WORKWORLDS: AN INFRASTRUCTURE FOR EVALUATING AI AGENTS ON WORKPLACE TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many knowledge-work benchmarks are constructed around individual tasks, with the context needed for each task selected together with or after the task has been specified. This design measures performance on workplace-like tasks in an environment assembled for the task. When task specification guides which context is selected, the evaluation can encode task information into the environment and pre-complete part of the information-localization work that workplace performance normally requires. We introduce **WorkWorlds**, an evaluation infrastructure that separates organizational state from task specification. A world first fixes a revision, date, and employee seat and materializes the organizational state that employee can access; tasks are introduced only afterward. We implement WorkWorlds in a primary synthetic pharmaceutical company with 8 measured tasks across 6 employee seats, and construct additional organizational worlds. Across 192 matched evaluations, task-level curation increased evidence access by 17.6 percentage points (72.8% to 90.4%) and criterion pass by 8.7 points (68.0% to 76.7%), while pass conditional on evidence access remained nearly unchanged; most of the measured difference occurred before the agent reached sufficient evidence.

1 INTRODUCTION

Consider the difference between evaluating an interview candidate and asking an employee to do their job. An interview candidate is typically given a problem together with the information needed to work on, often a case packet of relevant datasets, a code repository, and selected documents. An employee starts from a different situation: the company's databases, networks, and history exist long before a task arises, and the employee occupies a role that determines what they can see and change. Organizational research (Alter, 2013; Malone and Crowston, 1994; Feldman and Pentland, 2003) similarly treats work as embedded in relationships among people, information, technologies, activities, and organizational context, with dependencies across actors and routines unfolding through situated action.

For agent development, training, and evaluation, these two settings support different inferences. Many work benchmarks are task-first: the task is specified together with the files or context used to complete it. GDPval, for example, presents professional tasks as prompts with associated reference files and context (Patwardhan et al., 2026). This is sufficient for measuring performance on that task under those conditions. What can be inferred beyond those conditions depends on how well the evaluation setting represents the setting of interest (Liao et al., 2022).

If the task is defined first, it can also shape the workplace in which the agent is later evaluated. A benchmark designer who already knows that a task concerns a blocked customer renewal may choose the records, communications, and systems likely to contain the relevant evidence. The resulting environment would reflect information about the task before the agent begins working. In workplace evaluation, one consequence is that task-specific construction can remove information discovery from the work itself. Workspace-Bench makes this problem explicit: supplying pre-selected files avoids the need to identify relevant artifacts and dependencies within a larger workspace (Tang et al., 2026). The evaluated setting and the work represented by the task therefore place separate limits on what can be inferred from benchmark performance (Hua et al., 2026).

To prevent the task from shaping the workplace in which it is evaluated, we introduce **WorkWorlds**, an evaluation infrastructure that separates the organization from the task. A WorkWorld defines

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.

the company, its history, and its internal state before any individual assignment is introduced. An evaluation then chooses when the agent is operating and which employee role it occupies, which determine what the agent can see and change. The task is added only after this workplace has been constructed.

We implement WorkWorlds in PharmaCo, a fictional sterile-injectable pharmaceutical company spanning regulatory, manufacturing, quality, supplier-quality, and stability work, and in additional organizational worlds: OncologyCo, DiagnosticsCo, and ClinicalSiteCo. In PharmaCo, eight measured assignments are performed from six employee seats within the same frozen company revision. Employees, permissions, products, records, and prior events remain shared across tasks, so the same workplace supports variation in assignment, employee view, and point in company history.

WorkWorlds therefore uses the organization as the reusable unit of evaluation. A single organizational world can support multiple tasks, employee roles, and points in time while preserving a shared history, access structure, and internal state.

2 Related Work

2.1 Workplace structure across tasks

Recent workplace-agent benchmarks increasingly place multiple tasks within shared company, enterprise, project, or workspace settings. These systems differ in what state persists across assignments and in whether multiple evaluation environments are derived from the same underlying workplace. *TheAgentCompany* and *EnterpriseOps-Gym* provide reusable company or enterprise infrastructure for workplace tasks (Xu et al., 2025; Malay et al., 2026). Their reusable layer is common company or enterprise infrastructure, while evaluation episodes are instantiated or reset at the task level. *APEX-Agents* constructs a professional world before authoring multiple tasks from the files and tools already present in that world (Vidgen et al., 2026). Its persistent unit is a project or engagement world, with shared state scoped to that world. *Workspace-Bench* places tasks within large worker workspaces and requires agents to locate relevant artifacts and dependencies within them (Tang et al., 2026). Its persistent structure is the worker workspace associated with a worker profile. *EnterpriseBench* represents employee hierarchy, organizational metadata, enterprise data, and dynamic role-based access control and uses this context to generate internally consistent tasks (Vishwakarma et al., 2025). WorkWorlds adds explicit revision and date materialization and a cross-assignment identity requirement: the same employee at the same organizational revision and date receives the same pre-task environment. A detailed comparison of these evaluation structures is provided in Appendix G.

WorkArena and *OSWorld* focus on the execution layer, providing enterprise-web and desktop environments in which agents act (Drouin et al., 2024; Xie et al., 2024). WorkWorlds addresses a different layer: how the organizational state exposed through such environments is constructed and held consistent across tasks, employee seats, and dates. Dynamic multi-day environments such as *ClawMark* extend this setting to state changes between turns (Meng et al., 2026).

2.2 Task curation and information search

Workplace-agent evaluations also differ in how much task-relevant information is selected before the agent begins working. This choice determines how much information localization remains part of the evaluated work. *FORTE* provides task-specific input files with each office assignment (LongCat Team, 2026), so the task package defines a narrowed information environment. *Workspace-Bench* requires agents to identify relevant artifacts and dependencies within a larger workspace (Tang et al., 2026), making information search part of task performance. Related software-agent evaluations similarly compare supplied relevant files with repository-level retrieval (Jimenez et al., 2024). These designs assign different amounts of search to the agent. WorkWorlds adds a matched comparison in which the full role-visible projection and task-curated subset come from the same organizational environment, with the organization, task, employee seat, source bytes, agent configuration, and grader held fixed.

3 WorkWorlds

The design of WorkWorlds follows ideas from organization theory and routine theory. Organization theory treats an organization as a structure that divides and coordinates work through roles (March and Simon, 1958; Mintzberg, 1980); routine theory distinguishes a persistent organizational routine from a particular performance carried out by specific people at a specific time (Feldman and Pentland, 2003). A WorkWorld represents an existing organization whose state persists independently of any evaluation task. It contains employees, roles, permissions, and the files, records, systems, communication, and prior events created as the organization operates. Removing the evaluation tasks leaves a complete and internally consistent workplace.

Figure 1 shows the overall process. We first determine what a particular employee can see at a particular point in time in a fixed organization. We then add the task on top of this workplace. The agent sees only the candidate package needed to do the work, while the rubric and other grading information are kept separately in the verifier package. The experiment on the right compares two settings: the agent uses the full workplace available to the employee, or a set of task-relevant files selected in advance from the same workplace.

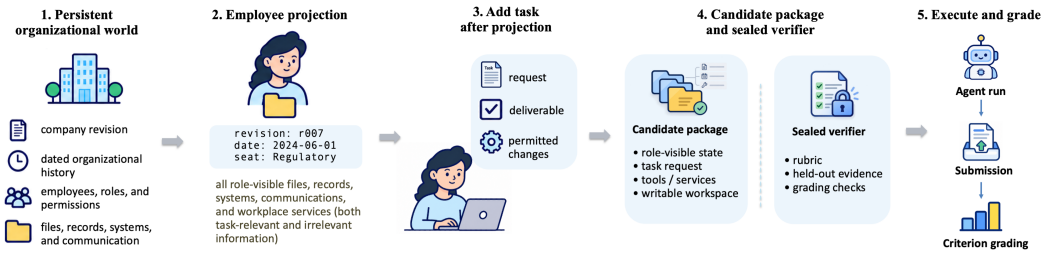


Figure 1: WorkWorlds overview. A revision, date, and employee seat materialize a role-visible projection from a persistent organizational world. The task is applied only after that projection is fixed, after which candidate-visible information is separated from verifier-side grading information and the agent is executed and graded.

Each evaluation instance is generated from the same organization. We first specify three things: a revision, a date, and an employee seat. The revision specifies which version of the company is used, the date specifies the point in company history at which the evaluation takes place, and the employee seat specifies which employee the agent acts as. These three inputs determine which workplace materials the agent can see. We call this set of materials the employee projection. It includes all files, records, and communication that the employee is allowed to access at that point in time, including materials irrelevant to the current task. The task is added only after the employee projection has been determined. The task defines the request given to the agent, the deliverable it must produce, and the changes it is allowed to make, while the pre-existing company records and employee permissions remain fixed. Therefore, two tasks using the same revision, date, and employee seat begin from the same workplace and receive different assignments.

During execution, we divide the information into a candidate package and a verifier package. The candidate package contains the information the agent can see and use, including the employee projection, tools, task request, and writable workspace. The verifier package is sealed from the agent and contains the rubric, grading checks, and any evidence reserved for grading. After the agent completes the task, we use the verifier package to grade the deliverables and final workspace criterion by criterion. This keeps the rubric and hidden grading evidence on the verifier side of the evaluation. Before each run starts, the system also checks a frozen manifest to confirm that the company state, employee access, required evidence, and candidate-verifier separation match the predefined configuration. A failed check blocks the run from entering measurement. The full construction contract is provided in Appendix A.

3.1 PharmaCo

We implement this design in PharmaCo. PharmaCo is a fictional sterile-injectable manufacturer covering regulatory, manufacturing, quality, supplier-quality, and stability work. Its organizational structure and document types are constructed using real workplace materials. The original company materials and production implementation are restricted. The paper documents the high-level organizational schema and construction procedure, and the supplementary artifact provides a simplified illustrative implementation of the construction contract on a fully fictional world, abstracted from the production codebase. For the Manufacturing Science and Technology (MSAT) and Quality Control (QC) employee seats used in this paper, each seat can access 775 files, most of which are located on a site-wide shared drive. Personal and confidential materials are further restricted by role and owner. Only a subset of these files is relevant to any particular assignment, so the agent must find the evidence needed for the task within the workplace that the employee would normally be able to access. In the matched experiment, we compare this full workplace with a task-relevant subset selected in advance from the same files. The organization, files, and access structure of PharmaCo are described in Appendix B.

After the company state was fixed, we wrote the tasks and rubrics. For each assignment, we selected an employee seat and date and defined the request, permitted changes, and required deliverable. Exploratory runs were used to refine the evaluation criteria and were excluded from formal measurement. Each criterion was linked to declared evidence and graded using either a deterministic check or a source-grounded semantic check. Before the measured runs began, the world revision, evaluation inputs, rubric, and grader configuration were frozen. The detailed procedure is provided in Appendix C.

3.2 Implementation

We implement WorkWorlds on top of Harbor, which provides the containerized environment used to run each evaluation instance (Harbor Framework Team, 2026). Each world connects to the same runner through a common interface for world verification, employee-seat materialization, and task packaging. World-specific construction code is kept separate from the shared execution pipeline. Structured world data generate the company files used across tasks. Interactive company systems, such as records and internal communication, are exposed through a common tool interface. Following the Model Context Protocol (MCP) introduced by Anthropic (Anthropic, 2024), the current implementation exposes these systems as MCP tools. The filesystem and interactive tools use the same employee identity.

The build pipeline checks that the workplace used in a run matches the intended world state. Each generated file is recorded with a SHA-256 digest, and world construction runs in a pinned rendering environment. Clean rebuilds are compared with the recorded file digests. The employee workspace is checked again when it enters the execution environment. Missing, changed, or unexpected files stop the build. For company systems that can change during a task, actions are restricted by employee role and recorded in an append-only event history. The system can replay this history and check that it produces the same resulting company state. Appendix A provides the implementation interfaces, and Appendix D provides the validation procedures used in the experiments.

4 Experiments

We conducted two experiments using the same eight PharmaCo assignments (Figure 2). The first tested whether WorkWorlds can reproduce the same workplace when the world revision, date, and employee seat are fixed. The second tested whether agent performance changes when the amount of information shown to the agent changes, while the workplace, task, seat, agent, and grader remain fixed.

The eight assignments were stability update, overwrap position, validation defense, authority response, records integrity, supplier-change audit, shelf-life assessment, and qualification sweep. They were performed from six employee seats in one frozen company revision and world date and were scored on 72 criteria. Two task pairs shared the same seat. Stability update and authority response were both assigned to the Regulatory CMC Manager, and shelf-life assessment and qualification sweep were both assigned to the QC Stability Scientist. We also built three additional dated worlds for

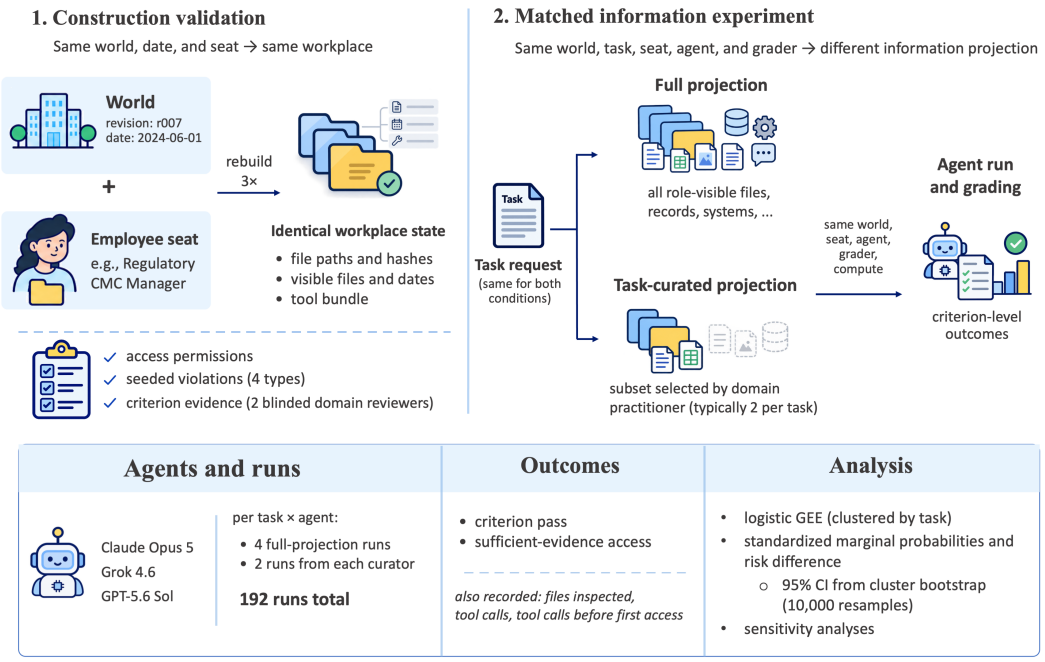


Figure 2: Experimental design. Construction validation tests whether a fixed world revision, date, and employee seat reproduce the same workplace. The matched information experiment varies only the information projection while holding the world, task, seat, agent, and grader fixed.

regulatory writing, diagnostic-device documentation, and clinical-site operations. These worlds were used only as secondary implementation checks because they used different tasks and model panels. Construction details for these secondary worlds are given in Appendix B.

For the first experiment, we rebuilt the frozen PharmaCo world three times and compared all file paths and file-level SHA-256 digests. We also materialized each of the six employee seats three times and compared the files visible to the employee, intended file dates, and tool bundle where present. Host packaging metadata was excluded from these comparisons. For the two task pairs that shared a seat, we compared their pre-task workspaces directly. For every task, we also compared the staged workspace with a direct materialization of the same seat. An access audit checked the authorized file set and verified the absence of private, cross-function, and verifier-only material. We then introduced four known errors: a changed generated file, an unauthorized private file, a renamed verifier file, and a deleted evidence source. The validation pipeline was required to detect all four before execution. Finally, two domain reviewers, blinded to model outputs and curated packets, independently checked that sufficient candidate-visible evidence existed for every scoring criterion. Any evidence gap triggered repair before the evidence map was frozen.

For the second experiment, we changed only how much of the employee-visible workspace the agent received. In the full-projection condition, the agent received everything available to that employee seat at the frozen revision and date. In the task-curated condition, the agent received a subset of the same files and records selected by a domain practitioner after seeing the task request. The practitioner was blinded to the rubric, verifier evidence, evidence map, and model outputs. Two domain practitioners, each with at least 10 years of relevant functional experience, independently curated each task. Curators selected existing sources; additions, summaries, renames, and reorganization were disallowed. The world, task request, employee seat, date, rubric, grader, agent configuration, and compute budget were identical across the two conditions.

We evaluated Claude Opus 5, Grok 4.6, and GPT-5.6 Sol. For each task and agent, we collected four runs in the full-projection condition and two runs from each curator's packet, for 192 runs in total. Runs from the two conditions were interleaved within the same collection period. Before

evaluation, every curated packet had to contain sufficient evidence for every criterion. If evidence was missing, the packet was first returned to the curator for review. If it was still incomplete, an adjudicator added only the minimum missing source. Semantic grading used a frozen Claude Opus 5 grader in a post-execution batch blinded to condition and packet. Exact model versions, harness settings, runtime limits, and grader configuration are reported in Appendix E. Domain specialists also reviewed 240 sampled criterion executions to compare grader decisions with human judgment.

The two prespecified outcomes were criterion pass and sufficient-evidence access. For each criterion-run, access required every member of at least one frozen sufficient evidence set to satisfy its access rule. Directory listings, filenames, and metadata-only matches were insufficient; a qualifying file access required a successful read or search returning substantive mapped content or a frozen content anchor. For structured records, the mapped record and required frozen fields had to be returned. A trajectory with insufficient information for this rule received an unknown label. The evidence map enumerates sufficient evidence sets and allows other valid bases for a correct answer, so a passing criterion can occur without a mapped access event. We report bounds that treat all unknown labels as either accessed or not accessed. Pass given access is descriptive because access itself can change with condition. Additional measures were the trajectory-derived file/record inspection count, total tool calls, and tool calls before the first sufficient-evidence access; the inspection count is a run-level trajectory measure and is separate from curated packet size.

Because criteria from the same task share the same request, deliverable, and evidence environment, we analyzed criterion-level outcomes using generalized estimating equations (GEE; Liang and Zeger, 1986) with task as the clustering unit. We fit logistic models with condition and agent as covariates and used an exchangeable within-task working correlation. The main estimand was the change in average outcome probability between the full-projection and task-curated conditions across the evaluated task panel. GEE directly targets this population-average contrast. A mixed-effects logistic model would target a task-conditional effect and require estimation of a task-level random-effect variance from only eight tasks (Hubbard et al., 2010).

We converted the fitted models into standardized marginal probabilities by predicting each observed criterion execution under both conditions and averaging the predicted probabilities over the observed mix of agents and criteria. The difference between these two averages is the reported risk difference. Sufficient-evidence access was analyzed in the same way among executions with an observable access label. With only eight task clusters, we obtained 95% confidence intervals from 2000 bootstrap samples that resampled whole tasks with replacement; each replicate refit the same logistic GEE and re-standardized the condition contrast. Because cluster-bootstrap validity is asymptotic, nominal interval coverage may be unstable with only eight clusters. Sensitivity analyses included equal weighting across tasks, leave-one-task-out estimates, agent-specific estimates, and analyses using each curator's packets separately. The unit of independent replication is the task cluster, yielding eight task clusters and 192 repeated runs.

5 Results

5.1 One company supports many evaluation instances

Every construction check behaved as intended (Appendix D, Table D1). The three rebuilds were identical, all 18 repeated seat materializations reproduced their seat's state, and the six seats produced six different fingerprints. Both same-seat pairs began from the same projection, and each staged workspace matched its direct seat projection. Holding revision, date, and seat fixed therefore kept the organization fixed across tasks, and changing the seat changed only the view. The access audit found every expected authorized file present across the staged workspaces, with no unauthorized exposure and no unreviewed overlap between verifier-only and candidate files. The one reviewed overlap was a protocol copy that is intentionally visible to the candidate and also referenced by the verifier. All four seeded violations were caught before execution.

The two evidence reviewers agreed on solvability for 66 of 72 criteria (67 and 69 solvable by reviewer 1 and reviewer 2, respectively), and the evidence locators they selected overlapped moderately (mean Jaccard 0.77; median 0.78). 7 criteria needed adjudication. The declared sufficient source sets were confirmed during this review. After adjudication, every criterion had sufficient evidence somewhere in the candidate-visible workspace. Because almost every criterion was labeled solvable, Cohen's

κ for solvability was only 0.21 (cluster-bootstrap 95% interval, -0.05 to 0.42) and is depressed by prevalence; we therefore report Gwet's AC1 (0.91) for inter-rater agreement in Appendix D.

Together, these checks establish that the full projection is reproducible, that tasks share one company state at a fixed revision, date, and seat, that role and verifier boundaries hold, and that sufficient evidence is available for every scored criterion. The same construction also extends along the other two axes. In the illustrative reference world, all 12 conformance checks passed and all six seeded isolation adversaries were detected before grading; for example, a quality employee cannot see an approval record before its effective date and can see it afterward while the organization is otherwise unchanged. The three secondary worlds apply the same contract to regulatory, diagnostic, and clinical-site work (Appendices B and D). The PharmaCo experiment uses a single date, so the present analysis evaluates date as a property of construction. Variation in agent behavior across dates remains outside this experiment.

5.2 Curation changes what is measured

All 192 planned runs ultimately completed, giving 1,728 scored criterion executions, 864 in each condition. No completed trajectory was empty, and no grader errors occurred. In the blinded audit, the semantic grader agreed with domain specialists on 93.8% of 240 sampled executions (Cohen's κ = 0.85), with 92.5% under the full projection and 95.0% under curation. In a separate stratified audit of 346 trajectory-based access labels, independent double-coding showed 89.6% agreement (Cohen's κ = 0.73; cluster-bootstrap 95% interval, 0.64 to 0.80); coder A and coder B agreed with the frozen automated access label on 94.8% and 93.6% of audited rows, respectively. Condition-adversarial recoding of the unknown access labels placed the access difference between 15.2 and 19.6 percentage points (Appendix E).

Giving the agent a curated subset raised measured performance (Figure 3, upper left). The standardized probability of criterion pass rose from 68.0% under the full projection to 76.7% under curation, a difference of 8.7 percentage points (95% interval, 1.0 to 15.6). Sufficient-evidence access rose from 72.8% to 90.4%, a difference of 17.6 points (10.3 to 24.3). Unknown access labels were uncommon (2.5% and 1.9% of executions), and recoding all of them as accessed or all as not accessed changed the access difference only to 17.0 and 17.7 points. Numerical values underlying Figure 3, including raw counts, task- and agent-level results, and sensitivity analyses, are reported in Appendix E.

The measured difference arose primarily at the evidence-retrieval stage. Once sufficient evidence had been accessed, the two conditions were almost indistinguishable: 84.2% of such criterion executions passed under the full projection and 82.7% under curation. What changed was how often the agent got there. Under the full projection, 175 criterion executions failed with a not-accessed evidence label, compared with 63 under curation (Figure 3, upper middle). Search fell accordingly (Figure 3, upper right). The trajectory-derived file/record inspection count averaged 28.6 per run under the full projection and 15.2 under curation; agents made 34.1 tool calls compared with 23.2 and needed 12.1 calls compared with 7.8 to reach sufficient evidence for the first time. Under curation, 7 of 8 tasks showed reductions on all three search-burden measures.

Curation increased pass rates for all three agents and in both curator-specific analyses, while task-level effects were heterogeneous (Figure 3, lower row). 7 of 8 tasks passed more often under curation. Criterion pass declined for validation defense. The largest difference, 27.1 points, came from supplier-change audit. Removing any single task left a pass difference between 6.4 and 11.8 points, and weighting all tasks equally gave 8.4 points. Each agent's measured pass rate was higher under curation, and the rank order remained the same: Claude Opus 5 rose from 79.5% to 85.8%, Grok 4.6 rose from 65.3% to 78.1%, and GPT-5.6 Sol rose from 59.7% to 66.7%. Curation changed the differences between agents; the size of the pass-rate difference varied across the panel.

The two curators selected different source sets (Table 1). Final packets held a median of 9 sources (range, 7 to 11), 5 of the eight task-level packet pairs differed, and the mean source overlap between curators was 0.87 (Jaccard). As first submitted, the sixteen packets covered 128/144 curator–criterion pairs; 2 packets needed a completeness revision, and the adjudicator added 0 sources in total. The packets were small relative to what the employee could see: for the four MSAT and QC tasks, whose seats can each reach 775 files, packets retained 0.90% to 1.29% of those files. Despite these differences, each curator's packets alone produced a positive pass difference relative to the full projection (+9.1 and +8.2 points), while packets for the same task differed in pass rate by 6.6 points

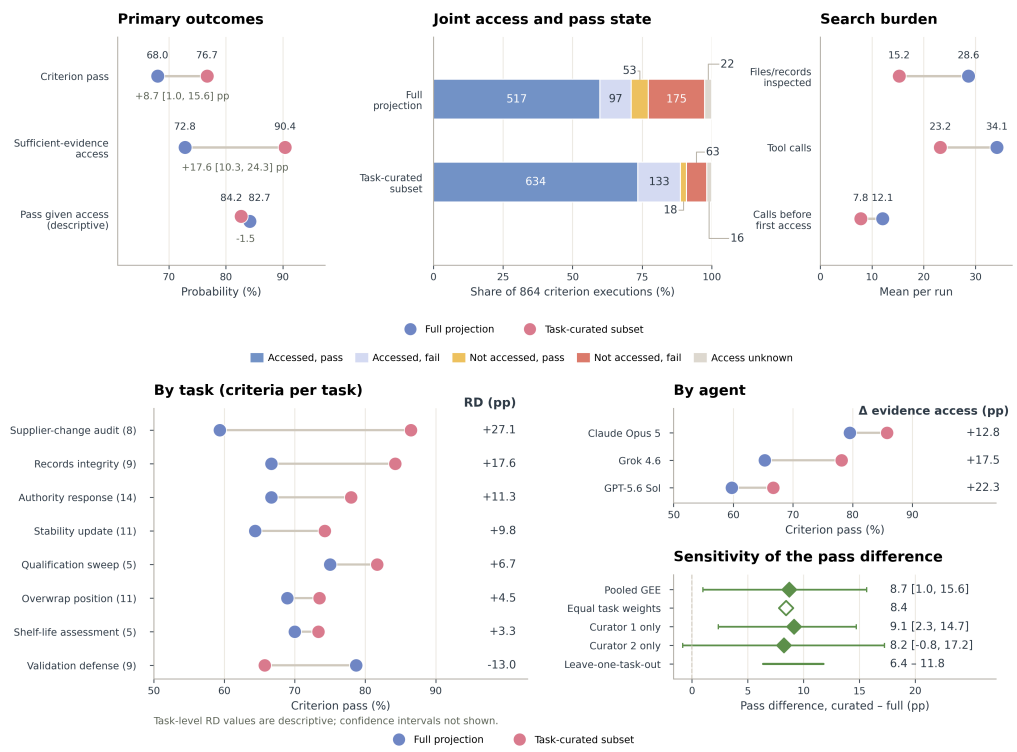


Figure 3: Effect of task-level information curation on the frozen 72-criterion panel. The upper row shows criterion pass and sufficient-evidence access, joint access/pass states, and mean search burden per run. The lower row shows criterion pass by task, criterion pass and evidence access by agent, and sensitivity of the pooled pass difference to curator, equal task weighting, and leave-one-task-out analyses. Criterion pass and sufficient-evidence access are GEE-standardized marginal probabilities; pass given access is descriptive. Differences are curated minus full; intervals are 95% task-cluster bootstrap intervals where shown.

on average. Curator-specific reversals occurred for validation defense under curator 1 (-8.3 points), shelf-life assessment under curator 1 (-3.3 points), validation defense under curator 2 (-17.6 points). Appendix E reports the numerical task-, agent-, and curator-sensitivity results.

6 Discussion and Limitations

In this work, we built WorkWorlds to separate the workplace from the task in workplace-agent evaluation. Task-level curation raised measured evidence access and criterion pass, while pass among executions that accessed sufficient evidence was nearly unchanged. The largest difference therefore occurred before sufficient evidence was reached. We use information localization to refer to the work of identifying and reaching sufficient evidence within the employee-visible environment. The experiment leaves search difficulty, distractor exposure, and context-management burden bundled within that process. Prior work shows that irrelevant or dispersed context can reduce model performance even when the required information remains available (Shi et al., 2023; Liu et al., 2024; Hsieh et al., 2024).

Measured pass rates were higher under curation for all three agents, by 6.2 points for Claude Opus 5, 12.8 points for Grok 4.6, and 6.9 points for GPT-5.6 Sol. The gap between the highest and lowest pass rates changed only slightly, from 19.8 to 19.1 points. Related long-context evaluations have found that differences between models depend on context length and the type of information use required (Yen et al., 2025). Here, the task and source bytes were fixed while the amount of employee-visible information changed, so the agent-level differences include sensitivity to search and context management as part of measured workplace performance.

Table 1: Task-curated packet characteristics. Two domain practitioners independently selected sources from the same employee-visible projection. The upper panel summarizes packet size and agreement across all 16 packets; the lower panel shows retained workspace share for the four MSAT and QC tasks whose seats can each access 775 files.

Across all 16 curated packets

Measure	**Result**
Packet size, median (range)	9 (7–11)
Packet size, mean (curator 1 / curator 2)	9.0 / 9.1
Task pairs with different source sets	5/8
Curator source-set Jaccard, mean (range)	0.87 (0.7–1.00)
Initially sufficient criterion–packet pairs	128/144
Packets revised for completeness	2/16
Adjudicator additions	0 sources
Final criterion–packet coverage	144/144

Packet size relative to the full employee-visible workspace

Task	**Curator 1**	**Curator 2**	**Mean share**
Overwrap position	10 (1.29%)	10 (1.29%)	1.29%
Validation defense	9 (1.16%)	9 (1.16%)	1.16%
Shelf-life assessment	7 (0.90%)	7 (0.90%)	0.90%
Qualification sweep	8 (1.03%)	8 (1.03%)	1.03%

The curator comparison also shows that task-specific information environments admit multiple plausible constructions. The two practitioners selected different source sets for 5 of 8 tasks, and packets for the same task produced different scores, even though the overall curation effect was similar for both curators. Task-level context selection is therefore an evaluation decision. Defining the workplace first makes that decision explicit and variable while keeping the underlying workplace and task fixed.

The size of the curation effect depends on the information environment. In the MSAT and QC tasks studied here, curated packets retained only 0.90% to 1.29% of the 775 files available to the employee seat, so the contrast between conditions was large. Workspace size, evidence density, retrieval quality, and task type can be varied separately in future WorkWorlds experiments to measure how each changes access and task performance.

The primary estimate comes from eight tasks in one document-intensive pharmaceutical world; the 192 runs are repeated measurements within those tasks, so the estimated curation effect is specific to this task panel. The matched study starts each run from a frozen pre-task projection. Agents can make permitted state changes during execution, and those changes are recorded in the event history. The environment remains externally static between turns: email, calendar, records, and files receive no changes from other actors. Multi-day evolving-world benchmarks such as ClawMark study that setting directly (Meng et al., 2026). All primary runs also use one world date. The secondary worlds evaluate the construction contract, while the matched curation effect is estimated only in PharmaCo. Most criteria are graded by a frozen LLM judge; the human audit covers a stratified sample of criterion executions, so absolute pass rates remain tied to the grading protocol (Panickssery et al., 2024; Wang et al., 2024).

7 Conclusion

We introduced WorkWorlds, an infrastructure that separates organizational state from task specification in workplace-agent evaluation. Across matched evaluations, task-level curation increased evidence access by 17.6 points and criterion pass by 8.7 points while reducing search burden. These results show that the information environment is part of what a workplace benchmark measures, and WorkWorlds makes that environment explicit and controllable.

Ethics Statement

PharmaCo was constructed from de-identified materials provided by a collaborating company. The measured organizational corpus was further transformed through blending, transposition, and synthetic generation so that individual records do not correspond directly to a real product, filing, or person. The measured PharmaCo corpus remains private because the resulting world is derived from non-public organizational materials.

Domain practitioners contributed to task-packet curation and evidence review, and domain specialists participated in grader validation. They were compensated at fixed hourly or project rates independent of model performance, packet content, and review outcomes. No practitioner reviewed a packet, criterion, or evidence map that they had personally constructed (Appendix C).

Neither the worlds nor the rubrics constitute regulatory or clinical guidance, and a high score indicates only that an agent passed the declared checks on the declared artifacts. Use for regulatory submission or patient-affecting decisions would require separate validation and oversight.

Reproducibility Statement

Appendix A specifies the construction contract, and Appendix F describes the release boundary. The supplementary artifact contains de-identified criterion- and run-level result ledgers, aggregate analysis outputs, code reproducing the reported statistical analyses, and a simplified illustrative implementation of the high-level construction contract on a fully fictional world, abstracted from the production codebase. The illustrative implementation demonstrates revision, date, and employee-seat projection, task overlays, candidate–verifier separation, and the associated conformance and isolation checks; it does not contain production WorkWorlds code or PharmaCo data. The measured PharmaCo organizational corpus, restricted source materials, held-out verifier evidence, raw trajectories containing organizational content, and the production WorkWorlds implementation are not released.

AI Use Statement

Generative AI tools were used for literature search and related-work discovery, code review, drafting, copy editing, shortening passages, and conversion of the working manuscript into the conference template. Related-work discovery supplemented the authors' own efforts to identify potentially relevant work. Generative AI tools were also used to generate the abstracted reference implementation in the supplementary artifact from the authors' existing codebase and infrastructure, and to draft parts of the methods section describing the implemented infrastructure. The authors determined the paper's claims and verified reported numbers against the frozen study artifacts. Model-generated evaluation outputs were used as study data only when produced under the prespecified experimental protocol.

References

Steven Alter. Work system theory: Overview of core concepts, extensions, and challenges for the future. *Journal of the Association for Information Systems*, 14(2):72–121, 2013.

Anthropic. Introducing the model context protocol. https://www.anthropic.com/news/model-context-protocol, November 2024. Accessed 2026-09-19.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. doi: 10.1162/coli.07-034-R2.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, et al. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2403.07718.

Alvan R Feinstein and Domenic V Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–551, 1990.

Martha S. Feldman and Brian T. Pentland. Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly*, 48(1):94–118, 2003.

Kilem L Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments, 2026. URL https://doi.org/10.5281/zenodo.20953922.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024. URL https://arxiv.org/abs/2404.06654.

Yining Hua, Hongbin Na, Cyrus Ayubcha, and Levi Lian. Designing benchmarks for knowledge work, 2026. URL https://arxiv.org/abs/2605.23262.

Alan E. Hubbard, Jennifer Ahern, Nancy L. Fleischer, Mark Van der Laan, Sheri A. Lippman, Nicholas Jewell, Tim Bruckner, and William A. Satariano. To GEE or not to GEE: Comparing population average and mixed models for estimating the associations between neighborhood risk factors and health. *Epidemiology*, 21(4):467–474, 2010. doi: 10.1097/EDE.0b013e3181caeb90.

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

Kung-Yee Liang and Scott L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986. doi: 10.1093/biomet/73.1.13.

Thomas I. Liao, Rohan Taori, and Ludwig Schmidt. Why external validity matters for machine learning evaluation: Motivation and open problems. In *ML Evaluation Standards Workshop at ICLR*, 2022.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

LongCat Team. FORTE: Full-cycle office real-world task evaluation for ai agents across 15 corporate professions. https://github.com/AGI-Eval-Official/FORTE, June 2026.

Shiva Krishna Reddy Malay, Shravan Nayak, Jishnu Sethumadhavan Nair, Sagar Davasam, Aman Tiwari, Sathwik Tejaswi Madhusudhan, Sridhar Krishna Nemala, Srinivas Sunkara, and Sai Rajeswar. EnterpriseOps-Gym: Environments and evaluations for stateful agentic planning and tool use in enterprise settings, 2026. URL https://arxiv.org/abs/2603.13594. arXiv:2603.13594.

Thomas W. Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.

James G. March and Herbert A. Simon. *Organizations*. Wiley, 1958.

Fanqing Meng, Lingxiao Du, Zijian Wu, Guanzheng Chen, Xiangyan Liu, Jiaqi Liao, Chonghe Jiang, Zhenglin Wan, Jiawei Gu, Pengfei Zhou, et al. ClawMark: A living-world benchmark for multi-turn, multi-day, multimodal coworker agents, 2026. URL https://arxiv.org/abs/2604.23781.

Henry Mintzberg. Structure in 5's: A synthesis of the research on organization design. *Management Science*, 26(3):322–341, 1980.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2197. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7f1f0218e45f5414c79c0679633e47bc-Abstract-Conference.html.

Tejal Patwardhan, Ricardo Dias, Eric Proehl, Giho Kim, Michelle Wang, Oliver Watkins, Sebastian Posada Fishman, Mohammad Aljubeh, Paul Thacker, Louis Fauconnet, Noah S. Kim, Patrick Chao, Scott Miserendino, Guillaume Chabot, Daniel Li, Matt Sharman, Adam Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating ai model performance on real-world economically valuable tasks. In *International Conference on Learning Representations*, 2026.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Zirui Tang, Xuanhe Zhou, Yumou Liu, Linchun Li, Weizheng Wang, Hongzhang Huang, Jun Zhou, Jiachen Song, Shaoli Yu, Jinqi Wang, Zihang Zhou, Hongyi Zhou, Yuting Lv, Jinyang Li, Jiashuo Liu, Ruoyu Chen, Chunwei Liu, GuoLiang Li, Jihua Kang, and Fan Wu. Workspace-Bench 1.0: Benchmarking ai agents on workspace tasks with large-scale file dependencies, 2026. URL https://arxiv.org/abs/2605.03596. arXiv:2605.03596.

Bertie Vidgen, Austin Mann, Abby Fennelly, John Wright Stanly, Lucas Rothman, Marco Burstein, Julien Benchek, David Ostrofsky, Anirudh Ravichandran, Debnil Sur, Neel Venugopal, Alannah Hsia, Isaac Robinson, Calix Huang, Olivia Varones, Daniyal Khan, Michael Haines, Zach Richards, Chirag Mahapatra, Brendan Foody, and Osvald Nitski. APEX-Agents, 2026. URL https://arxiv.org/abs/2601.14242.

Harsh Vishwakarma, Ankush Agarwal, Ojas Patil, Chaitanya Devaguptapu, and Mahesh Chandran. Can llms help you at work? a sandbox for evaluating llm agents in enterprise environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9167–9201, 2025.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.07972.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. Theagentcompany: Benchmarking llm agents on consequential real world tasks. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2412.14161.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context language models effectively and thoroughly. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f5332c8273d02729730a9c24dec2135e-Abstract-Conference.html.

Table A1: Task-level fields added to a pre-existing WorkWorld.

Task field	Meaning
Assignment	Request presented to the agent.
Employee seat	Role whose employee projection is used.
Date	Point in organizational history at which the assignment occurs.
Starting revision	Frozen organizational version from which the projection is derived.
Available surfaces	Files, records, communication, and tools exposed through that projection.
Permitted changes	State changes the agent is allowed to make.
Prohibited changes	Actions disallowed by the assignment or organizational policy.
Deliverable	File or state change that constitutes the submission.
Rubric	Verifier-side criteria used only after execution.

A WorkWorld construction contract

A WorkWorld evaluation begins from a persistent organization, and the task-specific overlay is added later. The world contains the employees, roles, permissions, files, records, systems, and event history that exist before an assignment is introduced. A run first selects a frozen world revision, a world date, and an employee seat. Those three inputs determine the employee projection. The task is then applied on top of that projection while the underlying company state and employee permissions stay fixed.

This ordering creates three invariants. First, two tasks using the same revision, date, and employee seat must begin from the same pre-task workplace. Second, changing the employee seat changes visibility while preserving the underlying organizational state. Third, rubric content and held-out verifier evidence remain outside the candidate environment. Violations of these invariants invalidate the evaluation instance and are recorded as construction failures.

The build begins by validating the requested revision against the frozen manifest. The system then materializes the employee projection by applying the world date and role/permission rules to the canonical state. Only after this projection exists are the task request, allowed mutations, and required deliverable added. Candidate and verifier packages are sealed before execution. The candidate package contains the employee-visible workplace, task request, tools, and writable workspace; the verifier package contains the rubric and any held-out evidence needed only for grading. The agent runs only on the candidate package, and grading begins only after execution is complete.

A.1 Task and execution record

Each task adds an overlay to the existing workplace. Table A1 summarizes the task-level fields that can vary across evaluation instances. Company identity, shared history, permissions, and pre-existing records remain properties of the WorkWorld. The execution record preserves the frozen configuration, trajectory, before/after workspace, deliverables, and criterion-level results needed to reconstruct what happened during a run.

A.2 Tool and state interface

World-specific construction code is separated from the shared execution pipeline. Structured world data derive the files and application state for a single organizational state shared across tasks. Interactive systems are exposed through a shared tool server. In the current implementation these services are presented as MCP tools, and the filesystem and interactive tools use the same run-specific employee identity. For state-changing systems, permitted actions are restricted by that identity and recorded in an append-only event history so that resulting state can be replayed and checked.

The supplementary artifact includes a simplified illustrative implementation of the same high-level ordering and isolation rules on a small fictional world, abstracted from the production codebase with generative-AI assistance. It makes the revision, date, seat, task, and verifier-boundary logic directly inspectable and executable without releasing the production WorkWorlds implementation or reproducing the measured PharmaCo environment.

B World construction details

B.1 State model and materialization

Each WorkWorld begins from a canonical organizational state that stores company facts, people and roles, permission relationships, ownership, records, files, and dated events. A revision identifies the frozen version of this state. The world date determines which time-dependent records are available at the evaluation point, and the employee seat applies role- and owner-specific visibility rules. Materialization then derives the candidate-visible files, record views, communication, and tool state from those same inputs.

The materialized projection is task-agnostic. It is created before the request is applied and therefore contains information that may be irrelevant to the current assignment. This is the property used in the matched experiment: the full-projection condition exposes the complete employee-visible projection, while the curated condition selects a subset from that same projection without changing source bytes, company history, seat, or task.

Build validation occurs at both the canonical-state and materialized-workspace levels. Generated files are recorded with file-level digests in a pinned rendering environment. The staged employee workspace is checked against the expected projection before execution, and missing, changed, or unexpected files stop the run. Candidate-visible and verifier-only inventories are checked separately so that task scoring material cannot silently enter the employee workspace.

B.2 World instances

PharmaCo. PharmaCo is a fictional sterile-injectable organization covering regulatory, manufacturing, quality, supplier-quality, and stability work. Its canonical state includes shared and role-restricted records, employee roles, products at different lifecycle stages, and task-agnostic workplace services. The measured evaluation uses six employee seats. For the measured MSAT and QC tasks, the relevant seats can each access 775 files, most through a site-wide shared drive, while personal and confidential material remains role- or owner-restricted. All eight measured task overlays were written after the company state was fixed, and the final panel contains 72 criteria.

OncologyCo. OncologyCo is a dated regulatory-writing world derived from public oncology materials. The candidate state is frozen before approval-day information becomes available. Three assignments cover an indication and U.S. prescribing-information position, a Module 2.7.3 efficacy summary, and a Module 2.5 clinical overview. Later materials remain outside the candidate environment.

DiagnosticsCo. DiagnosticsCo is a dated diagnostic-device world derived from public PMA materials. The candidate receives a pre-approval snapshot. Four assignments cover intended-use language, labeling, an analytical module, and a clinical write-up, while later approval and supplement states remain verifier-side where needed for checking.

ClinicalSiteCo. ClinicalSiteCo is a dated clinical-site world derived from public trial materials. The candidate state is frozen during site startup, before later trial events. Four assignments require a site informed-consent form, an investigator-site-file index plus screening log, a CRC-to-PI IRB memo, and a coordinated startup-week package.

These three secondary worlds serve as implementation checks. They use different tasks, rubrics, and in some cases different model panels, so their model scores remain separate from the primary PharmaCo experiment. They test whether the same revision/date/seat/task construction can be instantiated in different organizational settings while preserving the construction contract.

C Rubric authoring and review protocol

C.1 Expert recruitment and independence

Domain practitioners involved in packet construction or evidence review have at least 10 years of relevant functional experience. They are paid a fixed hourly or project rate independent of model performance, packet content, and review conclusion. Some practitioners may serve in more than one

expert role. Review assignments exclude any packet, criterion, or evidence map that the practitioner personally constructed.

C.2 Rubric development and freezing

Rubric development begins only after the request, seat, snapshot, baseline revision, available surfaces, mutation policy, and deliverable path are frozen. Exploratory candidates then run on this fixed task. Reviewers inspect the outputs and canonical company records without seeing candidate identity; when needed, they may consult verifier-only truth to document substantive failures, unsafe unsupported claims, and disagreements that require expert judgment. Exploratory outputs are excluded from the formal matched runs.

These observations are converted into short criteria describing whether the work reaches the intended director-, vice-president-, or regulated-work signability bar. Criteria specify substantive content requirements without prescribing fixed wording or house style. Every criterion is linked to declared evidence and assigned either a deterministic check or a source-grounded semantic check. The rubric and grader configuration are then frozen before fresh measurement runs begin.

The truth specification stores source pointers, facts that must remain stable, unsafe unsupported claims, and questions requiring professional adjudication. It omits points, required prose, and candidate-facing hints and remains entirely on the verifier side of the evaluation. This separation lets the grader use evidence appropriate for verification while preserving the candidate information environment.

D Construction checks and mechanism diagnostics

The construction checks in Section 5.1 use a fixed fingerprint specification. Temporary host timestamps and container image identifiers are packaging metadata and are excluded from the organizational-state fingerprint; candidate-visible bytes, intended file dates, and the task-agnostic tool bundle, where present, are included. In the access audit, an exact hash match between a verifier-only artifact and a candidate-visible file triggers review before any leakage classification. Zero-byte files are ignored, intentionally candidate-visible copies are removed from the verifier-only inventory, and any remaining benign overlap must be recorded in a frozen exception list before the final run. One such overlap, a protocol copy, was recorded.

For evidence-map solvability, reviewer 1 judged 67 of 72 criteria solvable and reviewer 2 judged 69, with agreement on 66 of 72 (91.7%). Gwet's AC1 was 0.91 (Gwet, 2008); Cohen's κ was only 0.21 (cluster-bootstrap 95% interval, -0.05 to 0.42) because both reviewers labeled almost every criterion solvable (Cohen, 1960; Feinstein and Cicchetti, 1990). Mean Jaccard overlap between selected locators was 0.77 and the median was 0.78 (Artstein and Poesio, 2008). Disagreements and the one jointly unsolvable criterion were adjudicated; the latter was the environment defect we repaired. Table D1 summarizes the frozen-panel checks.

The illustrative reference world exercises the published high-level contract without using measured company data or production implementation code. Its conformance suite passed all 12 checks across seat projection, date projection, verifier sealing, grading-route declaration, and cross-surface agreement. Six additional adversarial cases deliberately violated the isolation boundary or state identity, and all six were rejected before grading. These deterministic checks demonstrate the published construction contract and do not substitute for the production-world validation reported in Table D1. Table D2 combines the reference-world checks so that the demonstration remains separate from the measured PharmaCo validation.

Several implementation failures motivated these checks. Renderer drift during development caused clean builds from the same source to produce different rendered or scanned PDFs on hosts with different LibreOffice and Poppler versions. Manifest verification now rejects such builds and retains committed reference digests; the reference manifest is write-protected during ordinary builds. World time created a separate failure mode because applying a historical clock directly to provider TLS caused certificate errors, while using the host clock exposed real time to the agent. The current proxy/sidecar design exposes world time to the agent while upstream certificate validation remains in

Table D1: Construction checks for the frozen eight-task, 72-criterion PharmaCo panel.

Property	Check	Result
Reproducibility	Clean rebuilds identical	3/3
	Repeated seat materializations identical	18/18
Projection consistency	Same-seat pre-task projections identical	2/2
	Staged and direct projections identical	8/8
Role isolation	Distinct seat fingerprints	6/6
	Authorized files present	Pass
	Unauthorized exposures	0
Verifier isolation	Seeded violations detected	4/4
	Unreviewed verifier overlaps	0
Evidence validity	Reviewer agreement on solvability	66/72
	Criteria requiring adjudication	7
	Criteria solvable after repair	72/72
	Environment repairs required	1
	Mean evidence-locator Jaccard	0.77
	Median evidence-locator Jaccard	0.78

Table D2: Reference-world conformance and seeded isolation checks.

Check	Failure exposed	Result
Conformance suite		
Seat projection	Different-permission roles receive the same evidence or an incorrect projection hash	4/4
Date projection	Future-record leakage or a date change that does not change the projection	3/3
Verifier sealing	Verifier-only evidence enters the candidate side or required verifier material is missing	2/2
Explicit grading route	Declared deterministic/semantic route does not produce the expected verdict	1/1
Cross-surface agreement	Filesystem and record adapter expose different paths or content	2/2
Seeded isolation adversaries		
Verifier-only file copied to candidate package	Candidate/verifier isolation	Detected
Post-date approval record copied to early projection	Date projection	Detected
Quality-only record copied to operations seat	Seat projection	Detected
Candidate revision changed	Revision identity	Detected
Projection hash replaced	Manifest integrity	Detected
Record-surface path set changed	Cross-surface agreement	Detected

a real-clock process. Missing provider registration produces a visible handshake failure, preventing silent continuation.

Grader routing must also be explicit. An earlier package assumed that a shared grader would automatically fall back to semantic checking, which allowed package configuration and the route actually used for scoring to diverge. Each criterion now declares its grading route: semantic criteria include a source declaration and deterministic criteria use a task-owned check. These failures are therefore surfaced as build or run errors instead of being absorbed into an agent score.

E MATCHED EXPERIMENT ADDITIONAL RESULTS

All results in this appendix come from the 192-run matched experiment and use the full projection and task-curated subset labels throughout. The main-text Figure 3 reports the standardized headline

effects. This appendix focuses on the raw decomposition, task-level variation, agent-level variation, audit details, and curator sensitivity, with the headline estimates reported once in the main text.

Execution configuration. Candidate model versions were `claude-opus-5`, `grok-4.6`, and `gpt-5.6-sol`. Opus 5 ran through Claude Code with medium reasoning effort, Grok 4.6 through Cursor CLI with the high preset encoded in the model configuration, and GPT-5.6 Sol through Codex with medium reasoning effort. The six native PharmaCo tasks used a 3,600-s candidate timeout and the two QC tasks used a 1,800-s timeout. WorkWorlds set no fixed tool-call limit and no additional context-window cap. Semantic grading used `claude-opus-5` after execution in a batch blinded to condition and curator packet; the released `data/configs/` files store the sanitized grader and agent settings, and the SHA-256 digests of those files are recorded in the study object.

Access labels. The automated labeler operates at the criterion–run level against the frozen evidence map. A sufficient set counts as accessed only when every source in at least one mapped sufficient set satisfies its access rule. A path appearing only in a listing, search index, autocomplete result, or filename match is insufficient. File access requires substantive returned content or a frozen content anchor, and structured-record access requires the mapped record plus the frozen fields specified by the access rule. Partial reads count only when the required anchor is present in the returned portion. When the retained trace cannot support a decision, the label is `unknown`; retained reasons are `trace_truncated`, `missing_tool_output`, `ambiguous_source_match`, `unsupported_surface_log`, or `other`. The evidence map enumerates sufficient sets and allows other valid paths to a correct answer, so `not accessed` and criterion pass can co-occur.

Grader audit. Domain-specialist reviewers judged 240 criterion executions (120 per construction condition) sampled across all eight tasks and three agents, with stratification by condition, task, and agent. Reviewers were blinded to model identity, condition, curator packet, and the grader's verdict, while seeing the same frozen candidate artifacts and verifier-side criterion definitions as the semantic grader. Raw agreement was 93.8% overall (Cohen's $\kappa = 0.85$), 92.5% under the full projection, and 95.0% under curation. In the released audit table, 11 disagreements are grader pass versus specialist fail and 4 is specialist pass versus grader fail; the imbalance is consistent with a slightly permissive semantic grader and is left unadjusted. The released `grader_audit_ratings.csv`, sampling manifest, and `grader_audit_summary.json` reproduce these counts from the frozen ledger. 5 criteria used deterministic grading (120 executions), and the remaining 67 used source-grounded semantic grading (1,608 executions).

Access-label audit. Before applying the automated access labeler to the full dataset, two coders independently labeled a stratified 20% sample containing accessed, not-accessed, and unknown cases; the released `access_label_coder_a.csv` and `access_label_coder_b.csv` are the independent submissions, and `access_label_audit_ratings.csv` stores both human labels plus the frozen automated label applied to the same rows. The sampling manifest records the seed, stratum, and inclusion probability. The audit covered 346 criterion executions, including 7 unknown labels. Inter-coder three-class raw agreement was 89.6% and Cohen's kappa was 0.73 (cluster-bootstrap 95% interval, 0.64 to 0.80). Coder A agreed with the automated label on 94.8% of rows and coder B on 93.6%. Separately, condition-adversarial recoding of the unknown access labels yielded access differences from 15.2 to 19.6 points. Uniformly recoding all unknown labels as accessed or all as not accessed yielded differences from 17.0 to 17.7 points.

Analysis settings. The primary model uses a logit link with condition and agent as covariates and an exchangeable working correlation within task. Standardized probabilities average predictions over the observed agent distribution, in which each agent contributes one third of criterion executions. The primary interval uses a 2000-draw task-cluster bootstrap with seed 20270908: each replicate resamples the eight tasks with replacement, refits the same logistic GEE, and re-standardizes the condition contrast. Each task–agent cell was collected in two four-run blocks, each containing two full-projection runs and one run from each curator packet. Before formal scoring, only adapter smoke tests were run, without inspecting task performance. Each attempt records the model, execution configuration, run order, and status; the released attempt ledger contains 205 attempts for 192 completed runs; 12 runs required at least one retry after a transient execution failure.

Table E1: Raw matched outcomes for the frozen 72-criterion panel. Access rates use only executions whose access label is observable. Pass given access is descriptive.

Outcome	**Full projection**	**Task-curated subset**
Criterion executions	864	864
Passed / executed	589/864 (68.2%)	664/864 (76.9%)
Accessed / observable	614/842 (72.9%)	767/848 (90.4%)
Unknown access labels	22 (2.5%)	16 (1.9%)
Pass given access	84.2%	82.7%
Joint access and pass state, counts		
Accessed, pass	517	634
Accessed, fail	97	133
Not accessed, pass	53	18
Not accessed, fail	175	63
Search burden, mean per run		
Recorded file/record inspections	28.6	15.2
Tool calls	34.1	23.2
Tool calls before first access	12.1	7.8

Table E2: Task-level matched results. N is criterion executions per condition for that task. Access uses only observable labels; the last column recomputes the pooled pass difference with that task removed.

	Criterion pass			**Evidence access**		
Task (N/condition)	**Full**	**Curated**	**Δ pp**	**Full**	**Curated**	**Pooled Δ pp, task held out**
Stability update (132)	64.4%	74.2%	+9.8	76.2%	88.5%	+8.5
Overwrap position (132)	68.9%	73.5%	+4.5	76.0%	92.2%	+9.4
Validation defense (108)	78.7%	65.7%	-13.0	78.5%	78.5%	+11.8
Authority response (168)	66.7%	78.0%	+11.3	71.0%	95.8%	+8.0
Records integrity (108)	66.7%	84.3%	+17.6	73.8%	92.4%	+7.4
Supplier-change audit (96)	59.4%	86.5%	+27.1	57.4%	92.6%	+6.4
Shelf-life assessment (60)	70.0%	73.3%	+3.3	77.6%	88.1%	+9.1
Qualification sweep (60)	75.0%	81.7%	+6.7	72.9%	93.2%	+8.8

E.1 Raw outcome decomposition

Table E1 gives the unstandardized counts underlying the main analysis. The standardized GEE estimates in the main text differ slightly from these raw proportions because the model accounts for repeated criterion executions within tasks and includes agent as a fixed effect. The joint access/pass counts allow the descriptive pass-given-access quantity to be read alongside the two prespecified outcomes.

The packet-size statistics in Table 1 count unique curator-selected sources before execution. The inspection statistic in Table E1 is taken from the run trajectory and is reported separately from packet size.

E.2 Variation across tasks and agents

The effect varied across the task panel. Table E2 reports each task separately. Its last column recomputes the pooled pass difference with each task removed in turn. Table E3 reports the comparison by agent.

E.3 Sensitivity to curator choice

The two practitioners selected different source sets, so we also analyzed each curator's packets separately against the full projection. Curator 1's packets passed 334/432 criterion executions (77.3%), corresponding to a +9.1-point raw difference from the full projection (95% task-cluster bootstrap interval, 2.3 to 14.7); 6 of 8 task contrasts were positive. Curator 2's packets passed 330/432

Table E3: Agent-level matched results. Each agent contributes 288 criterion executions per condition.

Agent	Full pass	Curated pass	Pass Δ pp	Access Δ pp
Claude Opus 5	229/288 (79.5%)	247/288 (85.8%)	+6.2	+12.8
Grok 4.6	188/288 (65.3%)	225/288 (78.1%)	+12.8	+17.5
GPT-5.6 Sol	172/288 (59.7%)	192/288 (66.7%)	+6.9	+22.3

criterion executions (76.4%), corresponding to a +8.2-point raw difference from the full projection (95% task-cluster bootstrap interval, -0.8 to 17.2); 7 of 8 task contrasts were positive. Curator-specific reversals occurred for validation defense under curator 1 (-8.3 points), shelf-life assessment under curator 1 (-3.3 points), validation defense under curator 2 (-17.6 points). Both curator-specific analyses had positive pooled pass differences, although Curator 2's interval included zero.

Packet construction is summarized once in main-text Table 1; the appendix focuses on the additional outcome and sensitivity analyses.

F Reproducibility and release boundary

The release artifact contains the manuscript, figures, de-identified criterion- and run-level result ledgers, aggregate analysis outputs, code reproducing the reported statistical analyses, and a simplified illustrative reference world, abstracted from the production codebase, with conformance and isolation tests. The illustrative code implements only the high-level revision/date/seat/task construction contract on a fully fictional world. It is provided so that the ordering and isolation rules described in the paper can be inspected and executed without access to the measured company.

The illustrative code is not the production WorkWorlds implementation and does not contain the proprietary world builder, production materialization and permission logic, execution infrastructure, tool integrations, or PharmaCo files. The measured PharmaCo organizational corpus, restricted source materials, held-out verifier evidence, and raw trajectories containing organizational content are also not released. The released ledgers contain de-identified run- and criterion-level outcomes used for the reported analyses, while the production infrastructure and organizational content remain outside the artifact.

This boundary supports reproducibility of the experimental protocol, released outcomes, and statistical analyses, together with an executable illustration of the published construction contract. It does not provide implementation-level reproduction of the proprietary production infrastructure.

G Comparison with Related Workplace Evaluations

Recent workplace-agent benchmarks increasingly define structure above the individual task, including reusable enterprise services, professional project worlds, worker workspaces, and organizational metadata. These systems differ in what state is defined before a task is introduced, how employee or role-specific access is represented, and what consistency is required across different assignments. Table G1 summarizes the benchmarks most directly related to the construction problem addressed by WorkWorlds. We focus on properties that affect whether multiple evaluation instances can be interpreted as occurring within the same underlying workplace.

WorkArena and OSWorld define execution environments at the enterprise-web and desktop layers: WorkArena instantiates knowledge-work tasks in ServiceNow, while OSWorld instantiates reproducible desktop states across applications (Drouin et al., 2024; Xie et al., 2024). ClawMark adds multi-day service environments whose state can change between turns (Meng et al., 2026). These systems specify the interactive state in which an agent acts; the comparison below focuses on the persistent workplace structure from which multiple assignments are derived.

TheAgentCompany and EnterpriseOps-Gym define reusable company or enterprise infrastructure, while individual task runs are separately instantiated or reset (Xu et al., 2025; Malay et al., 2026). APEX-Agents moves the reusable unit to a professional project world: domain experts construct a data-rich world before multiple tasks are authored from its existing files and tools (Vidgen et al., 2026). Workspace-Bench defines large workspaces associated with worker profiles and evaluates

Table G1: Comparison of workplace structures defined above individual evaluation tasks.

Benchmark	Structure above task	Role/access structure	Relationship across tasks
TheAgentCompany (Xu et al., 2025)	Reusable simulated-company services and data	Company accounts, services, and simulated coworkers	Tasks are instantiated separately from a common company setup; task-dependent services reset per run
EnterpriseOps-Gym (Malay et al., 2026)	Reusable enterprise schemas, databases, tools, and policies	Policy-constrained tool access across eight enterprise domains	Expert-written tasks run in separately resettable enterprise instances
APEX-Agents (Vidgen et al., 2026)	Data-rich professional project world	Project context; no employee-specific role projection	Multiple tasks are authored from files and tools already present in the same project world
Workspace-Bench (Tang et al., 2026)	Large workspace associated with a worker profile	Worker profile defines the occupational context	Tasks require locating, reasoning over, and modifying dependencies within the workspace
EnterpriseBench (Vishwakarma et al., 2025)	Enterprise sandbox with organizational metadata	Employee hierarchy, dynamic role-based access control, and persona-specific access	Tasks are generated from organizational metadata and persona-relevant enterprise context
WorkWorlds	**Versioned organizational state**	**Employee seat materialized through declared roles and permissions**	**Revision, date, and seat fix the pre-task projection across assignments**

tasks that require agents to identify and manipulate dependencies within those workspaces (Tang et al., 2026). EnterpriseBench adds explicit organizational structure through employee hierarchy, enterprise data, organizational metadata, and dynamic role-based access control, and uses this context to generate internally consistent tasks (Vishwakarma et al., 2025).

WorkWorlds uses a versioned organization as the state above individual tasks. Organizational records, employees, permissions, products, and prior events are fixed independently of a particular assignment. Revision and date select the relevant organizational state, employee seat determines the role-visible projection, and the task is applied only after that projection has been materialized. This allows cross-task consistency to be checked directly by requiring the same revision, date, and employee seat to reproduce the same pre-task workplace.

The comparison shows a progression in the unit defined above individual tasks. TheAgentCompany and EnterpriseOps-Gym reuse enterprise infrastructure; APEX-Agents reuses a professional project world; Workspace-Bench defines worker-level workspaces; and EnterpriseBench grounds tasks and access in organizational data and hierarchy. WorkWorlds adds an explicit mapping from a versioned organization to employee- and time-specific pre-task environments. As a result, whether two assignments occur within the same organizational state is represented by the evaluation construction itself and can be checked before agent execution.