advances in Neural Information Processing Systems 37*, 2024. URL https://api.semanticscholar.org/CorpusID:267750054.

Ying, L., Truong, R., Sharma, P., Zhao, K. I., Cloos, N., Allen, K. R., Griffiths, T. L., Collins, K. M., Hernández-Orallo, J., Isola, P., et al. Ai gamestore: Scalable, open-ended evaluation of machine general intelligence with human games. *arXiv preprint arXiv:2602.17594*, 2026.

Yue, B., Zhu, Z., Zhang, Y., Feng, J., Yang, H., and Wang, M. Interactive benchmarks. *arXiv preprint arXiv:2603.04737*, 2026.

Zhang, D., Shen, Z., Xie, R., Zhang, S., Xie, T., Zhao, Z., Chen, S., Chen, L., Xu, H., Cao, R., et al. Mobile-env: Building qualified evaluation benchmarks for llm-gui interaction. *arXiv preprint arXiv:2305.08144*, 2023.

Zhang, Z., Cui, S., Lu, Y., Zhou, J., Yang, J., Wang, H., and Huang, M. Agent-safetybench: Evaluating the safety of llm agents. *arXiv preprint arXiv:2412.14470*, 2024.

Zhao, Q., Huang, Y., Lv, T., Cui, L., Sun, Q., Mao, S., Zhang, X., Xin, Y., Yin, Q., Li, S., and Wei, F. Mmlu-cf: A contamination-free multi-task language understanding benchmark. *ArXiv*, abs/2412.15194, 2024. URL https://api.semanticscholar.org/CorpusID:274859647.

Zhou, L., Pacchiardi, L., Martínez-Plumed, F., Collins, K. M., Moros-Daval, Y., Zhang, S., Zhao, Q., Huang, Y., Sun, L., Prunty, J. E., et al. General scales unlock ai evaluation with explanatory and predictive power. *Nature*, 652(8108):58–67, 2026.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023a.

Zhou, X., Zhu, H., Mathur, L., Zhang, R., Yu, H., Qi, Z., Morency, L.-P., Bisk, Y., Fried, D., Neubig, G., et al. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*, 2023b.

Zhou, X., Kim, H., Brahman, F., Jiang, L., Zhu, H., Lu, X., Xu, F., Lin, B. Y., Choi, Y., Mireshghallah, N., et al. Haicosystem: An ecosystem for sandboxing safety risks in human-ai interactions. *arXiv preprint arXiv:2409.16427*, 2024.

Zhu, H., Zhang, Q., Wang, J., Yang, Z., and Qiu, Y. Needle in the repo: A benchmark for maintainability in ai-generated repository edits, 2026. URL https://arxiv.org/abs/2603.27745.

Zhu, K., Chen, J., Wang, J., Gong, N. Z., Yang, D., and Xie, X. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *International Conference on Learning Representations*, 2023. URL https://api.semanticscholar.org/CorpusID:263310319.

Zhu, K., Du, H., Hong, Z., Yang, X., Guo, S., Wang, D. Z., Wang, Z., Qian, C., Tang, R., Ji, H., et al. Multiagentbench: Evaluating the collaboration and competition of llm agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622, 2025.

Zhu, Q., Cheng, Q., Peng, R., Li, X., Liu, T., Peng, R., Qiu, X., and Huang, X. Inference-time decontamination: Reusing leaked benchmarks for large language model evaluation. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:270619707.

A. More Alternative Views

This paper argues that interactive evaluation requires a design science for evaluating systems acting through trajectories. As we clarified throughout our main paper, this position does not imply that all evaluations should become interactive, that response-centered benchmarks should be discarded, or that every recorded trajectory should be converted into a single process score. Below we discuss several viable alternative views and clarify how our position accommodates or responds to them.

Alternative view 1: Interactive evaluation is not distinct from dynamic, live, executable, or holistic evaluation.

One might argue that the field already has many extensions of traditional benchmarking: live benchmarks, contamination-resistant benchmarks, dynamic datasets, executable coding tasks, tool-use benchmarks, holistic evaluation suites, and continuously refreshed evaluations. From this view, interactive evaluation may appear to rename existing trends rather than identify a distinct evaluation paradigm.

Response. These efforts are closely related and often complementary, but they are not identical to interactive evaluation as defined in this paper. The distinction is not whether a benchmark is difficult, fresh, realistic, executable, or multi-step. The distinction is whether the admissible evidence includes trajectories generated by consequential interaction, and whether the evaluation program maps those trajectories to judgments about system-level performance.

A live benchmark can remain response-centered if it simply refreshes fixed instances and scores final answers. An executable benchmark can remain response-centered if execution is used only to check a final submitted artifact. A tool-use benchmark may be interactive if tool results change subsequent observations and the trajectory matters to the judgment, but tool use alone is not sufficient. Likewise, a holistic evaluation suite may cover many capabilities without evaluating action-dependent trajectories. Interactive evaluation is therefore defined by action dependence and trajectory-to-judgment mapping, not by novelty, realism, or breadth.

This boundary matters because different benchmark families support different claims. Dynamic and live benchmarks address contamination and temporal freshness. Executable benchmarks improve verifiability. Holistic suites broaden coverage. Interactive evaluation addresses a different question: how should we evaluate systems whose behavior unfolds through state, feedback, tools, users, or other agents, such that earlier actions shape later evidence and the meaning of success? Our position is that this question requires explicit design principles rather than being absorbed into existing benchmark categories.

Alternative view 2: Interactive evaluation is too expensive and should be used sparingly.

A reasonable objection is that interactive evaluation can increase the cost of benchmarking. Long trajectories, executable environments, state snapshots, user simulators, counterpart agents, repeated runs, and trajectory-level judging can all make evaluation harder to run and harder to audit. If interactive evaluation becomes the default benchmark format, it may privilege large labs and reduce the accessibility that made response-centered benchmarks scientifically valuable.

Response. We agree that interactive evaluation should not be used for everything. The right criterion is not whether an evaluation can be made interactive, but whether the capability claim requires interaction as evidence. If the relevant evidence is already contained in a fixed input and final output, then a response-centered evaluation may be the better design: cheaper, cleaner, and easier to reproduce. Our position is instead claim-dependent. Interactive evaluation is needed when earlier system behavior changes later observations, states, opportunities, costs, risks, or counterpart responses, and when those changes matter to the judgment being made.

At the same time, the cost objection should not be overstated. Many apparently response-centered evaluations already require intermediate artifacts: generated code, tool calls, retrieved documents, execution traces, plans, test logs, edited files, or state changes. When such artifacts are already produced as part of evaluation, trajectory-level evaluation may not require a wholly new evaluation regime. It may instead require using evidence that the benchmark already creates but currently throws away. In these cases, interactive evaluation can improve evidential efficiency: the same evaluation run can support not only a final success judgment, but also claims about recovery, constraint satisfaction, action economy, unsafe operations, or robustness. The design challenge is therefore to add trajectory-level judgments only where they provide additional validity relative to their cost.

Alternative view 3: Response-centered evaluation remains sufficient for most purposes.

Another view is that response-centered evaluation has been successful because it captures the most important comparison

signal. Fixed inputs, standardized outputs, and simple scoring support clear leaderboards, scalable experimentation, and cumulative progress. From this perspective, interactive evaluation risks replacing a robust paradigm with a more complex one whose benefits are uncertain.

Response. We do not argue that response-centered evaluation should be discarded. On the contrary, it remains indispensable when the object of evaluation is naturally an input-output mapping, when final correctness is the relevant claim, or when interaction is incidental rather than consequential. The boundary is evaluative, not stylistic. Multiple turns, tool calls, or generated intermediate text do not by themselves require interactive evaluation. They require interactive evaluation only when the trajectory changes what evidence is available and what judgment the score can support.

The limitation of response-centered evaluation appears when final responses are no longer sufficient evidence. A system that reaches the correct final state by corrupting persistent state, ignoring user constraints, manipulating a counterpart, overusing tools, or failing to recover from induced errors has not demonstrated the same capability as a system that reaches the same outcome safely and robustly. In such cases, final success remains important, but it is incomplete. Interactive evaluation supplements response-centered evaluation exactly where process, state, feedback, and consequence change the meaning of success.

Alternative view 4: Task success is the only robust metric; process metrics are subjective.

A strong objection is that final task success has clearer face validity than process-level measurement. By contrast, metrics for process quality, communication, risk, recovery, or efficiency may encode contestable assumptions. If benchmarks assign scores to trajectories, they may reduce comparability or give a false sense of objectivity to normative choices about what counts as a good process.

Response. This objection identifies a real danger, but it supports our design-science view rather than undermining it. The problem is not that process metrics involve choices; all evaluation metrics involve choices. The problem is when those choices are hidden. Outcome-only metrics can appear neutral while silently permitting unsafe, brittle, manipulative, or wasteful behavior whenever such behavior still achieves the final goal. A benchmark that scores only success is not free of process assumptions; it implicitly assumes that all successful trajectories are equivalent for the intended claim.

The appropriate response is not to collapse outcome, process, and risk into a single opaque score. Interactive evaluations should report these dimensions separately whenever they support distinct claims. Final success should remain visible. Process measures should be tied, where possible, to externally checkable trajectory properties: state changes, action counts, constraint violations, failed tool calls, recovery after perturbation, rollback behavior, unsafe operations, communication failures, or robustness across seeds and counterpart variants. Aggregate scores may still be useful for ranking, but they should be treated as summaries of reported dimensions, not as substitutes for them. This is why interactive evaluation requires explicit evaluation programs rather than ad hoc trajectory judgment.

Alternative view 5: Trajectory-level evaluation may reward performative behavior rather than capability.

If benchmarks judge trajectories, systems may learn to produce trajectories that look good to humans or model judges. They may generate verbose plans, artificial self-corrections, or judge-pleasing rationales without improving actual competence. In the worst case, trajectory scoring could become another surface to game.

Response. This is an important trajectory-level analogue of benchmark gaming. It is one reason we distinguish trajectory evidence from exposed reasoning style. Interactive evaluation should not reward a trajectory because it appears thoughtful, verbose, or human-like. It should reward trajectory properties that are relevant to the capability claim and, when possible, grounded in observable interaction outcomes.

For example, recovery should be tested by whether a system detects and repairs an actual induced failure, not by whether it says that it is being careful. Efficiency should be measured by action cost, redundant tool use, or unnecessary state changes, not by whether the transcript appears concise. Safety should be measured through constraint violations, unsafe operations, or harmful state changes, not merely by safety-themed language. This shifts trajectory evaluation away from stylistic judgment and toward behavioral evidence. It also reinforces the need for perturbations, audits of suspiciously efficient trajectories, replayable logs, and transparent reporting of what the evaluator actually scores.

Alternative view 6: Interactive evaluation conflates model capability with system engineering.

A further objection is that interactive benchmarks often evaluate more than the base model. Tool wrappers, memory, retrieval systems, planners, sandboxes, interface affordances, orchestration policies, and prompting strategies may dominate

performance. If so, interactive evaluation may make it difficult to know whether progress comes from better models or better systems engineering.

Response. We agree that interactive evaluation often evaluates systems rather than isolated models. This is not a defect of the paradigm; it is a property of the deployment settings that motivate it. When an AI system acts through tools, environments, users, memory, or other agents, the relevant object of evaluation is frequently the assembled system. A benchmark that ignores wrappers, permissions, state, orchestration, or tool interfaces may provide a cleaner model-level comparison, but it may not support claims about deployed interactive behavior.

The implication is that benchmark reports must distinguish model-level and system-level claims. If the goal is to compare base models, then the surrounding scaffold should be controlled and reported. If the goal is to compare complete agents or deployed assistants, then the scaffold is part of the evaluated object and should be documented as such. Interactive evaluation therefore raises the standard for reporting: model identity alone is insufficient. Evaluations should specify tools, memory, retrieval, prompts, orchestration, sandbox permissions, environment versions, and logging protocols so that readers can interpret what the score is actually evidence about.

B. Related Works

As is the nature of a position paper, we discuss the most closely related benchmarks and evaluation paradigms in the main text when defining interactive evaluation and constructing the taxonomy. This appendix notes adjacent areas that inform the position but are not central to the main argument.

Detection & Mitigation of Response-Centered Evaluation Issues. A large body of prior work studies limitations of response-centered evaluation, including benchmark leakage (Jacovi et al., 2023; Balloccu et al., 2024), data contamination (Sainz et al., 2023; Xu et al., 2024), benchmark overfitting (McCoy et al., 2019), brittleness under distribution shift (Ailem et al., 2024), and the use of private, hidden, or live evaluation to reduce memorization and gaming (Kiela et al., 2021; White et al., 2024). To address these limitations, researchers have proposed contamination audits (Golchin & Surdeanu, 2023; Deng et al., 2023a), benchmark decontamination (Zhu et al., 2024), adversarial or counterfactually constructed examples (Ribeiro et al., 2020; Kiela et al., 2021), dataset refreshment (Ying et al., 2024), live evaluation (Chandran et al., 2024), more explicit reporting of evaluation conditions (Liang et al., 2023; Jacovi et al., 2023), and meta-annotation schemes to understand model performance across multiple benchmarks (Zhou et al., 2026). These efforts have been essential for clarifying when fixed-instance benchmark scores provide reliable evidence of model capability. These concerns motivate our risk analysis, but our focus is different: Here we ask how these problems change when the evaluated system acts through consequential trajectories rather than producing isolated responses, and what additional design requirements this creates for interactive evaluation.

Other Extensions to Traditional Benchmarks. Traditional benchmarks have also been extended in several directions beyond the response-centered format. Live and continuously refreshed benchmarks update evaluation instances over time to reduce memorization and better reflect current model capabilities (White et al., 2024; Jain et al., 2024; Kasai et al., 2022). Dynamic and procedurally generated benchmarks vary tasks, constraints, or environments to reduce benchmark-specific overfitting (Zhu et al., 2023; Qian et al., 2024; Ying et al., 2026). Private or hidden benchmarks restrict access to test instances to limit leakage and gaming (Oren et al., 2023; Zhao et al., 2024). Other benchmarks add execution, tools, browsing, long-context inputs, or real-world task constraints so that model outputs can be checked against tests, external evidence, or richer task specifications. These efforts are complementary, but they mainly improve how benchmark instances are sourced, protected, refreshed, or verified. Our focus is action-dependent evaluation: settings where earlier system behavior shapes later observations, states, opportunities, or counterpart responses. The central question is therefore not only how to make tasks fresher or harder, but how trajectory evidence should be mapped to claims about system-level performance.

C. Additional Details for the Roadmap

C.1. Roadmap Categories.

To examine the temporal pattern behind Figure 1, we use three roadmap categories. **Response-centered evaluation** refers to benchmarks where the main evidence is a final response to a fixed instance. **Task-driven extensions** refer to benchmarks

that add execution, tools, web access, or multi-step task completion, but remain primarily organized around fixed task outcomes rather than consequential trajectory evidence. **Interactive evaluation** refers to benchmarks where trajectories generated by consequential interaction enter the admissible evidence and are mapped to judgments about system-level performance.

C.2. Benchmark Collection.

To analyze temporal trends across the three roadmap categories, the representative benchmark list alone is not sufficient. We therefore expand it through two semi-automated retrieval channels. First, we perform citation-based snowball sampling from the representative benchmarks and retain citing papers whose titles match benchmark-related keywords, such as "bench," "arena," and "gym." Second, snowball sampling can miss early-year work that predates or coincides with our anchors. We therefore run stage-neutral Semantic Scholar searches for each year from 2020 to 2026.

We then deduplicate papers across channels using arXiv IDs when available and normalized titles otherwise, and apply a shared quality filter to obtain the final candidate set. A paper is retained if it appears in a top venue, or has citation velocity at least 1.5, or has at least 50 GitHub stars. We define citation velocity as

$$\mathrm{CitationVelocity}(p) = \frac{\mathrm{Citations}(p)}{\max(\mathrm{MonthsSincePublication}(p),\ 3)}.$$

All candidate papers are then classified by an LLM-based classifier into the three roadmap categories or *Not Relevant*; only papers assigned to the three roadmap categories are included in the trend analysis. Classification uses each paper's title and abstract, follows the roadmap definitions above, and is based on the paper's primary contribution rather than whether it merely evaluates a new model on existing benchmarks. We validate the classifier on the manually curated anchor set and apply it to the expanded corpus only after it achieves over 90% agreement with our manual labels. For ambiguous cases, the classifier assigns Interactive Evaluation only when the paper explicitly emphasizes trajectory evidence, stateful interaction, or agent feedback loops. We use this analysis as descriptive evidence for broad temporal trends, rather than as an exhaustive census of all evaluation work.

C.3. Industry-Academic Comparison.

Panel (c) of Figure 1 compares evaluation-stage composition between recent frontier industry reports and academic benchmark papers from 2024–2026. The industry sample contains 43 distinct benchmark families extracted from the most recent public model cards or technical reports of OpenAI, Anthropic, Google DeepMind, and Alibaba/Qwen, with each benchmark family counted once per source document regardless of variants or subtasks. The academic sample is the 2024–2026 subset of the benchmark collection described above, containing 360 benchmark papers. Bars report percentage shares within each group, so each group sums to 100%. A Pearson χ^2 test gives $\chi^2(2) = 7.09, p = 0.029$, indicating a statistically significant difference in stage distribution. We interpret this comparison as descriptive evidence that the transition toward task-driven and interactive evaluation is uneven across the evaluation ecosystem, rather than as an exhaustive census of either community.

D. Illustrative Scenarios

This section provides two illustrative scenarios showing how the proposed framework can be applied end to end. The goal is to make concrete how interactive evaluation changes both sides of the evaluation mapping $E : \mathcal{X} \to \mathcal{Y}$: what trajectory evidence enters $\mathcal{X}$, and what evaluation program E is needed to turn that evidence into valid claims.

We choose these two scenarios because they represent two broad classes of settings where interactive evaluation is especially valuable. Coding-agent evaluation illustrates domains involving multi-step reasoning, iterative planning, execution, debugging, and revision, where the trajectory itself contains important evidence about process quality, recoverability, and robustness beyond final task completion. Multi-agent social evaluation illustrates settings in which simulation, interaction, and adaptive environments are necessary for probing coordination, negotiation, communication, and other socially situated behaviors that cannot be captured through isolated responses alone. While the examples are concrete, the underlying design principles generalize more broadly across many other interactive evaluation settings within each category.

D.1. Coding Agents.

Mapping to the Definition. Coding-agent evaluation naturally benefits from interactive evaluation as the system acting in a repository-level feedback loop. The agent works through repository-level actions: inspecting files, running commands, observing test failures or error traces, editing code, and revising its solution based on feedback (Yang et al., 2024). Because these actions shape later observations and repair opportunities, the relevant evidence is the full interaction trajectory instead of single final patch.

From Response-Centered to Interactive Evaluation. Traditional code-generation benchmarks evaluate a final output against references or tests (Hendrycks et al., 2021; Austin et al., 2021), but repository-level agents raise broader claims. Two agents may both pass hidden tests, yet one may rely on principled diagnosis and localized edits while another relies on brittle edits or visible-test overfitting. A final pass/fail label treats these trajectories as equivalent, even though they support different claims about debugging competence, maintainability, and deployment reliability.

Mapping to the Taxonomy. In our taxonomy, coding-agent evaluation mainly belongs to Tools and Environments because the agent interacts with repositories, command-line tools, test suites, and executable environments. The input artifact $\mathcal{X}$ should include repository state, issue text, tool calls, file edits, test executions, error traces, and final patch. On the evaluation-program axis, coding agents should require not only Task Success, but also Process Quality and Efficiency, Recoverability and Robustness, and risk-sensitive evaluation for transparent failure diagnosis.

Applying the Design Principles. The evaluation program E should distinguish whether the issue is resolved, whether the patch is localized and maintainable (Zhu et al., 2026), whether the agent uses tests and errors to recover, and whether it avoids collateral damage. SWE-bench and recent coding-agent benchmarks already evaluate agents in repository-level or long-horizon programming settings (Jimenez et al., 2023; Khatua et al., 2026; Feng et al., 2026). However, they still often reduce repository-level interaction to final resolution, leaving diagnosis quality, recovery behavior, patch locality, and collateral risk under-specified. Concretely, coding-agent benchmarks should specify tool access, test access, retry policy, repository reset conditions, and logging format, and should report final resolution separately from trajectory-level measures.

Risks and Open Issues. Coding-agent evaluation also exposes trajectory-level risks: agents may game visible tests, exploit benchmark-specific repository patterns, or produce patches that pass current tests while introducing hidden regressions. Scores can also be sensitive to environment setup, dependency versions, tool access, timeout limits, and retry policies. Future evaluations therefore need replayable trajectory logs, environment versioning, and reporting standards that distinguish genuine debugging competence from benchmark-specific exploitation.

D.2. Multi-Agent Social Systems.

Mapping to the Definition. Multi-agent social evaluation is an interactive setting because the system acts in a social feedback loop with other agents whose beliefs, strategies, and future behavior can change in response to its actions (Davidson et al., 2024; Abdelnabi et al., 2023). The evaluated agent does not merely produce an isolated utterance; it communicates, negotiates, coordinates, refuses, adapts to counterpart behavior, and may adjust its strategy as the interaction unfolds. These actions shape subsequent messages, commitments, conflicts, and opportunities for coordination, so the relevant evaluation evidence is the full social trajectory rather than only the final group outcome.

From Response-Centered to Interactive Evaluation. Response-centered evaluation can assess an isolated utterance, but social agents raise broader claims about coordination, fairness, communication, and robustness. For example, a negotiation agent may make a reasonable and clearly phrased offer in one turn, while still exploiting counterpart concessions, treating stronger and weaker counterparts differently, or breaking down under unusual communication styles. These properties require evidence from the full interaction trajectory, not only from an isolated response.

Mapping to the Taxonomy. In our taxonomy, multi-agent social evaluation mainly belongs to Other Agents because the system interacts with counterparts whose goals, information, and behavior may change over time. The input artifact $\mathcal{X}$ should include messages, role assignments, private and shared information, commitments, proposals, refusals, conflict points, counterpart behavior, and final outcomes. On the evaluation-program axis, social agents require not only Task Success, but also Safety, Alignment, and Social Competence, as well as Recoverability and Robustness when agents must repair misunderstanding or adapt to strategic counterparts.

Applying the Design Principles. The evaluation program E should distinguish whether the social goal is completed, whether agents coordinate effectively, or whether communication is fair and transparent. Benchmarks such as SOTOPIA, MultiAgentBench, BattleAgentBench, MASEval, and CooperBench already evaluate agents in cooperative, competitive, or mixed-motive social settings (Zhou et al., 2023b; Zhu et al., 2025; Wang et al., 2024; Emde et al., 2026; Khatua et al., 2026). However, they still often reduce social interaction to aggregate success or judge-level preferences, leaving coordination process, fairness, recovery from misunderstanding, and social risk under-specified. Concretely, multi-agent benchmarks should specify role assignments, information asymmetries, counterpart behavior, memory, turn-taking rules, stopping criteria, and hidden goals or constraints, and should report group success separately from trajectory-level measures.

Risks and Open Issues. Multi-agent social evaluation introduces interaction-specific risks. An agent may appear cooperative with one counterpart but become exploitative, evasive, or overly deferential with another; small changes in roles, private information, or power asymmetry can change what competence means (Lewis et al., 2017). Because judgments depend on norms, incentives, and counterpart behavior, scores may conflate genuine coordination with persuasion, pressure, or strategic withholding (Manheim & Garrabrant, 2018; Perez et al., 2022). Evaluations should make roles, incentives, norms, and judge criteria explicit, and report whether success reflects robust social competence or undesirable social strategies.

E. Representative Benchmark List

This section reports the representative benchmark list (Fig. 1) and the taxonomy discussion (Fig. 3) in the main paper. The list is intended as a transparent resource for the roadmap analysis rather than an exhaustive census of all evaluation benchmarks. We record metadata such as citation counts and GitHub stars as accurately as possible at the time of collection[1]. However, note that these values should be interpreted only as approximate indicators rather than definitive measures of benchmark importance or influence. In few cases, GitHub star counts may be unavailable for older benchmarks or for works without an official public repository, in which case we leave the field blank rather than infer a value. Some other benchmarks created Github repositories substantially later than their release dates, which may also make star counts unevenly comparable across works. Similarly, citation counts can reflect many factors beyond evaluative significance, including publication venue, age, community size, and research trends. We therefore include these metrics primarily as suggestive signals that help visualize the current evaluation landscape, while aiming to make our benchmark selection and categorization process easier to inspect, reproduce, and refine in future work, rather than presenting them as authoritative rankings of benchmark quality or impact.

Benchmark	Year	Task Type	Citations	Stars
Stage 1: Response-Centered Evaluation				
GLUE	2018	Reading Comprehension	10,589	—
DROP	2019	Reading Comprehension	1,438	—
CommonsenseQA	2019	Commonsense Reasoning	2,666	168
MMLU	2020	Knowledge & Multitask Reasoning	7,833	1.4k
GSM8k	2021	Math Reasoning	8,894	1.4k
MATH	2021	Math Reasoning	5,004	1.3k
MBPP	2021	Code Generation	3,556	37.8k
HumanEval	2021	Code Generation	9,594	3.2k
Big-bench	2022	Broad Capability Probing	2,560	3.2k
TruthfulQA	2022	Truthfulness & Factuality	3,638	910
LongBench	2023	Long-Context Understanding	1,191	1.1k
Chatbot Arena	2024	Human Preference Evaluation	1,196	39.5k
LoCoMo	2024	Long-term Memory	390	828
Omni-Math	2024	Math Reasoning	175	93
LongMemEval	2025	Long-term Memory	234	738
Stage 2: Task-Driven Extensions				

(continued on next page)

[1]The data in this table were last updated on May 7, 2026.

(continued from previous page)

Benchmark	Year	Task Type	Citations	Stars
SWE-bench	2023	Code & Software Engineering	2,397	4.8k
API-Bank	2023	Tool Use & API Calling	502	1.6k
Mind2Web	2023	Web Navigation	1,113	986
ToolBench	2023	Tool Use & API Calling	1,451	5.6k
TaskBench	2023	Task Automation & Planning	142	24.7k
LiveCodeBench	2024	Code Generation & Execution	1,384	857
Stabletoolbench	2024	Tool Use & API Calling	178	232
Travelplanner	2024	Planning & Constraint Satisfaction	372	510
OSS-Bench	2025	Code & Software Engineering	3	8
MM-BrowseComp	2025	Web Navigation	21	67
BrowseComp	2025	Web Navigation	364	4.4k
DeepPlanning	2026	Planning & Constraint Satisfaction	10	16.3k
Terminal-Bench	2026	Code & Software Engineering	80	2.2k
Longcli-bench	2026	Code & Software Engineering	5	33
Stage 3: Interactive Evaluation				
AppWorld	2024	App / Software Use	146	413
AndroidWorld	2024	Web / Computer Use	267	752
τ-bench	2024	Tool / Assistant	496	1.2k
VisualWebArena	2024	Web / Computer Use	527	466
OSWorld	2024	Web / Computer Use	616	2.8k
WebArena	2024	Web / Computer Use	1,174	1.5k
Sotopia	2024	Social Interaction	391	300
UserBench	2025	Tool / Assistant	33	60
Agent-SafetyBench	2025	Tool / Assistant	157	134
ToolSandbox	2025	Tool / Assistant	130	247
Multi-agent Bench	2025	Social Interaction	152	46
SimWorld	2025	Embodied / Open-World	3	561
ARE (GAIA2)	2025	App / Software Use	16	482
RealWebAssist	2025	Web / Computer Use	17	10
BuilderBench	2026	Embodied / Open-World	2	32