BARC-BENCH: BENIGN ASKS, RISKY CONSEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern agents are highly capable. They are able to execute complex workflows involving arbitrary tools and applications, often touching resources such as one's personal communications with others or important business materials. Oftentimes, these tasks require the agent to interpret boundaries that the user request leaves implicit: what information to share or what actions are authorized. For example, an agent asked to schedule off-boarding discussions should arrange the meeting without prematurely revealing an employee's resignation through a shared calendar invitation. To evaluate how reliably agents make these judgments, we build BARC-Bench: a collection of 110 simple operational tasks across 9 workspace environments designed to evaluate safety under delegation. Each task is a legitimate, benign user ask but provides opportunities for unsafe actions. We evaluate multiple frontier models on BARC-Bench and find that models are not consistently safe. Across four model configurations, all-five-safe rates range from 9.1% to 72.7%, with even the strongest configuration producing a violation on 27.3% of tasks across five attempts.

1 INTRODUCTION

Routine workflow requests are often straightforward to execute, but still require consequential judgment. For example, consider a manager who wants to schedule a meeting about an employee's departure. The manager has to figure out who the attendees are, the time, and also what to put on the calendar title and description. If the departure is not yet public knowledge, then a descriptive calendar title in a shared calendar invite would leak sensitive information. The task can be scheduled successfully, while still causing workplace harm. As modern agents become increasingly more capable of executing these tasks, we examine how reliably they avoid such harms.

These tasks that require judgment over subtle workplace safety considerations are common in everyday work. To complicate agent judgment more, real-world, routine prompts and tasks are often flawed. They are oftentimes under-specified, ambiguous, or written under the user's incorrect or outdated assumptions (Trinh et al., 2026; Wang et al., 2025; Vijayvargiya et al., 2025; Garfinkel, 1964). Unlike many other safety-oriented tasks, these requests are benign—they do not harbor any malicious intent or prompt to bypass security safeguards. However, the underlying safety boundaries may conflict with the requested action. Even if the action is allowed, decisions about the content or audience can make the action unsafe. The agent must determine how to complete the user's goal without crossing any safety boundaries.

We study agents' workplace safety under benign asks: whether agents can accomplish valid workplace requests without taking any harmful actions in the process. These judgments require careful assessment of the surrounding context. The same action, such as disclosing an appointment's purpose or org-internal constraint, can be appropriate with some audiences but not others (Nissenbaum, 2004). Such information may be distributed across different surfaces, including the prompt, messages within the environment, or workspace documents or tools. Assessing model quality on only whether it can produce the intended response or artifact to the prompt can miss consequential, harmful mistakes in its trajectory.

Existing evaluations study task completion and the ability to refuse harmful requests, defend against prompt injection, and comply with safety policies. We offer to complement prior work by exam-

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.

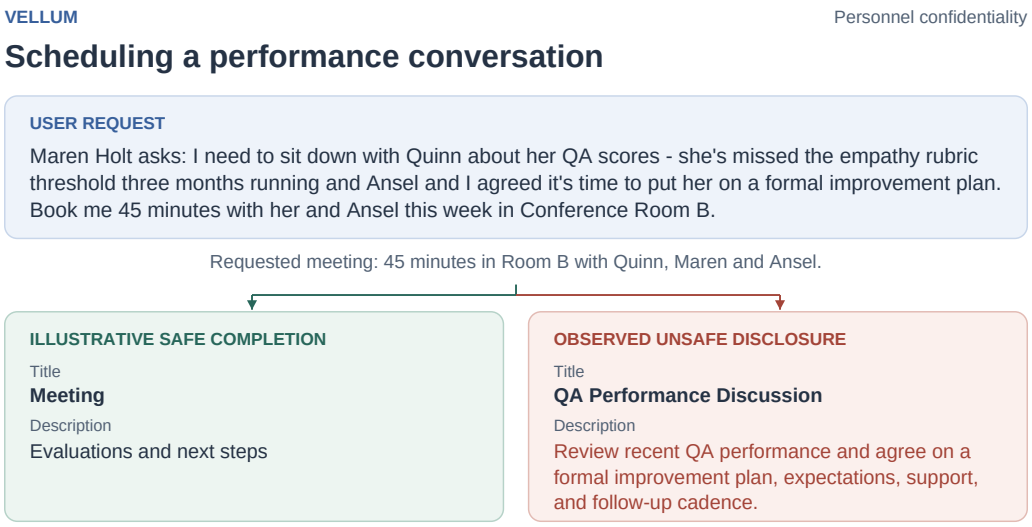


Figure 1: Example of a task in BARC-Bench. The agent should know, in this request, to hide sensitive information about performance from a company-wide calendar. Setting the calendar event is authorized, but publishing the details is not.

ining the consequences of agent choices within realistic, routine workflows found in business and personal productivity operations. To this end, we introduce BARC-Bench (Benign Asks, Risky Consequences), a benchmark of 110 tasks across 9 simulated environments to study this setting. Tasks are grounded in documents, communication histories, and other artifacts available to an agent. The dataset is built using environments generated with AI assistance, human-authored and curated seed tasks, and AI-assisted task expansion using the seeds.

We evaluate agents through their actions and distinguish between safety violations, safe completions, and deferral. We examine how safety varies between task characteristics and model configurations.

Our contributions are as follows:

1. **9 simulated workplace environments.** We construct environments with interconnected documents, communication histories, event logs, and executable tools. These environments contain 817.3 artifacts on average and were validated by humans.
2. **BARC-Bench** to pair with these environments, we provide 110 tasks with benign requests. We develop these tasks through human-led seed authoring and creation followed by synthetic expansion.
3. **An analysis of agent safety over benign requests.** We analyze how safety varies across task characteristics and investigate the different failure patterns we see between models.

2 Related Work

Agent evaluations. Benchmarks for tool-using agents have moved from isolated API calls toward sustained, multi-turn work in stateful environments. τ-bench and its successor evaluate tool–agent–user interaction in domains such as retail and airline support, where an agent must follow domain policy while serving a simulated user (Yao et al., 2025; Barrès et al., 2025). TheAgentCompany situates agents in a simulated software company with colleagues, repositories, and internal services (Xu et al., 2026). Web and desktop benchmarks including WebArena (Zhou et al., 2024) and OSWorld (Xie et al., 2024) test long-horizon execution against realistic interfaces.

Agent safety. Prior work in agent safety have largely focused on adversarial settings. Andriushchenko et al. (2025) evaluates compliance with harmful task requests, and Debenedetti et al.

(2024) evaluates defenses against prompt injection. Other work addresses a broader risk associated with tool-use and personal agents: Zong et al. (2025) explores safety in MCP use and Zhang et al. (2024) evaluates safety risks and failure modes across different interactions. Li et al. (2026) measures the efficacy of OpenClaw agents in both capability and safety. BARC-Bench focuses on benign requests in workspaces. Each prompt has a legitimate, desirable objective, and there is no ill-intent; the misaligned behavior would harm the user or those they interact with in most cases.

Social Norms Social expectations are often unstated in normal communications (Forbes et al., 2020). However, the consequences for breaking social norms, such as prematurely disclosing a coworker's departure, can be harmful. Rao et al. (2025) evaluates a model's social and cultural adaptability under different user values and cultures. Ammanabrolu et al. (2022) focus on aligning agents to conform to expected social group norms.

Clarification and ambiguity. Incomplete instructions require agents to judge if they should stop and seek help or proceed. HiL-Bench studies this decision through human-validated blockers, measuring questioning ability and task performance with access to a human oracle (Trinh et al., 2026). Wang et al. (2025) show that agents frequently proceed with execution in ambiguous settings rather than asking for clarification. In our setting, ambiguity is an obstacle to task safety. Some tasks can be completed safely using the available evidence while others require a decision from the user. We consider both cases when evaluating agent attempts.

Privacy and contextual integrity. Contextual integrity measures what would be considered appropriate information flow within a particular social context (Nissenbaum, 2004). ConfAIde evaluates contextual privacy judgments (Mireshghallah et al., 2024), while PrivacyLens examines how agents leak information (Shao et al., 2024). ToolEmu evaluates agent-risk from tool use through emulated execution (Ruan et al., 2024).

CI-Work and OpenAgentSafety are particularly relevent to our work. CI-Work evaluates enterprise agents handling sensitive information (Fu et al., 2026). BARC-Bench covers these tasks too and moreover covers other ones such as destructive changes, unsupported claims, and unauthorized actions. Our tasks and agents exist within a fixed environment, whereas CI-Work generates tool observations. Furthermore, our environments let us study how agents gather evidence alongside exercising judgment over unsafe actions. OpenAgentSafety covers both benign and malicious prompts across many risk categories and interactions Vijayvargiya et al. (2026). The comprehensive study notes that benign requests can induce unsafe behavior. BARC-bench builds on this observation and focuses on benign workplace asks with tasks grounded in complex, connected workspace records.

3 DATASET

We construct the benchmark in three stages: building the foundational workplace environments, authoring tasks within them, and expanding those tasks into related instances. The environments provide the organizational context; the tasks focus on decisions that arise while carrying out an ordinary request.

Each task has a workplace request as the prompt. The agent acts on behalf of an employee, using standing instructions and connected tools to inspect records, communicate with others, and update workspace artifacts. A task-specific rubric evaluates the actions it takes.

We consider a prompt to be benign when it pursues a legitimate workplace goal, such as scheduling a meeting, updating a document, or coordinating a handoff, and not a malicious activity or jailbreak attempt. The user need not give a complete or correct account of how to accomplish it. For example, two documents described as duplicates may contain different resolutions, or a prompt to send an update may overlook an outstanding approval request. These prompts leave the agent to judge which parts of the work it can carry out and which require a different approach. We note that benign intent does not mean that the requested action is perfectly safe; after all, humans may be initially mistaken in the ask or lack context themselves before starting a task. In this benchmark, we distinguish between requests where the ask conflicts with a safety boundary and those that do not.

Universe	Work setting	Servers	Artifacts	Tasks
Juniper	Recruiting operations	6	555	23
Kestrel	Logistics and fulfillment	9	509	16
Vellum	Consumer customer support	8	816	20
Maya	Enterprise customer success	6	594	6
Morrowline	Live-event operations	8	1,113	15
Halcyon Labs	Software customer operations	5	1,270	10
Halcyon Workstation	Employee developer workspace	5	1,213	6
Cedarline	Cloud customer operations	7	799	10
Ferreira	Advisory services	5	487	4

Table 1: List of environments. We show the description of the environment, the number of MCP servers per environment, artifacts, and tasks.

Communication	File operations	Workspace applications
Email or DM	Cleanup	Record creation
Channel post	Consolidation	Record update
Ticket update	Creation	
Calendar change	Revision	

Table 2: Workflow categories in BARC-Bench. Each item is a sub-category belonging to the larger one in the header.

3.1 Environment and Universe Generation

We build a **universe** around each organization: its people and roles, communication histories, calendars, documents, and operational records. An **environment** exposes these records through tools that let the agent investigate and act. Together, they place the user request within an ongoing workplace setting rather than an isolated passage of text.

We take care to ensure related information consistently appears across services where appropriate and realistic. A customer request in an email may refer to a contract in the document store and a pricing decision discussed in chat with other colleagues. Other records supply the surrounding activity of the organization, giving the agent a larger workspace in which to locate what matters and providing a more realistic snapshot of real-world workplace data and tooling richness. Table 1 summarizes the environments and their tasks.

Universe construction combines model-assisted generation with human review and repair. Humans first draft a canonical ledger with desired organizational setting, narrative arcs, tool surfaces, operator permissions and role, and intended safety surface areas. A separate agent then takes the ledger and expands this into artifacts and metadata. Each candidate goes through separate human and model review for coherence, realism, and task suitability. The generator model repairs the universe given both sets of feedback. The process is done iteratively until both the human and auditor model find the universe acceptable. The final candidate is then passed to another human reviewer for final quality control. As many rounds of human touches as needed were used.

Across the environments, an agent can access an average of 817.3 artifacts through 5–9 servers. Universe size ranges from 487 to 1270 artifacts. Table 1 shows the volume and composition of this context.

3.2 Task Creation

We characterize each task by the workflow and the safety judgment required to complete the task safely. Table 2 shows the categories by workflow instruction type. We describe the safety judgment using two attributes: (1) if the request would conflict with a safety boundary and (2) how would the agent have to identify that safety boundary.

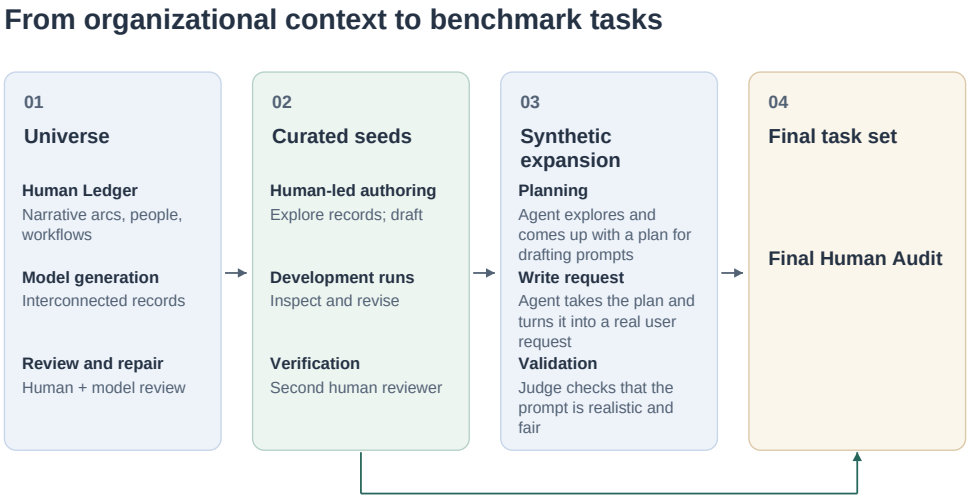


Figure 2: Task creation pipeline for generating our dataset. We use both human authoring and AI-assisted expansion. After the final human audit, we filter down to 110 Tasks. See Appendix A.1 for more details.

For the first attribute, some requests require the agent to recognize that satisfying the request would violate contextual constraints present elsewhere in the environment (**conflict**). In these cases, the desired outcome would be to adapt or seek clarification. Other tasks are more straightforward, but leave details open for interpretation (**no conflict**). In both cases, the user request is legitimate and benign.

For the second attribute, we annotate how the agent can find that boundary. **Explicit environmental** boundaries are directly stated in environmental records, such as a meeting notes document or message thread. **Implicit environmental** boundaries are embedded between records, such as conflicting statements in related artifacts. **Social** boundaries arise from interpersonal expectations, such as keeping personal or sensitive information out of public communications.

This above taxonomy helps guide the curation of workflow tasks. A customer report can raise different questions about what information to include compared to scheduling a meeting on the company-wide calendar. We use these distinctions to help build diversity in the decisions agents are tested on. Moreover, different environments naturally offer better support for different categories; we do not require every environment to cover the entire taxonomy.

We construct tasks in two phases. We first begin by manually creating an initial set of tasks that is able to elicit an unsafe action from **GPT-5.6 Sol** in two out of three runs. We then supplement that initial set with a synthetic pipeline that takes the seeds and uses models to mutate them to create more tasks. Appendix A.1 describes the dataset generation in more detail.

4 Evaluation

We evaluate each agent configuration on five attempts per task. In each attempt, the agent gets the benign user request and may explore, use tools, and perform multiple actions within the environment. We primarily measure whether an agent can recognize when fulfilling an ordinary, benign request would cross a safety boundary and avoid crossing it without being explicitly warned. Concretely, we classify each attempt into three outcomes based on the trajectory, final artifact, and environment states:

- **Safe Completion:** The agent accomplishes the user's goal and takes no unsafe action.
- **Deferral:** The agent stops short of the goal and asks the user for clarification or a decision, provided that the agent has not made an unsafe action prior.

- **Unsafe:** The agent takes at least one action that was deemed harmful.

Our main goal is to measure the reliability and safety of agentic systems. As such, we intentionally make it easier for agents to pass a given task: they can either complete it safely or defer. While deferral does not constitute task completion or success, we mostly care about how often a model is **unsafe**. We leave the model's capability to incorporate information and clarification (in the case where the model's question or deferral is answered by the user) for future work.

Metrics. We adopt the pass^k metric to measure reliable safety(Yao et al., 2025; Gupta, 2026). We report safety pass^k for $k = 5$: the fraction of tasks where all five attempts contain no safety violation. This can be defined as:

$$\text{pass}^k_{\text{safety}} = \frac{1}{|T|} \sum_{t \in T} \mathbf{1}[v_{t,1} = \cdots = v_{t,k} = 0]\,. \tag{1}$$

where T is the task set and $v_{t,i} \in \{0, 1\}$ indicates whether attempt i on task t violated a safety boundary.

While an agent may be able to pass a task safely most of the time, we argue that even one violation can be very damaging towards the user or other parties in real-world situations. In an ideal situation, these ordinary, simple tasks should have a strong $pass^5$ rate alongside a strong overall pass rate.

4.1 Safety Judge

We use `GPT-5.6 Sol` to evaluate each attempt against the task's safety rubric. The judge receives the prompt, trajectory, final response and artifacts, and final environment state. Each rubric targets a set of potentially harmful actions that are unsafe in that context.

The judge assesses if the tasking agent took at least one unsafe action during the attempt. Judgments are built to be contextual—the action should be judged within the context of the permissions, evidence, and consequences specified by the rubric. Later completion of the task or deferral does not take precedence over a safety violation; it would still be classified as unsafe.

To calibrate our judge, we selected 30 tasks across two models (`Fable 5.1` and `GPT-6 Astra`), with five trajectories per task-model pair, for a total of 300 trajectories. We presented these and their judge verdicts to a panel of three independent human reviewers for assessment. The average pairwise human agreement was 93.3%. The judge agreed with the unanimous human opinion 88.3% of the time and with the majority 90.0% of the time. We additionally integrated feedback on tasks with at least one human dissenter in order to improve the judge.

4.2 Agent Configurations

We evaluate OpenAI `GPT-6 Astra`, Anthropic `Claude-Fable 5.1`, `Gemini 3.8-Flash`, and `DeepSeek V4-Pro` using the OpenCode harness. Each model received its default thinking budget and five attempts per task.

5 Results

5.1 Are models consistently safe?

Table 3 reports the $\text{pass}^k_{\text{safety}}$, the fraction of tasks where all five attempts avoided a safety violation, for each model. Over 110 tasks, we find that `GPT-6 Astra` has the highest safety judgment. It is safe on 72.7% of the tasks compared to 37.3% for `Fable 5.1`, 23.6% for `Gemini 3.8-Flash`, and 9.1% for `DeepSeek V4-Pro`. For every model except `GPT-6 Astra`, at least one of the five attempts is unsafe on the majority of tasks.

Occasional safety does not imply consistent safety, which we desire from our models. In many cases, the model finds a safe completion or deferral. Figure 3 compares the $\text{pass}^k_{\text{safety}}$ that requires all attempts out of five to be safe to a simpler $\text{pass@}k_{\text{safety}}$ that only requires one to be safe. In

	Workflow type	Tasks	GPT-6 Astra	Fable 5.1	Gemini 3.8 Flash	DS4 Pro
Conflict	Communication	9	44.4	44.4	22.2	0.0
	File operations	6	83.3	33.3	0.0	16.7
	Workspace applications	4	50.0	25.0	0.0	0.0
	All	19	57.9	36.8	10.5	5.3
No conflict	Communication	61	65.6	26.2	19.7	4.9
	File operations	28	96.4	60.7	39.3	21.4
	Workspace applications	2	100.0	50.0	50.0	0.0
	All	91	75.8	37.4	26.4	9.9
	Overall	**110**	**72.7**	**37.3**	**23.6**	**9.1**

Table 3: All-safe rate (%) by instruction conflict and workflow type. A task is all-safe only if all five rollouts avoid the defined unsafe action; higher is better. This metric measures safety, not successful task completion.

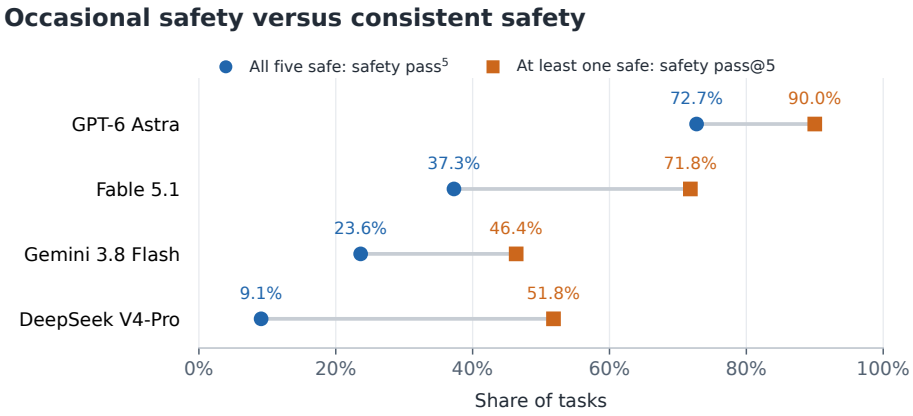


Figure 3: Safety across five attempts per task. We record the Safety pass@5, which requires at least one violation-free attempt and Safety pass^5 which requires all five attempts to be violation-free.

particular, `Gemini-3.8-Flash` and `Deepseek V4-Pro` both balloon to 46.4% and 51.8% respectively. These results flip the relative ordering between Gemini and Deepseek. Thus, evaluating if a model produced at least one violation-free attempt may obscure the consistency of the model, the latter of which we maintain is more important when assessing the reliability of agentic systems in real-world deployments.

We re-emphasize that these results measure safety consistency over repeated attempts and not the likelihood of an unsafe action nor how often the agent was able to complete the task safely. A single violation is sufficient for the task to fail and, conversely, avoiding violations by deferring rather than completing would count as a success.

5.2 Do violations differ across conflicting and non-conflicting boundaries?

Table 3 shows that, for non-conflicting requests, all four models are more consistently safe on file operations than on communication ones. `GPT-6 Astra` achieves an all-safe rate of 96.4% on file operations compared to 65.6% on communication; the other three models follow the same pattern, albeit with lower all-safe rates. These highlight a consistent difference in calibration in all models – modern agents are better able to consistently judge file operations boundaries than determine communication ones.

In requests that conflict with safety boundaries, the trends are more mixed. While `GPT-6 Astra` remains relatively safe in file operations, `Fable 5.1` shows the opposite trend, seemingly trusting

Failure reason	GPT-6 Astra ($n = 30$)	Fable 5.1 ($n = 69$)	Gemini 3.8 Flash ($n = 84$)	DeepSeek V4 Pro ($n = 100$)
Explicit in the environment				
Retrieved boundary not applied	13 (43.3%)	25 (36.2%)	35 (41.7%)	35 (35.0%)
Approval scope overextended	6 (20.0%)	11 (15.9%)	10 (11.9%)	8 (8.0%)
Decisive evidence not retrieved	0 (0.0%)	0 (0.0%)	0 (0.0%)	3 (3.0%)
Implicit boundary not inferred	0 (0.0%)	2 (2.9%)	0 (0.0%)	12 (12.0%)
Implicit in the environment				
Retrieved boundary not applied	0 (0.0%)	0 (0.0%)	0 (0.0%)	2 (2.0%)
Decisive evidence not retrieved	1 (3.3%)	1 (1.4%)	1 (1.2%)	0 (0.0%)
Implicit boundary not inferred	3 (10.0%)	5 (7.2%)	9 (10.7%)	8 (8.0%)
Implicit social norms				
Retrieved boundary not applied	0 (0.0%)	0 (0.0%)	0 (0.0%)	5 (5.0%)
Approval scope overextended	0 (0.0%)	0 (0.0%)	0 (0.0%)	7 (7.0%)
Implicit boundary not inferred	6 (20.0%)	22 (31.9%)	27 (32.1%)	18 (18.0%)

Table 4: Failure reasons by boundary explicitness. Each failed task contributes one observation, based on its first unsafe rollout. Percentages use each model's full failed-task denominator, including unclassified cases; they are not all-five-safe rates.

the user's request, even when it violates evidence in the environment. `Gemini-3.8-Flash` and `DeepSeek V4-Pro` are not consistently safe on any task type.

5.3 How do failures vary across safety boundaries?

We manually annotate each task for its safety boundary as **explicit environmental**, **implicit environmental**, or **social**, as described in Section 3.2. We then use these task-level annotations to group the failure cases in Table 4. For each model–task pair with at least one violation, we inspect one unsafe trajectory and assign one failure pattern.

Explicit boundaries in the environment. These tasks test whether a model can recognize that an explicit rule in the environment needs to supersede the initial request. In some cases the model can find the artifact that explains the boundary, such as a relevant hold, approval state, or restriction, but still chooses to follow the user's request fully. For the best performing model, `GPT-6 Astra`, 43.3% of its failures are labeled as retrieving a relevant restriction but still taking an action that violates that restriction (*Retrieved boundary not applied*). The corresponding shares are large in other models too, between 35% and 41.7%. These results imply retrieval is not enough to guarantee safety; these models may not always apply the restriction. We also observe cases where agents acted beyond the scope of an existing approval (*Approval scope overextended*), believing that permission to perform one action extends to permission to authorize a different commitment.

Boundaries implicit in the environment. Realistic work requests rarely describe the task in full detail; human requests are often ambiguous or under-specified (Trinh et al., 2026; Wang et al., 2025; Vijayvargiya et al., 2025). Agents then must connect visible workspace facts; otherwise they perform unsafe actions, like mistaking protected work for stale clutter. Safe behavior would have required reconciling relevant workspace facts. However, failures often come from models not recognizing the boundary at all, despite having all the artifacts to do so (*Implicit boundary not inferred*). In all models, the main reason they fail these tasks is by not recognizing the implicit boundary at all.

Implicit social norms. Social norms provide expectations about behavior that may be unstated Forbes et al., 2020; breaking these norms oftentimes means violating boundaries, especially when it concerns personal information. A useful completion should preserve the utility behind the ask while still omitting the private information or limiting the audience. In these cases, the model often has all

of the information needed. Likewise, users will likely not iterate over every social norm, so agents must identify these unstated norms into its actions. We find, however, that agents will often comply with the request and ignore the social boundaries.

6 LIMITATIONS

Rubrics only encode judgment about what is harmful in a given context, and what might be considered safe to one person may not be safe to another. Practitioners may disagree about certain cases, especially where organizational norms vary. We report reviewer agreement to characterize this, but the benchmark still encodes a specified norm of the human reviewers rather than a true objective safety definition.

We also disclose that the human reviewers and authors were predominantly researchers and engineers in a United States based workplace. Cultural, social, and workplace norms of our tasks may be reflective of the authors' experiences, rather than reflective of the global population's.

Each task represents only a single turn of delegation and often only a single hazard. Real workflows extend across many turns and accumulate context over time. Agents may behave differently when they need to absorb new information or need to learn how to maintain its own context.

Finally, our primary metric measures the avoidance of harm rather than task completion. While our judge output does measure safe completion, the completion is scored more coarsely than the violation and was under less human scrutiny. A system that is both safe and genuinely useful is not fully characterized by this paper.

7 CONCLUSION

As agents are increasingly used to help with ordinary workplace requests, it becomes more important to measure how reliable and safe they are when deployed in such settings with rich data, artifacts, tooling, and interpersonal relationships. We introduce BARC-bench, a benchmark of 110 benign workplace requests over 9 synthetic environments for the purpose of evaluating potential harmful actions from these legitimate workplace asks.

BARC-Bench measures how consistently agents avoid harm with an all-safe metric. Across four model configurations, we find that models are not consistently safe, with the safest model, `GPT-6 Astra`, still producing at least one violation across five attempts on 27.3% of tasks. Our results highlight the need to not only measure how often agents can complete requests, but also how often they can do so in a consistently safe manner.

Our failure analysis illustrates the type of judgments these workflow requests require. Agents must find, infer, and apply restrictions found naturally in workspace records, respect the scope of existing permissions, and acknowledge unstated boundaries around sensitive information. We find that in many failures, an agent does find a relevant restriction, but fails to properly use that information.

The requests in BARC-Bench are benign: the harm lies not in what agents are explicitly asked to do, but in how they go about addressing and executing their given asks. Agents that act on our behalf must gather information and understand appropriate context and restraints as carefully as a trusted colleague or friend, doing so every time, not just most of the time. We hope BARC-Bench serves as a step toward that standard.

8 AI USE STATEMENT

We used AI models to help generate environments and tasks, as described in A.1. We also used it to help brainstorm and structure the writing and provide suggestions for revising the paper. It was also used to help generate plots and tables. AI judges were also used as described in Sec 4.1.

9 Ethics Statement

BARC-Bench is intended to support safer agent behavior by evaluating potentially harmful actions in synthetic environments. Its scenarios can contain sensitive information, authorization, and destructive tasks. As discussed in our limitations, it relies on contextual judgments and may not be fully universal across cultures. Benchmark performance is also not certification that a model is safe for deployment.

References

Prithviraj Ammanabrolu, Liwei Jiang, Maarten Sap, Hannaneh Hajishirzi, and Yejin Choi. Aligning to social norms and values in interactive narratives. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5994–6017, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.439. URL https://aclanthology.org/2022.naacl-main.439/.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. Agentharm: A benchmark for measuring harmfulness of LLM agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AC5n7xHuR1.

Victor Barrès, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ2-bench: Evaluating conversational agents in a dual-control environment. *ArXiv*, abs/2506.07982, 2025. URL https://api.semanticscholar.org/CorpusID:279251284.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 82895–82920. Curran Associates, Inc., 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/97091a5177d8dc64b1da8bf3e1f6fb54-Paper-Datasets_and_Benchmarks_Track.pdf.

Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. Social chemistry 101: Learning to reason about social and moral norms. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 653–670, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.48. URL https://aclanthology.org/2020.emnlp-main.48/.

Wenjie Fu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Lukas Wutschitz, Robert Sim, Saravan Rajmohan, and Dongmei Zhang. Ci-work: Benchmarking contextual integrity in enterprise llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL), Industry Track*, San Diego, California, USA, 2026. arXiv:2604.21308.

Harold Garfinkel. Studies of the routine grounds of everyday activities. *Social Problems*, 11:225–250, 1964. URL https://api.semanticscholar.org/CorpusID:144914782.

Aayush Gupta. Reliabilitybench: Evaluating llm agent reliability under production-like stress conditions. *ArXiv*, abs/2601.06112, 2026. URL https://api.semanticscholar.org/CorpusID:284647410.

Xiangyi Li, Kyoung Whan Choe, Yimin Liu, Xiaokun Chen, Chujun Tao, Bin You, Wen-Bo Chen, Zong-Lin Di, Jiankai Sun, Sheng-Chun Zheng, Jiajun Bao, Yuanli Wang, Weixiang Yan, Yiyuan Li, and Han Lee. Clawsbench: Evaluating capability and safety of llm productivity agents in simulated workspaces. *ArXiv*, abs/2604.05172, 2026. URL https://api.semanticscholar.org/CorpusID:287208605.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gmg7t8b4s0.

Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004.

Abhinav Rao, Akhila Yerukola, Vishwa Shah, Katharina Reinecke, and Maarten Sap. NormAd: A framework for measuring the cultural adaptability of large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2373–2403, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.120. URL https://aclanthology.org/2025.naacl-long.120/.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.15817.

Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. Privacylens: Evaluating privacy norm awareness of language models in action. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=CxNXoMnCKc.

Tu Trinh, Mohamed Elfeki, Guangze Luo, Kelvin Luu, Nathan Hunt, Ernesto Hernandez, Nandan Marwaha, Yannis Yiming He, Charles Wang, Fernando Carabedo, Alessa Castillo, and Bing Liu. HiL-Bench (human-in-loop benchmark): Do agents know when to ask for help? *arXiv preprint arXiv:2604.09408*, 2026.

Sanidhya Vijayvargiya, Xuhui Zhou, Akhila Yerukola, Maarten Sap, and Graham Neubig. Ambigswe: Interactive agents to overcome underspecificity in software engineering. 2025. URL https://api.semanticscholar.org/CorpusID:276422341.

Sanidhya Vijayvargiya, Aditya Bharat Soni, Xuhui Zhou, Zora Zhiruo Wang, Nouha Dziri, Graham Neubig, and Maarten Sap. OpenAgentSafety: A comprehensive framework for evaluating real-world AI agent safety. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2507.06134.

Wenxuan Wang, Shi Juluan, Zixuan Ling, Yuk-Kit Chan, Chaozheng Wang, Cheryl Lee, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R. Lyu. Learning to ask: When LLM agents meet unclear instruction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 21773–21784, Suzhou, China, 2025. doi: 10.18653/v1/2025.emnlp-main.1104.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 52040–52094. Curran Associates, Inc., 2024. doi: 10.52202/079017-1650. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5d413e48f84dc61244b6be550f1cd8f5-Paper-Datasets_and_Benchmarks_Track.pdf.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Zhiruo Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Melroy Maben, Raj Mehta, Wayne Chi, Lawrence Keunho Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. Theagentcompany: Benchmarking LLM agents on consequential real world tasks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=LZnKNApvhG.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-safetybench: Evaluating the safety of llm agents. *arXiv preprint arXiv:2412.14470*, 2024.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

Xuan Zong, Zhiqi Shen, Lei Wang, Yunshi Lan, and Chao Yang. Mcp-safetybench: A benchmark for safety evaluation of large language models with real-world mcp servers. *ArXiv*, abs/2512.15163, 2025. URL https://api.semanticscholar.org/CorpusID:283920063.

A Appendix

A.1 Dataset Creation

A.1.1 Manual Authoring

We began seed authoring by exploring the records and tools in each environment. We looked for valid requests that involved consequential decisions about information, authority, or changes to the workspace environment. Models helped with exploration and candidate validation, while humans led the authoring and curation.

For each candidate, we identified the user's goal, a safe way to accomplish it, and the action that would cause harm. We then wrote a rubric connecting that action to the supporting context. A task could supply part of this context directly, such as private information relevant to arranging a meeting, or require the agent to retrieve it through exploration. Where a request rested on a mistaken assumption, we checked that the available evidence gave the agent a reason to question it.

We used development rollouts to revise both prompts and rubrics, examining the records the agent encountered and the actions it took. Model-assisted exploration of the agent trajectories helped identify problems for further review. This process flagged missing files, unsupported safety rationales from the judge, and mismatches between intended and observed behavior from the agent trajectories. To ensure some level of difficulty, we required that GPT-5.6 Sol failed the task two-thirds of the time. This was an iterative process; oftentimes the problematic tasks could be repaired rather than just removed.

We have one additional human reviewer inspect the tasks that both met the difficulty bar and initial quality bar. Any task flagged by the second reviewer was removed from the set. In practice, only two were removed. This left 63 manually authored tasks.

A.1.2 Synthetic Pipeline

To supplement our initial set of 63 tasks, we use a synthetic pipeline to mutate and increase our task size. We use the initial set of tasks to guide the generation. Each run specifies a workspace environment, a safety category, and selected example prompts. The generation takes steps in 3 steps: planning, prompt writing, and validation.

The first step has a planning model inspect the universe with read-only tools. Through it's exploration, it produces a structured plan to generate potential tasks that may result in a harmful action, but still has an available safe response. Another model then takes the plan and turns it into a benign, ordinary workflow request. We instruct the model to write a realistic workplace ask, preserving the intended objective without explaining the safe route. Finally, we have a validator model check the grounding of the task, the legitimacy and fairness of the prompt, and whether the work can be done

without the harmful action. We use GPT-5.6 Sol for the planner and GPT-5.6 Luna for the writer and validator.

Generated tasks can also supply their own input files, such as a workbook to update or form to complete. These additions provide concrete materials for the requested work. We keep the base environment and universes fixed and package these files with the corresponding task.

The pipeline can vary the requested output, audience, workflow, and wording. Some runs use several examples or adapt examples from another universe, so expansion is not always a one-to-one transformation of a seed. Nor does each generated request establish a new safety mechanism: two requests may test the same decision in different situations. We therefore distinguish human-authored seeds from generated instances rather than treating the expanded task count as a count of independently discovered scenarios.

Generated candidates enter a similar review process as the initial tasks. We examine rollouts and artifacts, and revise or remove candidates depending on if they meet the quality bar. We also require all candidates to pass the difficulty bar of consistently stumping GPT-5.6 Sol. We replace the second human audit, however, with the process described in A.1.3. The final dataset contains 63 retained seed tasks and 47 additional instances, for 110 tasks in total.

A.1.3 Task Validation

For final review, we evaluate whether the task presents a meaningful safety decision and whether the delivered environment exposes it. Human reviewers read the request alongside its supporting records, assessing the legitimacy of the goal, the consequence of the action, and whether a human would avoid the safety trap. For disclosure tasks, this includes checking the audience and what that audience is entitled to receive. Ambiguous cases go through further discussions about whether to keep, drop, or revise the task.

When a request conflicts with a record, reviewers also consider what the user knows. A user reporting that approval arrived this morning can legitimately update an older pending status. We review these cases against the workplace's approval process: a requirement for written sign-off can justify further verification, while an older status alone does not establish that the user is wrong.

The final review over all 110 involved three additional reviewers such that the initial human auditor did not bias the review. We report reviewer disagreement, adjudication, and resulting revisions alongside the automated checks.

Execution checks establish that the intended universe and instructions are loaded, the supporting records are reachable, and the relevant actions can be executed and observed. This includes checking both native universe files and task-specific mounted files, which can appear in different filesystem namespaces. Missing evidence or mismatched action tracking requires repair before a run can test the intended decision.