DWBENCH: EVALUATING AI AGENTS AS DIGITAL WORKERS OVER ENTERPRISE GRAPH DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Enterprise knowledge work is broader than writing code: a digital worker must locate the right thread in a noisy inbox, reconcile facts scattered across emails, files, meetings, and chats, then produce a report, spreadsheet, or slide deck. We present **DWBench**, an agent-interface-agnostic benchmark for grounded retrieval and Office-artifact generation over ego-centric, permission-scoped enterprise graphs. We release two synthetic enterprise tenants (biotechnology and financial advisory) comprising 80 employees and 6,110 content units across email, chat, meetings, and 13 file types per tenant. 474 tasks spans single-fact retrieval, multi-hop reasoning, Word, Excel, and PowerPoint production, with assertions linked to supporting records for the evaluation. We also implement a Model Context Protocol (MCP) tool to perform the search under enterprise permission management. We evaluate three harnesses: GitHub Copilot (GHCP), OpenCode, and a reference agent (RefAgent), across 11 model-effort settings. The experiments comprise 15K executions across three harnesses, two tenants, and five task types using frontier models, including GPT-6-Astra, Claude-Opus-5 and Grok 4.6. They include paired-harness comparisons and model-scaling studies. Results show that mean assertion pass rate varies with the model, harness, and task type: Word generation attains mean assertion pass rates of 87.8–92.8%, whereas multi-hop search remains a bottleneck at 54.9–60.1%. GHCP achieves the highest mean assertion pass rate, exceeding OpenCode and RefAgent by 1.5 and 2.2 percentage points, respectively, while significant GHCP advantages over OpenCode in task completion rate at the 0.75 assertion threshold (a query is complete when at least 75% of its assertions are satisfied) are concentrated in specific models. Our findings cover model performance, harness impact and evaluation robustness across enterprise environments. Human validation supports our findings.

1 INTRODUCTION

Large language models are moving from direct chat assistants to versatile agents that operate applications—reading email inboxes, browsing shared drives, and editing documents (Xie et al., 2024; Drouin et al., 2024). This shift promises to automate *enterprise knowledge work* beyond coding (Drouin et al., 2024; Patwardhan et al., 2026). Such work often requires locating an email thread, reconciling it with meeting and chat records, and assembling a report, spreadsheet, or slide deck. This *search-then-produce* loop couples grounding over a noisy, permission-scoped organizational graph data (such as Microsoft Graph[1] and Google Workspace[2]) with synthesis into an editable deliverable (Choubey et al., 2025; Abaskohi et al., 2026; Patwardhan et al., 2026). Yet reliable, end-to-end performance remains largely unmeasured (Tang et al., 2026; Malay et al., 2026).

Backgrounds & Related Works. Existing agent benchmarks each capture only a slice of enterprise digital work (Table 1). SWE-bench (Jimenez et al., 2024) combines repository retrieval with code-patch generation but does not model an enterprise tenant. GAIA (Mialon et al., 2024), WebArena (Zhou et al., 2024), and τ-bench (Yao et al., 2025) emphasize short answers or state-changing interaction rather than substantive work products. WorkBench (Styles et al., 2024) and

[1]Microsoft Graph overview: https://learn.microsoft.com/en-us/graph/overview
[2]Google Workspace https://developers.google.com/workspace

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.

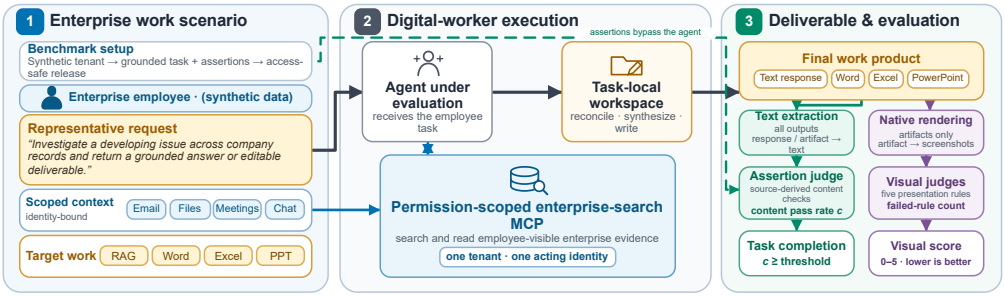


Figure 1: DWBench evaluates enterprise search-then-produce work. (1) A role-specific employee request draws on permission-scoped email, files, meetings, and chat and spans retrieval-augmented generation (RAG), Word, Excel, and PowerPoint tasks. (2) The agent uses an identity-bound MCP interface and task-local workspace to retrieve, reconcile, and write. (3) Source-derived assertions evaluate normalized text from the final work product: response text is used directly, while Office artifact content is extracted. The resulting pass rate c defines task completion at a threshold (0.75 in this study). In parallel, artifact screenshots enter a separate five-rule visual-quality evaluation whose score is the failed-rule count (0–5; lower is better). Full description: Appendix E.

OfficeBench (Wang et al., 2024) model workplace workflows, with OfficeBench spanning multiple applications and office-file outputs. HERB (Choubey et al., 2025) evaluates Deep Search over linked enterprise artifacts, whereas DRBench (Abaskohi et al., 2026) combines public and private retrieval with citation-bearing structured reports; neither primarily evaluates editable office documents. GDPval (Patwardhan et al., 2026) evaluates professional work products across 44 occupations, usually from task-local prompts and reference files rather than a persistent enterprise organization. SlidesBench, introduced with AutoPresent (Ge et al., 2025), evaluates slide generation, including settings with image search. Workspace-Bench (Tang et al., 2026) evaluates multiple harnesses over large role-specific workspaces with explicit file-dependency graphs and dependency-aware rubrics. EnterpriseOps-Gym (Malay et al., 2026) exposes Model Context Protocol (MCP)-backed (Anthropic, 2024) enterprise applications over interconnected relational tables with access controls and SQL state verification. According to Table 1, no prior benchmark in the table clearly combines all six, motivating an evaluation setting that joins enterprise graph grounding with permission-scoped organizational context, cross-source retrieval, and editable deliverable production under one scoring framework.

We introduce the Digital Worker Benchmark (DWBench), an agent-agnostic evaluation platform paired with a data-generation pipeline. The platform separates a *fixed evaluation core*—a simulated enterprise-search environment, an assertable task specification, and a family of rubric scorers—from a *swappable agent driver*, enabling matched comparisons of complete, as-configured systems under a common enterprise interface (Figure 1). The environment is published as a single MCP server, turning the enterprise-search stack into an agent-agnostic contract that a built-in reference agent and real coding agents consume through system-specific drivers without changing the environment contract. The data-generation pipeline simulates an enterprise tenant into an ego-centric graph of emails, files, meetings, and chats, and then authors an assertable task set directly from that graph, linking each task to supporting records and emitting a machine-checkable content rubric. This design makes task provenance inspectable and evaluates whether outputs are *grounded* in information that must be retrieved from the underlying enterprise data. Generated Office artifacts are also evaluated from native renderings with vision-language judges. While prior work covers only subsets of capabilities in Table 1, DWBench brings these capabilities together in one benchmark.

Contributions. This paper makes three contributions:

- We formalize evaluating agents as digital workers over an ego-centric, permission-scoped enterprise graph, coupling chatbot retrieval, deep research, and artifact generation in one benchmark. Table 1 makes this scope concrete relative to prior work along the six dimensions above.
- We build an end-to-end platform: an agent-agnostic MCP enterprise-search environment; a data-generation pipeline for grounded tenants, queries, and rubrics; and a multi-view evaluator combining a shared content-assertion judge for responses and extracted artifact text, native-rendering visual judges for Office artifacts, and blinded human validation.

Table 1: DWBench versus representative benchmarks along digital-worker dimensions.

Benchmark	Enterprise scenarios	Enterprise graph grounded	Artifact generation	Agent-agnostic	Evidence-linked eval	Retrieval + generation
SWE-bench (Jimenez et al., 2024)	partial	✗	✓	✓	partial	✓
GAIA (Mialon et al., 2024)	✗	✗	✗	partial	partial	✗
WebArena (Zhou et al., 2024)	partial	✗	✗	✓	partial	partial
τ-bench (Yao et al., 2025)	partial	✗	✗	partial	partial	✗
WorkBench (Styles et al., 2024)	✓	partial	partial	partial	partial	partial
HERB (Choubey et al., 2025)	✓	✓	✗	partial	✓	✗
DRBench (Abaskohi et al., 2026)	✓	✓	partial	✓	✓	partial
GDPval (Patwardhan et al., 2026)	partial	✗	✓	partial	partial	partial
Workspace-Bench (Tang et al., 2026)	✓	✗	✓	✓	✓	✓
EnterpriseOps-Gym (Malay et al., 2026)	✓	✓	partial	✓	partial	partial
SlidesBench (AutoPresent) (Ge et al., 2025)	partial	✗	✓	partial	partial	partial
OfficeBench (Wang et al., 2024)	✓	✗	✓	partial	partial	✓
DWBench (ours)	✓	✓	✓	✓	✓	✓

Legend and scope. ✓ denotes clear benchmark-level support, *partial* denotes support limited to a subset or weaker interpretation, and ✗ denotes absence. "Enterprise graph" requires a heterogeneous organizational graph spanning communications or linked business records; a file-only dependency graph does not qualify. "Artifact generation" includes substantive code or office work products. "Evidence-linked evaluation" ties task checks or rubrics to source material.

- We instantiate DWBench on two contrasting tenants and run a model × tenant × harness study with a self-made reference agent (RefAgent) based on ReAct (Yao et al., 2023), GitHub Copilot (GHCP), and OpenCode. The analysis preserves the full five-way task decomposition, quantifies model scaling and paired harness effects, audits operational events and evaluator sensitivity.

2 METHODOLOGY

DWBench comprises three components: i) an *enterprise-search environment* that provides a uniform interface to all agents (Section 2.1); ii) a *data-generation pipeline* that produces the grounding data and the assertable eval sets of queries and rubrics (Section 2.2); iii) *content and visual presentation evaluators* that score agent outputs against those rubrics (Section 2.3).

2.1 ENTERPRISE-SEARCH ENVIRONMENT AND THE MCP INTERFACE

The core design principle of DWBench is the separation of an enterprise-search environment from the agent workspace (see Figure 1). There are two reasons for this separation: i) due to the security concerns, enterprise users often need to enforce access controls before exposing information to an agent; ii) many agents rely heavily on the string-matching tools to traverse information within the workspace, which is inefficient given the scale and diversity of enterprise data.

The enterprise-search environment holds a tenant graph (Section 2.2), which represents the access relationships between information resources and users within the tenants. Agents can query information via a Model Context Protocol (Anthropic, 2024) server that exposes 11 tools for identity inspection, modality-specific search, full-item reads, and within-document keyword lookup. When making artifacts, agents are provided with a separate per-case working directory. Every agent consumes the same tool contract through its driver. Although agents share the same contract, we have observed that their capabilities in tool usage, information synthesis, and orchestration remain significant factors influencing the final results.

For data permission management, agents delegated by a user can observe only the emails, chats, and meetings that the user authored, received, or attended. By contrast, files are shared at the tenant level, so the file corpus is more diverse and noisier than the permission-scoped communications. We quantify the resulting ego-centric visibility of the communications graph in Section 3. Further implementation details of the enterprise-search environment appear in Appendix A.4.

2.2 DATA GENERATION: GROUNDING DATA AND EVAL SETS

DWBench has a two-stage data-generation pipeline (summarized in the benchmark-setup strip of Figure 1). The first stage produces the *grounding data*: a tenant simulation rolls a synthetic organization forward over a long horizon, emitting business events that materialize as emails, files,

meeting transcripts, and chat threads with authorship and access-control edges. The second stage produces the *eval set*: an LLM creates tasks from a subset of materials accessible to a sampled tenant user, so that each task is grounded in retrievable evidence. Every task is then checked for groundedness against its supporting evidences to ensure the query quality. We construct five task types in two families: (i) `chatbot`, where the agent answers the question in a chat window — subdivided into *retrieval*, built on a single supporting record, and *multi-hop*, built on multiple records; (ii) `artifact` (Word/Excel/PPTX), where the agent must search for the supporting materials and produce an editable deliverable. DWBench contains 474 tasks across the two tenants: 174 chatbot-retrieval, 131 chatbot-multihop, 34 Word, 66 Excel, and 69 PowerPoint. Appendix A.1 gives the full composition and a representative example of each task type.

Additionally, the LLM generates content assertions for evaluating the answers. To ensure that these assertions distinguish grounded answers from unsupported ones. We instruct multiple advanced models to write gold answers with the supporting materials given explicitly and bluff answer without access to any material. We hereby consider using those assertions that could pass gold answers but might fail by bluff ones. As a result, our assertion score shows clear margins across models and harnesses. More details of the content assertions are provided in Appendix A.3.

2.3 Evaluation: content and visual presentation scoring

Per-task scores and task completion rates. DWBench evaluates the answer quality by grading every task against content assertions (Figure 1, stage 3). We adopt LLM judge based on GPT-5.4-mini. To confirm the quality of evaluation, we re-judge the outputs against assertions using Claude Haiku-4.5 and human validation. The per-task assertion pass rate $c \in [0, 1]$ is the fraction of the task's assertions satisfied in the final chatbot response or in text extracted from the generated artifact with MarkItDown[3]. A task without a gradeable output is a non-completion with assertion pass rate 0. A task is complete when $c \geq 0.75$. We also conduct a sensitivity analysis over this threshold. A configuration's task completion rate is the fraction of its evaluation tasks that meet the threshold. Appendix A.5 gives the judge prompt, parsing, and aggregation procedure.

Visual presentation evaluation. We also score artifact presentation independently of assertion correctness. Each Word, Excel, or PowerPoint artifact is rendered through native Microsoft Office and supplied in full, as an ordered page-image sequence, to two vision-language judges, Claude Opus-4.6 and GPT-5.5. They mark failures of five binary rules—legibility, intactness, visible structure, consistency, and interpretable colour encoding—whose sum is a failed-ruble count from 0 to 5 (lower is better). Note that, visual presentation evaluation here provides *format-specific comparative diagnostics* of presentation failures rather than a holistic measure of artifact quality. Appendix A.6 gives the rendering protocol, prompt, agreement analysis, and interpretation limits.

3 Enterprise Tenants

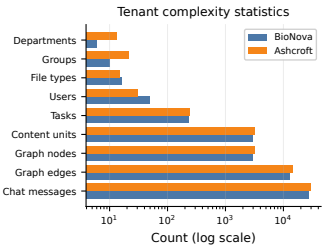
We instantiate DWBench on two independent synthetic tenants of different domains. Both are simulated over a ∼100-day horizon but are structurally different—**BioNova**, a 50-person biotechnology R&D firm, and **Ashcroft**, a 30-person financial-advisory firm—so that results are not tied to a single organizational style.[4] Figure 2 summarizes the two tenants along five axes of complexity: organizational scale, modality heterogeneity, corpus and graph size, temporal noise, and permission scoping. Comprehensive per-tenant statistics are tabulated in Appendix A.2.

3.1 Scale and heterogeneity

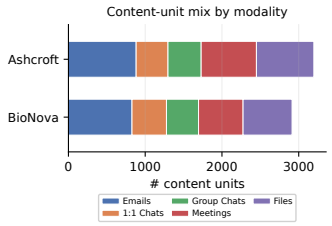
Each tenant materializes into an enterprise graph of roughly 3,000 content units and 12,000–14,000 edges. Content is spread across five content modalities and tens of distinct file types with no uniform schema (Figure 2b,c). Many multi-hop and artifact tasks draw facts from multiple records and modalities. For example, a decision can be announced in a meeting, but further refined over chat,

[3] MarkkItDown 0.1.7: https://pypi.org/project/markitdown/0.1.7/

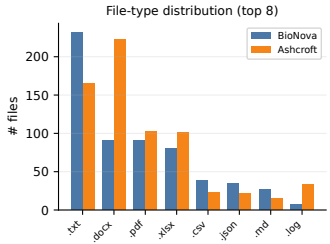
[4] The LLM-generated tenant names might coincide with real company names, but the generated content is not modeled on any real company.



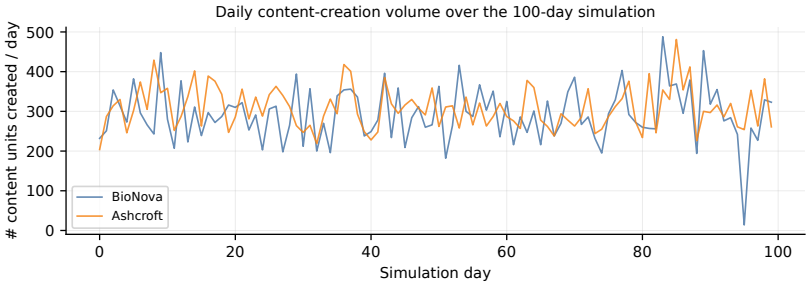
(a) Tenant complexity statistics



(b) Content-unit mix by modality



(c) File-type distribution (top 8)



(d) Daily content-creation volume over the 100-day simulation

Figure 2: Tenant construction. (a) Scale; (b) modality mix; (c) file types; (d) temporal volume. Full interpretation and statistics: Appendices E and A.2.

and eventually formalized in an attached file. This cross-record dispersion is a plausible contributor to the multi-hop retrieval difficulty documented in Section 4.

3.2 Advancing storylines and irrelevant events

Content accrues continuously over the simulated horizon rather than arriving as a clean batch (Figure 2d). Business events unfold as overlapping storylines (product decisions, compliance reviews, supplier escalations), so at any point the corpus contains in-progress threads, superseded versions, and large volumes of task-irrelevant "distractor" content. An agent answering a task must therefore separate the relevant thread from a continuously growing, bursty backdrop.

3.3 Permission scoping and the ego-centric slice

Communications are permission-scoped, so an agent delegated by a user can only observe the user's emails, chats, and meetings. By contrast, files are shared at the tenant level and are readable by any identity. Within the communications graph, the mean per-user visible fraction is just 8.1% (BioNova) and 13.3% (Ashcroft). The full per-user distribution shown in Figure A.1 confirms both tenant means fall below 15% on communications. Agents operate over a *per-user* ego-centric slice rather than the global tenant. Communications retrieval is therefore more focused with respect to the user's scoped view, while the shared file corpus provides broader document access but also contains more irrelevant content.

4 Experiments

Experiment configuration. In this paper, we use DWBench to compare multiple model families and agent harnesses. Specifically, we benchmark Claude-Opus-4.7, 4.8, and 5 from Anthropic; GPT-5.4, GPT-5.5, GPT-5.6 Sol, and GPT-6-Astra from OpenAI; and Grok 4.5, 4.6 from xAI to study their capabilities and model evolution. All models are evaluated using the highest reasoning effort setting. We further investigate the impact of model size by comparing GPT-5.6-Terra and GPT-5.6-Luna against GPT-5.6-Sol. Additionally, we study the effect of reasoning effort by comparing GPT-6-Astra under low, high and max reasoning effort settings. For agent harnesses, we

Table 2: Per-task *assertion pass rate* (fraction of a task's assertions satisfied), mean ± standard deviation, by the five query types—two chatbot search types (single-fact *Retr.* and *Multi.*-hop) and three artifact types (*Word*/*Excel*/*PPTX*)—grouped by tenant. Large standard deviations reflect a bimodal per-task distribution. Opus-5, GPT-6-Astra, and Grok-4.6 are shown; complete per-task, overall-completion, and paired results appear in Appendix Tables B.1, B.2, and B.3. Boldface marks the single highest unrounded mean within each tenant and task column.

		Chatbot		Artifact		
Arm	Model	Retr.	Multi.	Word	Excel	PPTX
BioNova (pharma R&D)						
RefAgent	Claude-Opus-5	0.80 ± 0.38	0.50 ± 0.43	0.96 ± 0.05	0.77 ± 0.30	0.82 ± 0.27
RefAgent	GPT-6-Astra	0.62 ± 0.39	0.71 ± 0.30	0.95 ± 0.06	**0.79** ± 0.33	0.82 ± 0.27
RefAgent	Grok-4.6	0.80 ± 0.36	0.65 ± 0.34	**0.99** ± 0.03	0.70 ± 0.36	0.75 ± 0.30
GHCP	Claude-Opus-5	0.83 ± 0.35	0.48 ± 0.44	0.97 ± 0.05	0.75 ± 0.35	0.77 ± 0.33
GHCP	GPT-6-Astra	0.75 ± 0.36	**0.74** ± 0.30	0.97 ± 0.04	0.79 ± 0.30	**0.84** ± 0.24
GHCP	Grok-4.6	0.81 ± 0.36	0.55 ± 0.39	0.98 ± 0.04	0.73 ± 0.32	0.80 ± 0.26
OpenCode	Claude-Opus-5	**0.85** ± 0.34	0.54 ± 0.44	0.98 ± 0.04	0.76 ± 0.37	0.76 ± 0.32
OpenCode	GPT-6-Astra	0.74 ± 0.36	0.70 ± 0.31	0.97 ± 0.05	0.76 ± 0.36	0.79 ± 0.27
OpenCode	Grok-4.6	0.78 ± 0.38	0.49 ± 0.39	0.98 ± 0.07	0.71 ± 0.39	0.80 ± 0.26
Ashcroft (financial advisory)						
RefAgent	Claude-Opus-5	**0.96** ± 0.16	0.78 ± 0.28	0.98 ± 0.04	0.89 ± 0.19	0.83 ± 0.22
RefAgent	GPT-6-Astra	0.63 ± 0.37	0.77 ± 0.26	0.95 ± 0.07	0.85 ± 0.24	0.86 ± 0.23
RefAgent	Grok-4.6	0.87 ± 0.30	0.70 ± 0.34	0.88 ± 0.24	0.71 ± 0.41	0.80 ± 0.28
GHCP	Claude-Opus-5	0.91 ± 0.26	0.73 ± 0.29	0.97 ± 0.05	0.81 ± 0.31	0.85 ± 0.22
GHCP	GPT-6-Astra	0.74 ± 0.36	**0.81** ± 0.26	0.97 ± 0.08	**0.90** ± 0.20	0.87 ± 0.22
GHCP	Grok-4.6	0.93 ± 0.25	0.63 ± 0.36	0.96 ± 0.07	0.78 ± 0.31	0.80 ± 0.28
OpenCode	Claude-Opus-5	0.93 ± 0.24	0.74 ± 0.26	**0.99** ± 0.03	0.84 ± 0.25	**0.88** ± 0.21
OpenCode	GPT-6-Astra	0.79 ± 0.32	0.78 ± 0.23	0.95 ± 0.08	0.84 ± 0.27	0.84 ± 0.25
OpenCode	Grok-4.6	0.92 ± 0.25	0.57 ± 0.37	0.91 ± 0.24	0.72 ± 0.36	0.76 ± 0.31

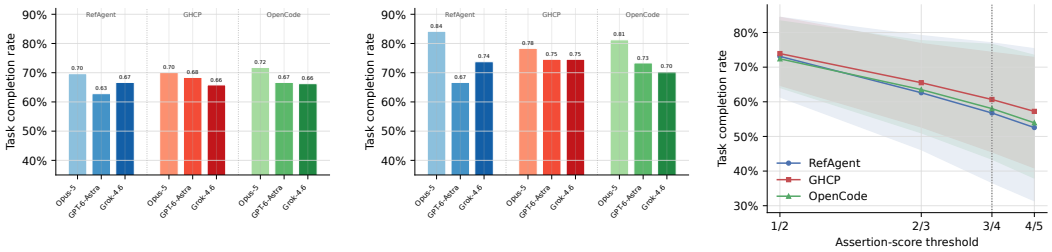
implement a RefAgent which is a controlled ReAct harness (Yao et al., 2023), adopt the proprietary GitHub Copilot framework (GHCP)[5], and use the opensource framework OpenCode[6]. Each arm is evaluated using the same tenant, eval set, MCP interface, search environment, and evaluator, while planning, tool orchestration, CLI behavior, and provider policies are treated as the components of the measured system. The overall experiment setup comprises 11 model configurations across three agent harnesses and two additiona effort configurations on RefAgent and GHCP.

Statistical analysis. We present harness and model comparisons, model- and effort-scaling analyses, task-type and completion-threshold sensitivity, tool-use and provider-failure analyses, and visual, human, and re-judge robustness studies. Paired tests and bootstrap confidence intervals quantify uncertainty where applicable.

4.1 Results

Content assertion results. Table 2 summarizes mean per-task assertion pass rates for Opus-5, GPT-6-Astra, and Grok-4.6 across both tenants and all three agent harnesses, whereas Figure 3 reports task completion rate at the 0.75 threshold for the same frontier-model subset, all based on GPT-5.4-mini judge. Word generation is strongest overall and multi-hop search is weakest, while the leading model–harness combination varies across tenants and capabilities. Appendix Tables B.1–B.3 report the complete per-task, overall-completion, and paired results for all 11 settings. We also conduct two sets of judge-robustness evaluation: an intra-model re-judge based on the same model and using the same text responses/artifacts changes 4.4% task-completion outcomes, while the inter-model re-judge between GPT-5.4-mini and Haiku-4.5 on 1,000 randomly sampled outputs yields 92.2% outcome agreement. Appendix B documents experiment protocols and complete results, Appendix C.2 reports provider filtering, and Appendix D.1 gives the judge-robustness results.

[5]GitHub Copilot: https://www.npmjs.com/package/@github/copilot/v/1.0.81-14
[6]OpenCode: https://www.npmjs.com/package/opencode-ai/v/1.18.21



(a) BioNova task completion rate (b) Ashcroft task completion rate (c) Task-completion sensitivity

Figure 3: Agent capability and threshold robustness. (a,b) Task completion rate at the 0.75 threshold by tenant; colors group results by harness. (c) Mean and range of task completion rate as the assertion threshold changes. Complete per-task, overall-completion, and paired model results appear in Appendix Tables B.1, B.2, and B.3.

Visual presentation results. For the visualization evaluation, we sample 155 tasks across all task types resulting in 1,391 successfully rendered artifacts generated by combinations of Claude-Opus-5, GPT-6-Astra, Grok 4.6 and three agent harnesses. We employ two VLM-based judges: Claude-Opus-4.6 and GPT-5.5 to independently evaluate the artifacts using the same rubrics. Both judges rank GPT-6-Astra best on Word and Claude-Opus-5 best on PowerPoint, while Grok-4.6 perform worst on slides. Rankings for Excel differ across judges. For example, under the Claude-Opus-4.6 judge, the mean failed-rule counts (0–5; lower is better) were 0.16/0.75/0.75 for Astra/Opus-5/Grok on Word, and 0.51/0.77/1.19 for Opus-5/Astra/Grok on PowerPoint. In contrast, Excel scores largely overlapped, with mean failed-rule counts ranging from 2.07 to 2.28. Overall inter-judge agreement was Cohen's $\kappa = 0.29$ across 1,374 shared artifacts. Therefore, we treat these scores as complementary, comparative diagnostics rather than absolute defect rates of the artifacts. Additional details and results are given in Appendix A.6.

Human validation results. We ran a blinded, stratified study of benchmark realism, eyes-on output quality, and automatic assertion evaluation: 50 queries—five per tenant $\times$ task-type stratum—were assigned to 10 jobs, each containing one query per task type and three anonymized arm outputs (150 total). Reviewers rated scenario fit and grounding need, then output quality from the original editable Office files, and only afterward inspected automatic assertion verdicts; all ratings used anchored 0–5 scales and extracted text served only as navigation. Human ratings support the enterprise and content claims: enterprise fit averages 4.92/5 and answerability without correct grounding 0.08/5; content verdicts agree with GPT-5.4-mini on 90.0% of all ratings. For Pass/Fail judgments, Cohen's kappa is 0.71, indicating strong agreement between human and LLM judgments. Generated Office artifacts average 4.31/5 in eyes-on quality, with modest alignment to the two-judge visual score (Spearman rank correlation 0.34). Appendix A.7 gives the full protocol and results.

4.2 Analysis

Multi-hop search is the bottleneck. Based on the complete 11 model results as shown in Table B.1, multi-hop task completion rate is 45.4% (RefAgent), 41.6% (GHCP), and 39.3% (OpenCode). Single-item retrieval task completion rate is 59.3%, 68.5%, and 63.3%. By contrast, Word task completion rate is highest (89.6–96.0%), while Excel (52.8–59.1%) and PowerPoint (59.7–62.8%) are intermediate. The pattern is therefore more specific than a general retrieval/generation dichotomy: reconciling evidence across multiple enterprise records is hardest, whereas artifact difficulty depends strongly on format.

Task completion rankings are stable across assertion pass rate thresholds. We recompute task completion rate at assertion pass rate $c \geq 1/2$, $2/3$, $3/4$, and $4/5$. Configuration rankings relative to the $3/4 = 0.75$ baseline remain stable: Spearman $\rho = 0.945$, 0.988, and 0.991 at $1/2$, $2/3$, and $4/5$. Figure 3c and Appendix Tables D.3–D.2 give the complete sweep.

Both harness and model choices are associated with material differences. Across the 11-model complete results, task completion rate for GHCP exceeds RefAgent by $+3.9$ points (95%

CI $[+2.4, +5.4]$), OpenCode exceeds RefAgent by $+1.3$ points ($[-0.2, +2.7]$), and GHCP exceeds OpenCode by $+2.6$ points ($[+1.5, +3.8]$). Figure 4 compares Claude-Opus version, GPT-5.6 model-size, and GPT-6-Astra thinking-effort scaling. Claude-Opus task completion rate rises from 4.7/4.8 to 5 in every harness (60.3/64.3/77.0% in RefAgent, 62.9/62.9/74.3% in GHCP, and 59.7/64.1/76.6% in OpenCode). Task completion rate along the flagship GPT sequence rises from GPT-5.4 to GPT-5.6-Sol for RefAgent (36.7 to 49.4%) and GHCP (45.6 to 61.2%), while OpenCode is non-monotonic (48.1/43.9/51.9%); Appendix Figure B.1 gives this comparison. Within GPT-5.6, task completion rate is highest for the largest Sol in GHCP and OpenCode, whereas RefAgent's Luna/Terra/Sol rates are close (46.6–49.4%). GPT-6-Astra task completion rate rises from low to high effort in RefAgent and GHCP, while high-to-max gains are smaller. Results characterize model version, size, effort, and harness patterns without implying one universal ordering. Appendix Table B.5 reports additional paired version and effort contrasts. Appendix B.4 gives the detailed table.

GHCP–OpenCode differences are model-dependent. The all-task GHCP–OpenCode task-completion interval excludes zero for GPT-5.6 Sol ($+9.2$ points), GPT-5.6 Luna ($+8.2$), GPT-5.5 ($+7.8$), and GPT-5.6 Terra ($+5.7$). The remaining seven model intervals cross zero. The search-only GPT-5.4/GPT-5.5 analogue of the GHCP–OpenCode contrast is $+4.0$ points ($[+0.1, +8.0]$) over 610 paired model–case executions. Appendix Table B.3 reports all model-level effects.

Tool depth is diagnostic rather than causal. Mean total observed calls per search task are 12.6 for RefAgent, 15.0 for GHCP, and 14.5 for OpenCode. Artifact generation tasks require substantially more tool use, averaging 19.9, 22.7, and 23.3 calls, respectively. The production harnesses, GHCP and OpenCode, make approximately 13 MCP calls on chatbot tasks and 17 on artifact tasks. Task completion rate peaks at intermediate call-count bins and declines in the 26+ bin, where difficult tasks, retries, and timeouts become more prevalent. Figure 5 and Table C.1 in the appendix characterize trace depth against the performance.

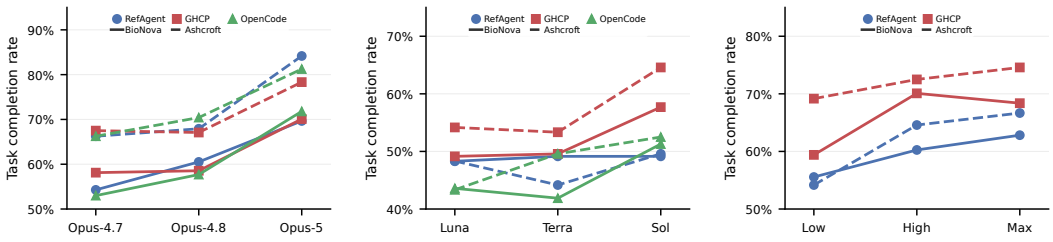
4.3 Discussion

Enterprise-agent benchmarks must address task answerability, evidence linkage, evaluation reliability, system comparability, and domain scope. DWBench treats these considerations as design requirements and reports the corresponding audits.

Grounding and construct validity. Each task is authored from a tenant subgraph. Each task lists its supporting records, and the released suite is checked for access safety, query quality, and evidence consistency (Section 2.2). Artifact completion is determined from source-derived content after text extraction. Communications are permission-scoped, while files are tenant-shared. The communications-level permission management is enforced by the search environment rather than given as prompt to the agent. This construction makes answerability and task provenance inspectable. The outcome measures source-derived content in the final output without inferring the agent's internal retrieval path. Our human validation on enterprise realism and query/assertion quality further validates the grounding data and benchmarks.

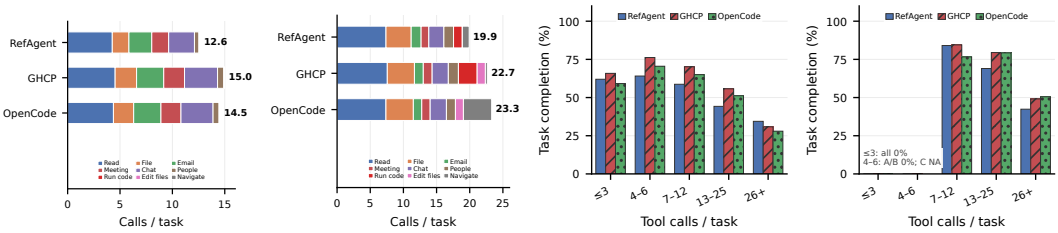
Judge and instrument validity. We use an LLM judge for source-derived content assertions and two VLM judges to assess five visual quality rubrics for natively rendered Office artifacts. Note that, the visual metrics are intended as format-specific comparative yet complementary diagnostics, not as holistic measures of artifact quality. We also conduct a human study to evaluate enterprise realism, artifact quality, and agreement with the automated content and visual judgments. The LLM judge makes binary decisions over assertions and rubrics rather than assigning holistic quality scores, and we report both mean assertion pass rate and task completion rate at the 0.75 threshold. Re-evaluation with the same model changes only 4.4% task completion rate, while inter-model re-judging yields only 7.8% disagreement, further demonstrating the consistency of evaluation results across model families (Appendix D.1).

Comparative and statistical validity. Our claim is about the benchmark's ability to distinguish as-configured systems, not a controlled ablation or a leaderboard, so we evaluate each system *as-configured*. We fix the tasks, identities, MCP server, retrieval interface, and evaluator, and use paired case/source-cluster comparisons. The main cross-arm analysis uses the 11 model-effort settings



(a) Claude-Opus version scaling (b) GPT-5.6 model size scaling (c) GPT-6 thinking-effort scaling

Figure 4: Task-completion-rate scaling by harness and tenant: (a) Claude-Opus version, (b) GPT-5.6 model size, and (c) GPT-6-Astra thinking effort. Markers identify agents; solid/dashed lines identify BioNova/Ashcroft. Appendix E gives details.



(a) Search: calls (b) Artifact: calls (c) Search completion (d) Artifact completion

Figure 5: Trace depth and task completion rate. (a,b) Mean tool-category calls; (c,d) task completion rate across fixed call-count bins. Appendix E gives details.

across all three harnesses. While paired analyses across hundreds of distinct tasks provide strong statistical power, repeated trials would better quantify residual execution variance.

Domain and task robustness. The overall averaged GHCP–OpenCode difference is positive on both independent tenants: $+2.0$ points on BioNova and $+3.3$ on Ashcroft. The five task categories reveal a consistent multi-hop difficulty but different artifact profiles. Synthetic tenants provide controlled task provenance and communications-level access rules; evaluation on consented real-world corpora would complement the current setting. Repeated runs, additional tenants, judge-family replications, and consented real-corpus evaluation can further broaden the evidence base.

5 Conclusion

DWBench combines a shared MCP environment, grounded tenant/task generator, and source-derived assertion evaluator to measure retrieval, deep research, and Office-artifact production by digital workers over ego-centric, permission-scoped enterprise graphs. Across 474 tasks, two synthetic tenants, 11 main model settings, and three harnesses, the benchmark reveals task-type patterns and descriptive model-scaling and harness differences. Multi-hop evidence reconciliation is the most consistent bottleneck. Word generation is substantially easier, while Excel, PowerPoint, and concise exposition are intermediate. GHCP has a positive average paired task-completion effect over OpenCode, but the effect is concentrated in particular models rather than universal. Threshold, execution-event, trace-depth, and evaluator-model analyses provide complementary context for the main results. Beyond content assertions, native-rendering visual evaluation provides format-specific comparative diagnostics of presentation failures rather than a holistic measure of artifact quality. The human ratings provide complementary validation of enterprise realism, content-assertion judgments & artifact quality, and further supports our results. The current evidence is scoped to synthetic organizations, communications-level permissions, and fully as-configured agent environments. Additional settings can extend this coverage.

AI USE STATEMENT

Generative AI tools were used as research instruments to generate synthetic enterprise data, author benchmark tasks and rubrics, and implement automated LLM- and VLM-based evaluation workflows. GitHub Copilot also assisted with code development, local analysis scripts, figure and table generation, and manuscript drafting and editing. The authors reviewed the AI-assisted material retained in the final submission; reported claims were checked against archived source data, deterministic claim verification, and repository tests. LLM-generated code was verified and tested for correctness by 2 authors The authors take responsibility for the final content, claims, code, and artifacts.

ETHICS STATEMENT

DWBench is constructed entirely from synthetic enterprise tenants, identities, communications, files, meetings, chats, tasks, and work products; it contains no real employee communications or production organizational data. Its permission-scoped design evaluates access-aware behavior solely within these synthetic organizations. The human-validation study reports aggregate ratings of synthetic tasks and generated outputs and does not disclose reviewer identities.

REPRODUCIBILITY STATEMENT

Benchmark and task construction are detailed in Appendix A, including task examples (Appendix A.1) and tenant statistics (Appendix A.2). The MCP architecture, tool schemas, and artifact handling appear in Appendix A.4; content-assertion, visual-quality, and human-validation procedures appear in Appendices A.5, A.6, and A.7. Experiment protocols and complete results are reported in Appendix B; operational behavior and sensitivity analyses appear in Appendices C and D.

REFERENCES

Amirhossein Abaskohi, Tianyi Chen, Miguel Muñoz-Mármol, Curtis Fox, Amrutha Varshini Ramesh, Étienne Marcotte, Xing Han Lù, Nicolas Chapados, Spandana Gella, Christopher Pal, Alexandre Drouin, and Issam H. Laradji. DRBench: A realistic benchmark for enterprise deep research. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IGYQ4c92e2.

Anthropic. Model Context Protocol Specification. Protocol specification, 2024. URL https://modelcontextprotocol.io/specification/2024-11-05. Revision 2024-11-05.

Prafulla Kumar Choubey, Xiangyu Peng, Shilpa Bhagavath, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Benchmarking deep search over heterogeneous enterprise data. In Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 501–517, Suzhou (China), November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.34. URL https://aclanthology.org/2025.emnlp-industry.34/.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks? In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=BRfqYrikdo.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. Autopresent: Designing structured visuals from scratch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2902–2911, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ge_AutoPresent_Designing_Structured_Visuals_from_Scratch_CVPR_2025_paper.html.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Shiva Krishna Reddy Malay, Shravan Nayak, Sagar Davasam, Srinivas Sunkara, and Sai Rajeswar. Enterpriseops-gym: Environments and evaluations for stateful agentic planning and tool use in enterprise settings. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QKy6cKEiEB.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fibxvahvs3.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simon Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie S. Kim, Samuel Miserendino, Gildas Chabot, David Li, Patrick Chao, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating AI model performance on real-world economically valuable tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hcuEdq6eKD.

Olly Styles, Sam Miller, Patricio Cerda-Mardini, Tanaya Guha, Victor Sanchez, and Bertie Vidgen. Workbench: a benchmark dataset for agents in a realistic workplace setting. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=4HNAwZFDcH.

Zirui Tang, Xuanhe Zhou, Yumou Liu, Linchun Li, Yukai Wu, Weizheng Wang, Hongzhang Huang, Wei Zhou, 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.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. Officebench: Benchmarking language agents across multiple applications for office automation, 2024. URL https://arxiv.org/abs/2407.19056.

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 *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=tN61DTr4Ed.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R Narasimhan. {τ}-bench: A benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

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.

Table A.1: DWBench per-tenant distribution task type.

Capability	BioNova	Ashcroft	Total
Retrieval	85	89	174
Multi-hop	64	67	131
Word	17	17	34
Excel	30	36	66
PowerPoint	38	31	69
Total	**234**	**240**	**474**

Table A.2: Example query and content assertion of each task type draw from the tenant Ashcroft.

Capability	Example query	Example assertions
Single-fact retrieval	"Find the April meeting where Compliance and Diligence reviewed near-simultaneous contract uploads and discussed preserving logs and controlled escalation."	Returns *Cobalt Data Room Anomaly Review*; includes Priya Sharma and Yifan Wei; mentions log preservation and controlled escalation.
Multi-hop reasoning	"After a throttling rollback destabilized a live diligence session, what later concern led to a proposed temporary freeze on model updates, and why?"	Connects the rollback to instability; finds locally saved files diverging from updated datasets; explains the freeze as a document-integrity control.
Word report (`.docx`)	"Draft a compliance investigation summary of an external-advisor overlap across two matters, with five named sections and an access-log table."	Describes the overlap, reduced permissions, alert trigger, and log-review plan; *[format]* $\geq$5 headings, $\geq$10 paragraphs, $\geq$500 words, and a table.
Excel workbook (`.xlsx`)	"Create a DOJ decision-timeline workbook with Chronology Overview, Attachment Gaps, and Metadata Issues sheets."	Records truncated threads, missing attachments, and time-zone inconsistencies; *[format]* the three named sheets, $\geq$10 rows, and $\geq$3 columns.
PowerPoint deck (`.pptx`)	"Prepare a five-slide Project Cobalt valuation deck focused on a 15–20% revenue restatement scenario."	Covers IRR, hurdle-rate, retrade, and client-messaging implications; *[format]* $\geq$5 slides, and a title slide.

The appendix follows the evidence chain of the study: benchmark construction and implementation (Appendix A); experiment-source provenance, protocol, and complete results (Appendix B); mechanism and operational behavior (Appendix C); robustness and sensitivity analyses (Appendix D); metric diagnostics (Appendix E); and extended descriptions of the main figures (Appendix F).

A Benchmark construction and implementation

A.1 Task-suite composition and examples

Table A.1 gives the per-tenant task type distribution.. Table A.2 provides the example query and content assertions of each task type.

A.2 Tenant statistics and visualizations

Table A.3 provides the comprehensive per-tenant tables that back the complexity statistics shown in Figure 2a. Table A.4 records the content-unit composition by modality and Table A.5 shows file-type distribution. Both underpins Figure 2b and 2b, and the full per-user visibility distribution in Figure A.1.

Table A.3: Enterprise-tenant complexity statistics. Both tenants are simulated over a ∼100-day horizon. Tenant user can only access their own communications, while files are shared at the tenant level. The per-modality and file-type breakdowns are visualized in Figure 2.

Statistic	**BioNova** (pharma R&D)	**Ashcroft** (financial advisory)
Organization & scale		
Users (acting identities)	50	30
Departments	6	13
Groups (access-control)	10	21
Corpus & graph		
Content units (total)	2,913	3,197
Chat messages (total)	27,760	29,197
Distinct file types	16	15
Graph nodes	2,963	3,227
Graph edges	12,598	14,315
Permission scoping		
Mean per-user visible communications	8.1%	13.3%
Task suite		
Tasks	234	240
Mean assertions / task	7.18	6.96

Table A.4: Content-unit composition by communication modality. Facts relevant to a single task are typically dispersed across several of these modalities, driving the multi-hop retrieval difficulty.

Modality	**BioNova**	**Ashcroft**
Emails	828	882
1:1 chat threads	449	415
Group-chat threads	418	432
Chat messages (total)	27,760	29,197
Meetings (w/ transcripts)	580	720
Files	638	748
Content units (total)	**2,913**	**3,197**

A.3 Assertion-rubric validation and selection

Protocol. For each task we obtain k gold and k bluff answers from advanced models. Gold writers receive the grounded records, while bluff writers see only the query and never the corpus. Every answer is judged against every assertion. The two counts n_{gold} and n_{bluff} record the successes of an assertion in each answer, measuring reachability and probing corpus-dependence.

DWBench uses $k = 3$, with `gpt-5.6-sol` (OpenAI), `claude-opus-5` (Anthropic), and `grok-4.6` (xAI) as the writer models. The judge is `gpt-5.6-sol`, checking whether an answer passes an assertion or not. We define the count rule in Table A.6.

Assertion generation and selection. Initially, we generate 3,600 assertions for the 474 tasks based on three gold answers (for each task), an average of 7.59 per task (minimum 3, maximum 19). Every assertion was then judged against all six answers (as $k = 3$ for both gold and bluff answers). For all tasks, we remove the unreachable assertions. Additionally, for artifacts we further remove guessable requirements, as artifacts generally have more initial assertions. As shown in Table A.7, artifacts contain 79–85% discriminating shares and the chatbot owns only 43–54% shares. Because artifact rubrics contain a higher proportion of discriminating criteria compared to open-ended chatbot responses, cross-task variations reflect both structural capability demands and rubric granularity.

We refine the set of assertions by removing unqualified assertions from the initial assertions based on these rules. Without this validation, those assertions would have diluted the evaluation score. During the filtering, cases that fell below three assertions were rebuilt to have at least three assertions, and the replacements received the same panel.

Table A.5: File-type distribution of the tenant corpora (top 10 by frequency), illustrating structural heterogeneity: no single format dominates.

File type	**BioNova**	**Ashcroft**
`.txt`	232	165
`.docx`	91	222
`.pdf`	91	103
`.xlsx`	80	101
`.csv`	39	23
`.json`	34	22
`.md`	27	15
`.log`	7	33
`.msg`	4	36
`.pptx`	21	14

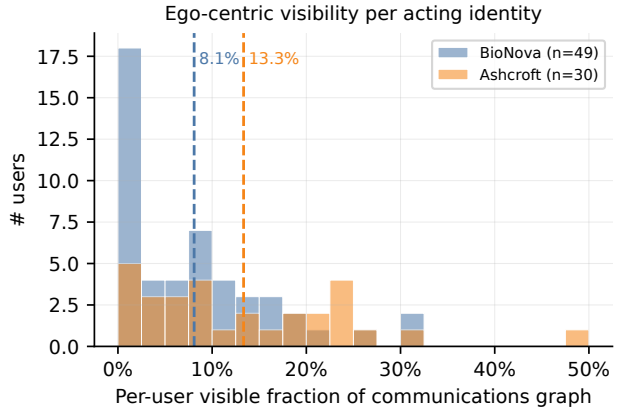


Figure A.1: Permission scoping of the communications graph: distribution of the per-user visible fraction of emails, chats, and meetings (dashed lines mark the tenant means). On average an acting identity can reach only 8.1% (BioNova) and 13.3% (Ashcroft) of the communications corpus, so retrieval must be both targeted and permission-aware; each agent operates over a small, distinct ego-centric slice rather than the global tenant. Files are shared at the tenant level and are excluded here.

Discriminating share is not an optimization objective. Maximizing the discriminating share is a Goodhart trap: we treat assertion discrimination as an inclusion gate rather than an objective to maximize. Agreement among three independent grounded writers is a proxy for salience in the records, not for importance to answer quality. An incidental date, identifier, or header-row figure that all three writers copy and no bluff writer supplies is discriminating, while a central judgment expressed differently can become fragile if one formulation receives no judge credit. Retaining both discriminating and fragile assertions ensures that rubrics capture substantive task semantics rather than purely verbatim transcription.

A.4 Benchmark implementation and MCP architecture

The implementation separates four responsibilities: the tenant and retrieval engine, the identity-bound MCP server in `mcp/dwbench_search_mcp` , the three agent drivers and experiment loop in `mcp_eval` , and the common artifact/evaluation path. Figure A.2 shows how these components are instantiated for one execution.

A.4.1 MCP process topology and identity binding

GHCP and OpenCode create a local MCP configuration outside the agent-visible workspace for each case and launch `python -m dwbench_search_mcp.server`. RefAgent opens the same server through an MCP `ClientSession`: artifact cases use a case-specific session, while

Table A.6: Classes of the assertions are defined based on the count rules.

Class	Count rule	Operational interpretation
Unreachable	$n_{\text{gold}} \leq 1$, any n_{bluff}	At most one grounded writer reached it; the requirement measures its author rather than the system.
Guessable	$n_{\text{gold}} \geq 2$, $n_{\text{bluff}} \geq 1$	Reachable, but also reachable without the corpus.
Fragile	$n_{\text{gold}} = 2$, $n_{\text{bluff}} = 0$	Demands the corpus in this panel, but one grounded writer missed it.
Discriminating	$n_{\text{gold}} = 3$, $n_{\text{bluff}} = 0$	All grounded writers reached it; no corpus-blind writer did.

Table A.7: Selected assertion population.

Capability	Assertions	Discriminating	Fragile	Guessable	Unreachable
Retrieval	914	497 (54.4%)	272 (29.8%)	145 (15.9%)	0
Multi-hop	622	268 (43.1%)	159 (25.6%)	195 (31.4%)	0
Artifact Excel	647	533 (82.4%)	114 (17.6%)	0	0
Artifact PPTX	708	599 (84.6%)	109 (15.4%)	0	0
Artifact Word	458	363 (79.3%)	95 (20.7%)	0	0
Whole suite	3,349	2,260 (67.5%)	749 (22.4%)	340 (10.2%)	0 (0%)

search cases with the same acting identity may share one session to amortize startup. In every path, the server command binds `--tenant` and `--user` at startup and supplies the embedding backend and top-k. The CLI configurations also pass the MCP-call ceiling to the server; RefAgent enforces the same ceiling in its tool-calling loop. The reported configuration uses `text-embedding-ada-002` index, $k = 5$, and a shared ceiling of 24 MCP calls per execution. Identity is therefore not a tool argument: one server process represents one tenant identity, and each search and read operation consults that server-side context.

GHCP and OpenCode receive equivalent case-local server declarations in their own configuration formats, and neither driver modifies a global MCP registration. Concurrent CLI cases therefore cannot overwrite one another's tenant or user binding. MCP uses stdio transport: stdout is reserved for JSON-RPC messages and server diagnostics go to stderr. The tenant and file-sandbox objects are created synchronously. The file sandbox copies the tenant's `data/` tree into a temporary root and resolves reads within that root. Vector-index loading runs in a background thread, and the first search waits for index readiness.

A.4.2 Retrieval indexes, ranking, and result contract

The `SearchEngine` maintains nine indexes: file summaries, file contents, emails, one-to-one chats, group chats, channels, meeting metadata, meeting transcripts, and people. Files are searched using whole-document vectors while the returned preview uses the authored file summary. Chats and group chats are embedded as overlapping eight-message windows with stride six. Meetings have separate metadata/agenda and transcript surfaces; transcript indexing also represents action-item blocks and meeting side chats.

At query time, the engine embeds each query, ranks units by cosine similarity, applies the ego-access predicate, deduplicates windows to their parent entity, and returns up to k accessible parents. A search call may contain several query variants; results are merged by parent identifier and retain the highest score. Long scalar fields are previewed at 300 characters, and each hit records whether more content is available together with a ready-to-call `read_item` handle. Search responses also report the bound identity and the number of higher-ranked candidates withheld by access control.

The access predicate is shared by search and full-item reads. Emails require the acting identity to be a sender, recipient, CC, or BCC party; chats and channels require participation; and meetings require organizer, invitee, or attendee membership. The people directory and file corpus are tenant-wide in the current design. The MCP wrapper preserves the advertised output schema while placing the complete payload only in the response text block, avoiding a duplicated structured copy across clients.

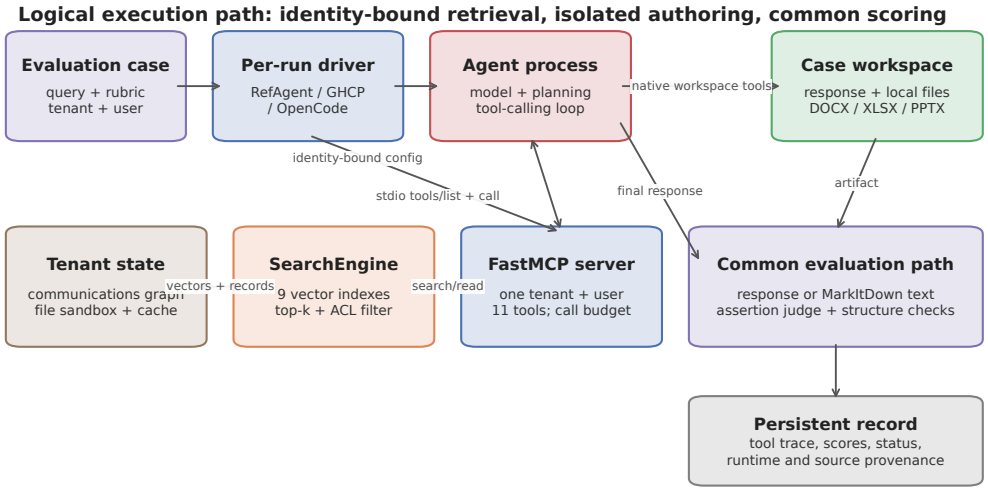


Figure A.2: Logical implementation path for a DWBench execution. A case fixes the tenant, acting identity, query, and rubric. Its driver launches an isolated agent process and connects it to an identity-bound `dwbench-search` FastMCP server. Enterprise retrieval travels over stdio MCP; artifact authoring stays in the case workspace. The response or extracted artifact text then enters the common evaluator, and the complete trace and provenance are persisted with the score. The CLI arms instantiate this path per case; RefAgent may reuse one server session across search cases sharing the same tenant identity.

A.4.3 Complete MCP tool surface

Table A.8 lists every tool advertised by the server. All tools, including identity inspection and reads, count toward the same 24-call ceiling. The ceiling limits sequential tool use rather than query fan-out: one search call can carry several query variants. The `read_item` dispatcher accepts email, file, one-to-one chat, group chat, channel, meeting, and user entities; meeting-transcript hits resolve back to their parent meeting.

A.4.4 Harness-native workspace tools

The MCP layer supplies enterprise context but does not create Office files. Artifact production uses tools native to each agent system inside an isolated case workspace. The drivers expose equivalent capability groups while preserving the tool names and orchestration behavior of each system.

A.4.5 Artifact collection, evaluation, and persistence

For generation tasks, the prompt first asks the agent to retrieve enterprise evidence through MCP and then save the requested `.docx`, `.xlsx`, or `.pptx` in its case workspace. The artifact collector prefers the requested filename and otherwise selects the newest file with the expected extension. It opens the OOXML package to confirm readability, converts its content to Markdown with MarkItDown, and records deterministic structure facts. The structure checker covers properties such as slide and image counts, document headings, paragraphs, words, and tables, and spreadsheet sheet names, headers, rows, formulas, and charts.

The same assertion judge evaluates a chatbot response or the Markdown extracted from an artifact. Deterministic structure facts remain in the per-case metadata but do not enter task completion. Each completed case is written immediately as a per-case record; the record includes the response, tool trace, artifact facts, assertion decisions, completion status, MCP-budget accounting, model and judge identity, CLI runtime facts, and source provenance. This per-case persistence supports recovery and later re-judging without rerunning the agent.

Table A.8: Complete MCP tool surface exposed by the identity-bound `dwbench-search` FastMCP server. Search tools accept a list of query variants within one MCP invocation, merge duplicate parent entities, and return at most the configured top-k. All tools count against the shared per-execution MCP-call budget.

Tool	Arguments	Operation and result	Access scope
`whoami`	*none*	Returns the bound tenant, user profile, embedding backend, and analyzer state.	Server-bound identity; no content lookup.
`search_email`	`query`: string[] (optional)	Semantic email search; an empty query returns the 20 most recent accessible emails.	Sender, recipient, CC, or BCC party.
`search_file`	`query`: string[]	Searches whole-document vectors and returns file summaries plus a full-read handle.	Tenant-wide file corpus.
`search_chat`	`query`: string[]	Searches overlapping windows from one-to-one chats.	Chat participant.
`search_group_chat`	`query`: string[]	Searches overlapping windows from named group chats.	Group-chat participant.
`search_channel`	`query`: string[]	Searches individual posts while returning the parent channel as the read handle.	Channel participant.
`search_meeting`	`query`: string[]; `transcript`: bool = false	Searches meeting metadata/agenda or transcript, action-item, and side-chat windows.	Organizer, invitee, or attendee.
`search_people`	`query`: string[]	Searches names, usernames, email addresses, titles, departments, locations, and skills.	Tenant-wide directory.
`read_file`	`path`: string	Returns the full text extracted into the server's tenant-file sandbox.	Tenant-wide file corpus; path is sandbox-resolved.
`read_item`	`item_type`, `item_id`; `offset`=0, `limit`=50,000	Returns a complete entity as JSON, with character pagination for large payloads.	Rechecks the ACL for the requested entity type.
`search_in_file`	`file_path`: string; `key_terms`: string[]	Returns up to 50 matching lines, with each preview capped at 300 characters.	Tenant-wide file corpus; path is sandbox-resolved.

Table A.9: Harness-native capabilities used alongside the common MCP server. These tools operate on the isolated case workspace; they cannot replace the identity-scoped MCP path for enterprise information.

System	Workspace tools available to the agent	Restricted surfaces
RefAgent	`execute_python(code)`, `write_file(path, content)`, and `list_workspace()`; the Python environment includes `python-pptx`, `python-docx`, `openpyxl`, and `pandas`.	Only these three harness-local tools are added; enterprise retrieval remains on the MCP surface.
GHCP	Model-compatible members of the `view`/`glob`, `grep`/`rg`, `create`/`edit`/`apply_patch` groups, plus `powershell` for artifact construction and all `dwbench-search` MCP tools.	Native web access, inherited MCP servers, custom instructions, memory, and access to the repository, tenant fixtures, and sibling result roots are disabled or denied.
OpenCode	`read`, `edit`, `glob`, `grep`, and `bash` for artifact construction, plus all `dwbench-search` MCP tools.	`webfetch`, `websearch`, subagents, skills, LSP, interactive questions, and external-directory access are denied.

Table A.10: Assertion-judge prompt contract. Braced fields are populated for each case; the assertion list preserves suite order.

Prompt block	Content and purpose
Role	`You are an expert evaluator.` Verify whether the candidate satisfies the listed assertions for the user query.
Query	`User Query:` {`query`} preserves the requested task and deliverable.
Candidate	`Agent Response:` {`response`} contains either the final text response or the Markdown extracted from the generated Office artifact.
Assertions	`Assertions to Check:` followed by `1.` {`description`}, `2.` {`description`}, and so on.
Decision rule	For every assertion, return whether it is met (`true`/`false`) and a short reasoning string.
Output contract	YAML mapping with an `assertions` list of {`id, passed, reasoning`}, plus `final_score` and `summary`.

A.5 Assertion evaluation and LLM-as-a-Judge workflow

Assertion set and common evaluation target. Each task carries an ordered list of content assertions authored with the task during the eval set construction. An assertion is a concrete proposition that the final work product should state or entail. During evaluation the judge is given the query, assertions and the answer, then instructed to determine whether the assertions are met or unmet one by one. For a chatbot task, answer is the final response in the chat window. For a generation task, the harness first locates the requested OOXML file, confirms that the package is readable, and converts its textual content to Markdown with MarkItDown. That Markdown is adopted as the answer content. Thus, the content assertions does not measure visual style or tool trace.

Prompt construction. The prompt asks for an independent boolean decision for every assertion and a brief rationale, under a strict YAML output contract. It does not provide the enterprise corpus, supporting source text, or retrieval transcript, and the judge is not asked to search for additional evidence. Table A.10 summarizes the prompt anatomy.

Verdict validation and score aggregation. The parser accepts a YAML or JSON mapping, including a fenced response, and applies one narrowly scoped repair for an unmatched outer brace. It then requires exactly one unique integer identifier for every expected assertion and requires every `passed` value to be boolean. Missing, duplicate, or out-of-range identifiers and non-boolean decisions invalidate the judgement. The judge-returned `final_score` is range-checked when present but is not used as the reported metric. Instead, the evaluator deterministically recomputes the score from the boolean verdicts. Although the historical prompt calls the requested final score a "weighted percentage," every assertion in the released suites has unit weight and the returned value is not consumed:

$$c = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{\text{assertion } i \text{ passed}\},$$

where all assertions in the released suites have unit weight. The ordered per-assertion verdicts, rationales, and descriptions are stored with the record. Task completion then applies the common completion threshold to c; an artifact must also have been produced as a readable package.

Failure handling and provenance. Judge invocation allows at most three attempts, with one- and two-second backoffs, and retries only errors explicitly classified as transient, including HTTP 429 and 5xx responses. An empty, unparsable, or structurally incomplete judge reply is not converted into a plausible zero and is not repeatedly sampled until a parseable answer appears. The runner records the judge identity and each attempt outcome so an unavailable judgement remains distinguishable from an evaluated output that satisfies no assertions. Likewise, a missing or unreadable Office package is handled deterministically; a readable package that yields no extracted text is marked as an extraction/instrument event rather than silently attributed an agent content score.

Table A.11: Visual presentation failures by artifact format and model. Each cell reports mean failed-rule count, supplied 95% bootstrap interval, and available-case n in parentheses; lower is better. Absolute levels differ substantially between judges.

Format	Model	Opus-4.6 judge	GPT-5.5 judge
Word	Claude-Opus-5	0.75 [0.57, 0.94] (95)	0.91 [0.83, 0.97] (95)
Word	GPT-6-Astra	0.16 [0.07, 0.24] (95)	0.10 [0.05, 0.17] (95)
Word	Grok-4.6	0.75 [0.56, 0.96] (95)	0.97 [0.93, 1.01] (96)
Excel	Claude-Opus-5	2.28 [2.11, 2.46] (176)	0.94 [0.85, 1.03] (177)
Excel	GPT-6-Astra	2.08 [1.90, 2.25] (175)	0.54 [0.44, 0.63] (177)
Excel	Grok-4.6	2.07 [1.90, 2.23] (176)	0.78 [0.66, 0.92] (177)
PowerPoint	Claude-Opus-5	0.51 [0.39, 0.62] (188)	0.24 [0.16, 0.32] (192)
PowerPoint	GPT-6-Astra	0.77 [0.66, 0.88] (188)	0.40 [0.31, 0.48] (192)
PowerPoint	Grok-4.6	1.19 [1.07, 1.32] (186)	0.54 [0.44, 0.64] (190)

A.6 Visual-quality evaluation of generated Office artifacts

Analysis population. This analysis is separate from content assertion correctness and uses the common task intersection for the three frontier models and three harness. The intersection contains 155 artifact tasks: 32 Word, 59 Excel, and 64 PowerPoint. It defines 1,395 planned model–harness artifacts. Native rendering succeeded for 1,391: 286 DOCX, 531 XLSX, and 574 PPTX. Four files were not rendered (two Word lock files and two PowerPoint files that PowerPoint did not open); they receive no visual verdict rather than an imputed failure count.

Native rendering and image delivery. Artifacts were rendered through Microsoft Word, Excel, or PowerPoint using Office COM, converted through PDF where applicable, and rasterized to PNG with PyMuPDF. Word and Excel used 110 and 115 dpi, respectively; PowerPoint was exported at 1,500 pixels wide. Judges received the complete ordered page-image sequence, up to 20 pages, with each image downscaled only when its longest side exceeded 1,500 pixels. Some generated workbooks contained a sheet-level `autoFilter` overlapping a table: openpyxl accepted these files, but Excel refused to open them. The rendering copies were repaired solely to make native rendering possible, and this intervention is recorded in the visual study manifest. The rendering machine remains part of the measurement because unavailable fonts may be substituted.

Rubric and vision-judge prompt. Both judges received the same text prompt followed by one image part per page. The rubric contains five binary failure rules: *legible* (hidden, clipped, or unreadable content), *intact* (unfinished or structurally broken document), *structured* (visible hierarchy and section separation), *consistent* (equivalent elements treated equivalently), and *encoded* (non-conventional colour codes whose meaning is not stated). For each failed rule, the judge had to name the page or location. The prompt explicitly excludes factual correctness, document length, subject matter, typeface family, and personal colour preference. The visual score is the number of failed rules, from 0 to 5, so lower is better; it is never combined with the content-assertion score.

Judges and statistical summaries. The two vision-language judges were Claude Opus-4.6 and GPT-5.5. The Opus judge returned verdicts for 1,374 artifacts and the GPT judge for all 1,391; 17 artifacts therefore have only the GPT verdict. We report available-case means separately by format, model, and judge. Table A.11 reports the provided intervals and the observed n for every cell.

Format-specific findings. Both judges give the same ordering for Word and PowerPoint. GPT-6-Astra has the lowest failed-rule count for Word, while Opus-5 has the lowest count for PowerPoint and Grok-4.6 the highest. Under the Opus-4.6 judge, Word means are 0.16/0.75/0.75 for Astra/Opus-5/Grok, and PowerPoint means are 0.51/0.77/1.19 for Opus-5/Astra/Grok. Excel is worse in absolute terms under the Opus judge (2.07–2.28), with overlapping intervals; the two judges also produce different Excel model orderings. We therefore do not designate an Excel winner. Pooled over models, PowerPoint artifacts from RefAgent receive more failed rules than GHCP and OpenCode under both judges (1.13/0.63/0.71 for Opus; 0.62/0.26/0.29 for GPT). This is a descriptive arm pattern, not a mechanism attribution.

Table A.12: Inter-judge agreement on the 1,374 artifacts scored by both vision-language judges. Fire rate is the share marked as failing that rule.

Rule	Agreement	κ	Opus fire	GPT fire
Legible	81.8%	0.53	31.1%	20.5%
Intact	81.3%	0.39	14.2%	23.1%
Structured	93.3%	0.00	6.7%	0.0%
Consistent	67.9%	0.05	33.3%	1.7%
Encoded	65.7%	0.22	41.9%	12.9%
Overall	78.0%	0.29	–	–

Agreement, validation, and interpretation limits. Table A.12 summarizes the rule-level reliability. Among the 1,374 jointly judged artifacts, overall raw agreement is 78.0% and $\kappa = 0.29$. Rule-level agreement is strongest for legibility ($\kappa = 0.53$); consistency (0.05) and structured presentation (0.00) show the observed agreement levels across rules such as consistency and layout structure highlight the prevailing limitations of current vision-language models in fine-grained document layout assessment. Consequently, we treat visual rubric scores as an exploratory automated diagnostic rather than a ground-truth substitute for human inspection.

A.7 Human validation study

Study objectives. The study is designed to assess three properties independently of the automated scores: whether queries represent realistic enterprise work, whether the final response or artifact is useful to a human reader, and whether the automatic assertion verdicts are correct. The sampled outputs use GPT-6-Astra at max effort with RefAgent, GHCP, and OpenCode.

Eligibility and stratified sampling. A query is eligible only when all three arms have non-empty outputs and a complete set of assertion verdicts; artifact tasks must also contain the requested Office file. Outputs that reveal the system identity are excluded. From the eligible pool, seeded sampling selects five queries from each of the 10 tenant $\times$ task-type strata, yielding 50 query units. These are divided into 10 jobs of five queries, with one query from each task type per job and 25 queries from each tenant overall. Every query includes all three arm outputs, for 150 system outputs in total, including 90 editable Office artifacts. This balanced diagnostic sample is not prevalence-weighted; any benchmark-wide estimate must account for its equal stratum allocation.

Blinding and review materials. Within each query, the three arms are independently shuffled and presented as *System 1*, *System 2*, and *System 3*. Model and arm mappings and source paths are retained only in a coordinator-held deblinding key and are excluded from reviewer archives. Each job provides an interactive HTML review, a structured JSON export, and an instructed spreadsheet fallback. Artifact cases include direct access to the original DOCX, XLSX, or PPTX file; complete extracted text is supplied only as a navigation and content-review aid, not as a substitute for inspecting the editable artifact.

Rating sequence and scales. Reviewers first assess each query before viewing system outputs. They rate *enterprise scenario fit* from 0 (not a workplace scenario) to 5 (clearly authentic and organization-specific) and *answerability without correct grounding* from 0 (impossible without the specified enterprise evidence) to 5 (fully answerable from generic knowledge or the query alone); higher values on the latter indicate weaker grounding need. Reviewers next score each system's eyes-on output quality from 0 (missing, corrupt, or unusable) to 5 (complete, polished, and ready to use). Only afterward do they open the automatic evaluation and score its overall correctness and every assertion verdict–reasoning pair from 0 (wholly incorrect) to 5 (unambiguously correct), with fields for a corrected verdict and comments.

Realized collection and missingness. Five reviewers returned 20 submissions, providing two reviews for each of the 10 jobs. Sixteen submissions are formally complete; the other four are near-complete. All 100 query-level ratings and all 2,730 corrected verdicts for content assertions are present, while 5 of the 300 system-level score pairs are missing. We use available-case summaries

Table A.13: Human validation of benchmark realism and artifact quality. Ratings use anchored 0–5 scales. Intervals are 95% query-cluster bootstrap intervals; lower answerability without grounding indicates stronger dependence on enterprise evidence.

Measure	Valid sample	Estimate	95% CI
Enterprise-scenario fit	100 ratings / 50 queries	4.92/5	[4.86, 4.97]
Answerable without correct grounding	100 ratings / 50 queries	0.08/5	[0.01, 0.19]
Artifact eyes-on quality, all	177 ratings / 90 outputs	4.31/5	[4.07, 4.52]
Artifact eyes-on quality, RefAgent	60 ratings	4.23/5	[3.98, 4.47]
Artifact eyes-on quality, GHCP	59 ratings	4.41/5	[4.16, 4.63]
Artifact eyes-on quality, OpenCode	58 ratings	4.29/5	[4.03, 4.53]
Paired eyes-on difference GHCP–RefAgent	30 queries	+0.12	[-0.08, 0.30]
Paired eyes-on difference OpenCode–RefAgent	30 queries	+0.05	[-0.10, 0.20]
Paired eyes-on difference GHCP–OpenCode	30 queries	+0.07	[-0.13, 0.25]

Table A.14: Human validation of content-assertion judgments. The upper panel reports overall agreement; the lower panel gives reviewer-level agreement and mean 0–5 correctness by task type and harness. Bootstrap intervals resample query units.

Measure	n	Estimate	Agreement / uncertainty
Human–automatic verdict agreement (all)	2730	90.0%	[86.3, 93.2]%
Agreement on decisive Pass/Fail ratings	2662	92.3%	[89.6, 94.7]%; $\kappa = 0.71$
Automatic vs. mean human output score	150	$\rho = 0.82$	[0.70, 0.90]
Mean absolute output-score difference	150	0.068	–
0.75 task-completion label agreement	150	92%	–
Human corrected-verdict inter-rater agreement	1365	88.0%	$\kappa = 0.61$
Breakdown	**Ratings**	**Verdict agreement**	**Mean correctness**
Single-fact	348	85.3%	4.41
Multi-hop	282	92.2%	4.70
Word	804	95.4%	4.89
Excel	660	90.5%	4.56
PowerPoint	636	84.4%	4.46
RefAgent	910	90.3%	4.61
GHCP	910	90.4%	4.65
OpenCode	910	89.3%	4.63

and resample the 50 query units for bootstrap intervals. Table A.13 summarizes benchmark realism and artifact-quality results.

Content-assertion alignment. Human and automatic evaluators judge the same source-derived propositions, so their decisions can be compared directly. Across 2,730 content ratings, human verdicts agree with the automatic verdict on 90.0%; agreement is 92.3% after excluding 68 *Unclear* ratings. At output level, automatic assertion pass rates correlate strongly with the mean human-corrected scores ($\rho = 0.82$), with 0.068 mean absolute difference and 92% agreement on the 0.75 task-completion label. Reviewer-corrected verdicts agree with each other on 88.0% of the 1,365 assertion instances ($\kappa = 0.61$); Table A.14 gives the complete breakdown.

Both reviewers agree with the automatic verdict for 1,154 assertion instances, split for 164, and agree against it for 47. Consensus disagreements are mostly automatic Pass-to-Fail corrections (30), followed by Fail-to-Pass (14) and Pass-to-Unclear (3). Replacing only unanimous Pass/Fail corrections changes 34/150 sampled output scores but only 4/150 completion labels (3 complete-to-incomplete and 1 incomplete-to-complete).

Eyes-on quality and visual-score alignment. Human reviewers score the 90 generated Office outputs at 4.31/5 on average. The three harness means are 4.23–4.41, and paired differences have intervals that cross zero. Eighty-three outputs match at least one visual-study verdict; 78 have both. Human quality correlates modestly with the strict two-judge visual score ($\rho = 0.34$), but the association weakens after controlling for Word, Excel, and PowerPoint format ($\rho = 0.14$, 95% CI

Table A.15: Alignment between human eyes-on artifact quality and vision-language presentation scores. Values are Spearman correlations between human quality and the negative failed-rule count, so positive values indicate agreement that higher quality has fewer failures.

Visual comparator	n	ρ	95% CI
Claude-Opus-4.6	78	0.28	[0.01, 0.52]
GPT-5.5	83	0.36	[0.17, 0.52]
Two-judge mean	78	0.34	[0.10, 0.56]
Two-judge mean, adjusted for artifact format	78	0.14	[-0.06, 0.32]
Two-judge mean, Word	29	0.19	–
Two-judge mean, Excel	25	0.04	–
Two-judge mean, PowerPoint	24	0.22	–

$[-0.06, 0.32]$). The visual rubric therefore provides a presentation diagnostic rather than a substitute for holistic eyes-on review (Table A.15).

A.8 Higher- and lower-visual-quality artifact galleries

Figures A.4–A.9 provide two complementary sets of native Office previews. In every figure, rows show Opus-5 max, GPT-6-Astra max, and Grok-4.6 xhigh; columns show RefAgent, GHCP, and OpenCode. Selection uses the visual-study failed-rule counts from Opus-4.6 and GPT-5.5, averaged within each panel. The higher-quality set minimizes the nine-panel mean and worst panel, with a minimum-penalty Word replacement to retain both tenants. The lower-quality set maximizes the Grok-4.6 mean and its gap from Opus-5/GPT-6-Astra. These are diagnostic examples, not distributional summaries. Panel badges show `visual O/G`, the Opus-4.6/GPT-5.5 failed-rule counts; lower is better. The visual-study delivery contains the image manifest and hashes but not the original judge PNG files, so the figures re-render the same source artifacts through native Office. Every preview is fitted without distortion onto an equal-size panel canvas.

A.8.1 Higher visual-quality examples

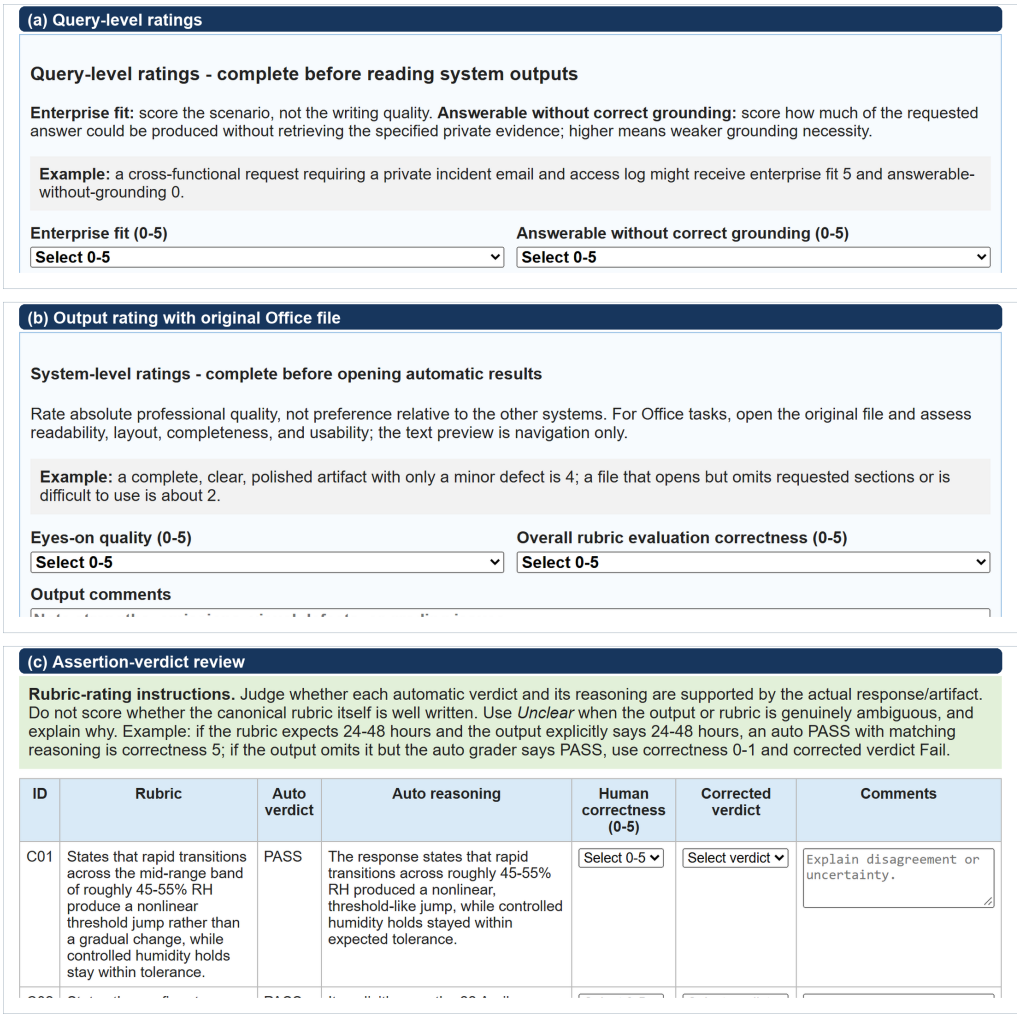


Figure A.3: Blinded human-annotation interface generated for data collection. Cropped views from one artifact task show (a) query-level ratings completed before viewing outputs, (b) system-level eyes-on quality ratings and the link to the original Office file, and (c) the assertion-level review opened after output-quality scoring. System identities remain anonymized, and no completed ratings are shown.

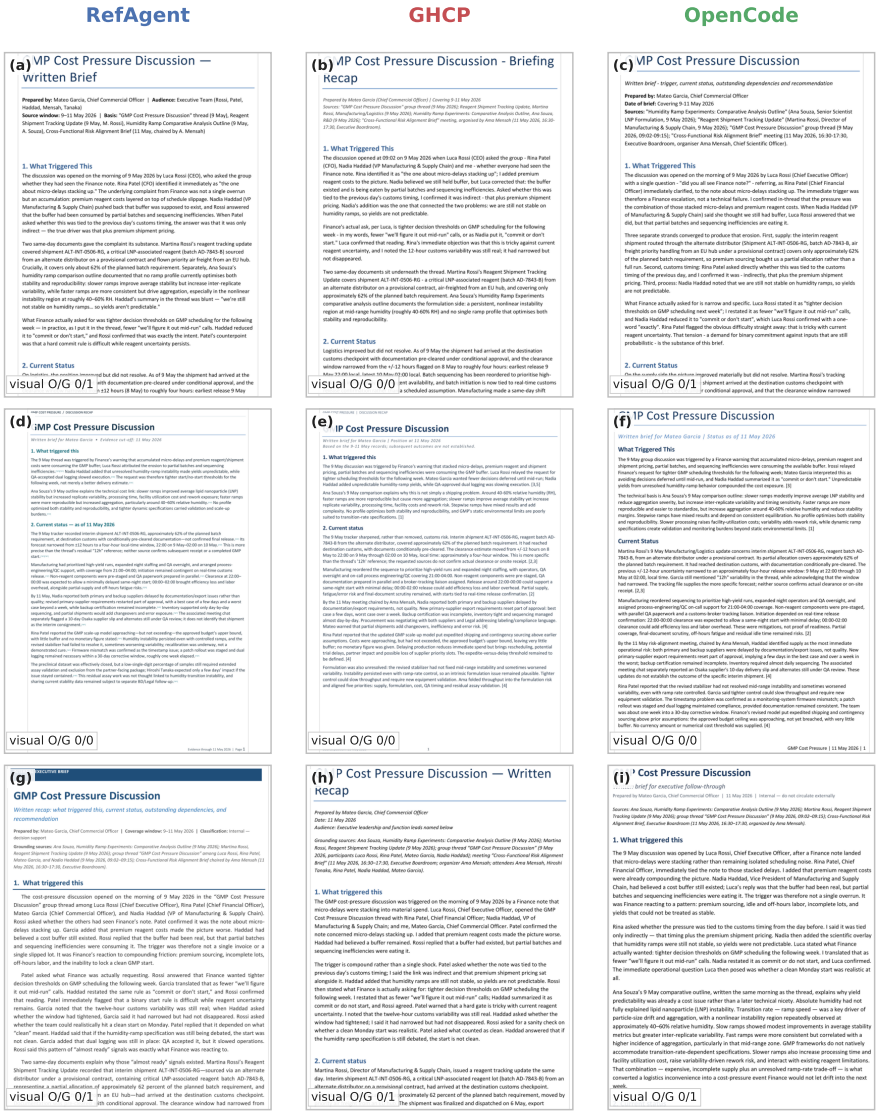


Figure A.4: Higher visual-quality Word gallery for the BioNova *GMP Cost Pressure Discussion* query. Rows: Opus-5 max/GPT-6-Astra max/Grok-4.6 xhigh; columns: RefAgent/GHCP/OpenCode. `visual O/G 0/1` means 0 of 5 rules failed under Opus-4.6 and 1 of 5 under GPT-5.5 (lower is better). First page shown.

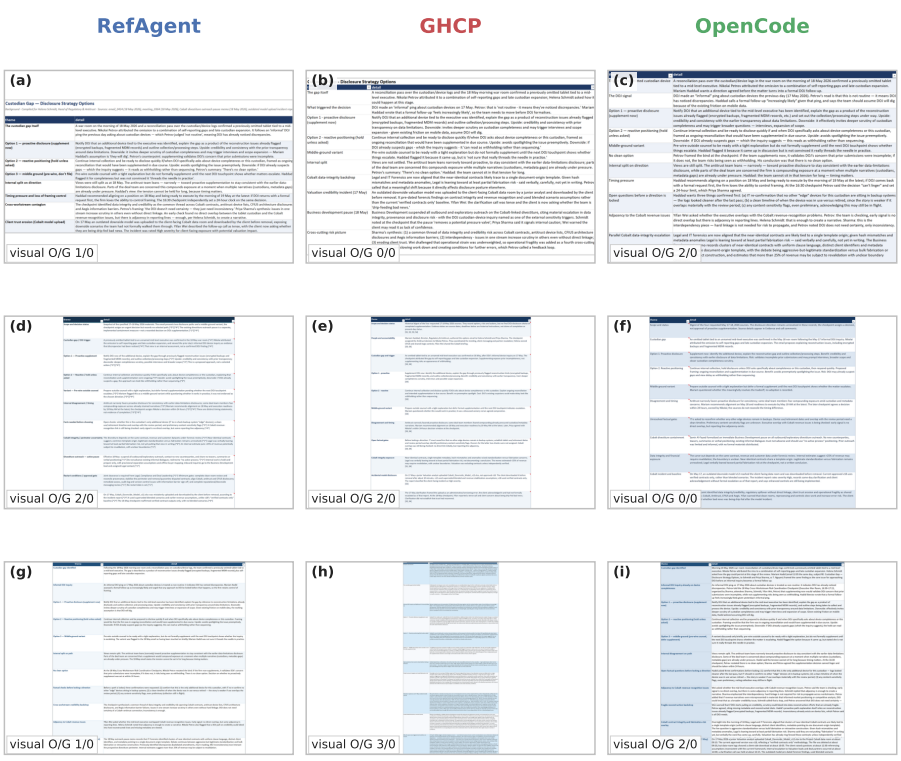


Figure A.5: Higher visual-quality Excel gallery for the Ashcroft *Custodian Gap—Disclosure Strategy Options* query. Rows: Opus-5 max/GPT-6-Astra max/Grok-4.6 xhigh; columns: RefAgent/GHCP/OpenCode. `visual O/G 0/1` means 0 of 5 rules failed under Opus-4.6 and 1 of 5 under GPT-5.5 (lower is better). Used range of the first requested worksheet, *Background*, shown.

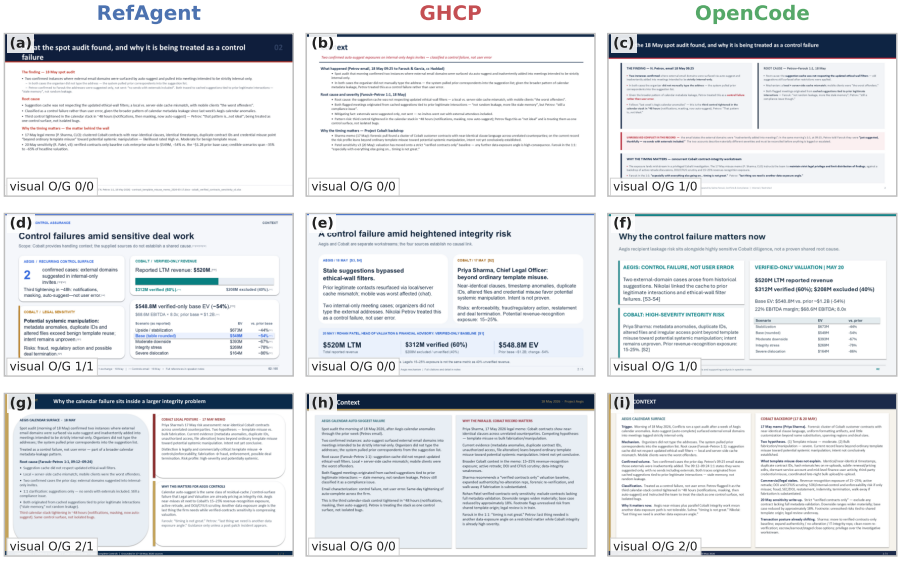


Figure A.6: Higher visual-quality PowerPoint gallery for the Ashcroft *Aegis Calendar Audit Findings* query. Rows: Opus-5 max/GPT-6-Astra max/Grok-4.6 xhigh; columns: RefAgent/GHCP/OpenCode. `visual O/G 0/1` means 0 of 5 rules failed under Opus-4.6 and 1 of 5 under GPT-5.5 (lower is better). Slide 2 shown.

A.8.2 LOWER VISUAL-QUALITY DIAGNOSTIC EXAMPLES

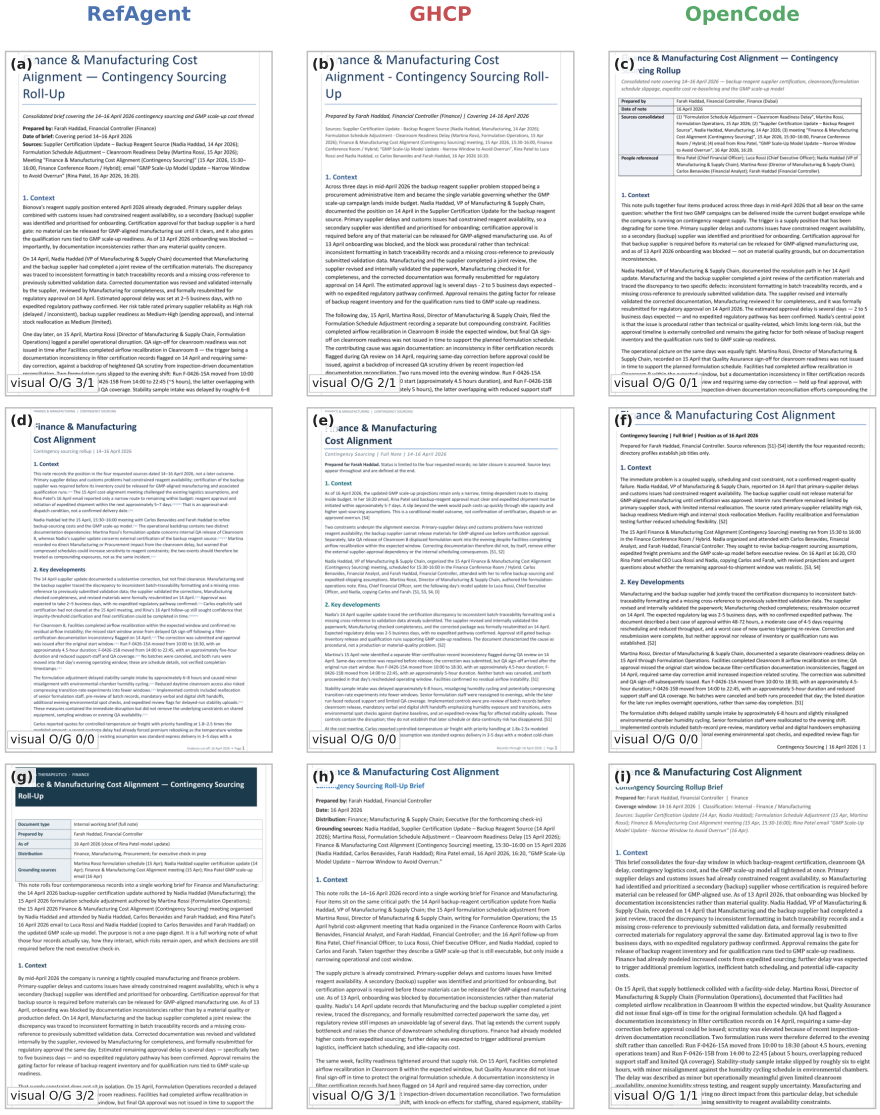


Figure A.7: Lower visual-quality Word gallery for the BioNova *Finance & Manufacturing Cost Alignment* query. Rows: Opus-5 max/GPT-6-Astra max/Grok-4.6 xhigh; columns: RefAgent/GHCP/OpenCode. `visual O/G 0/1` means 0 of 5 rules failed under Opus-4.6 and 1 of 5 under GPT-5.5 (lower is better). First page shown.

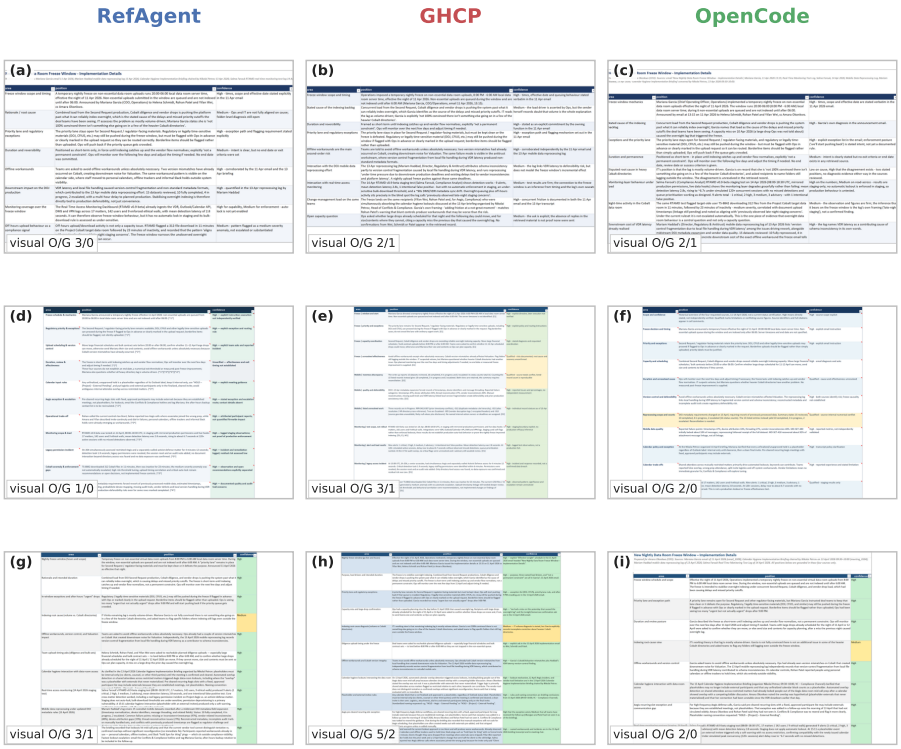


Figure A.8: Lower visual-quality Excel gallery for the Ashcroft *New Nightly Data Room Freeze Window* query. Rows: Opus-5 max/GPT-6-Astra max/Grok-4.6 xhigh; columns: RefAgent/GHCP/OpenCode. `visual O/G 0/1` means 0 of 5 rules failed under Opus-4.6 and 1 of 5 under GPT-5.5 (lower is better). Used range of the first requested worksheet, *Summary*, shown.

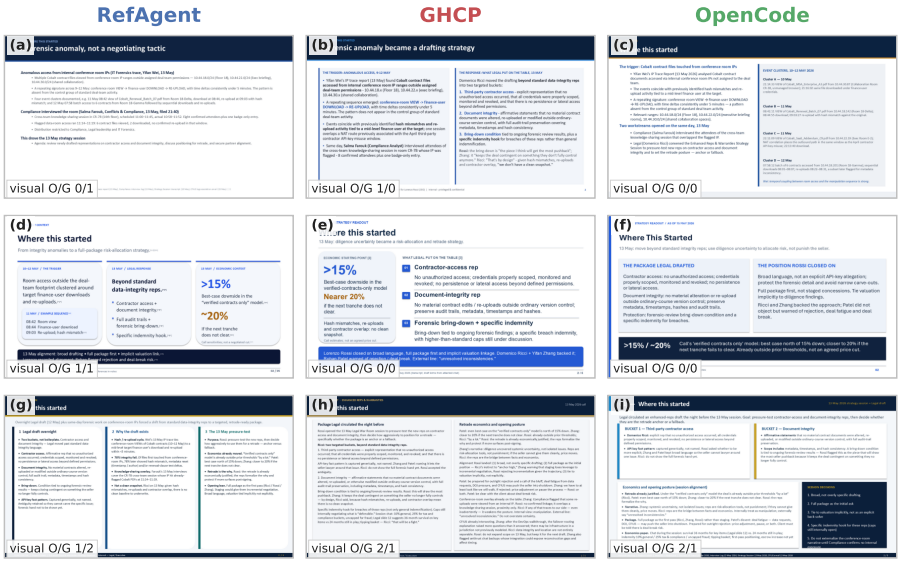


Figure A.9: Lower visual-quality PowerPoint gallery for the Ashcroft *Cobalt Enhanced Reps & Warranties Strategy* query. Rows: Opus-5 max/GPT-6-Astra max/Grok-4.6 xhigh; columns: RefAgent/GHCP/OpenCode. `visual O/G 0/1` means 0 of 5 rules failed under Opus-4.6 and 1 of 5 under GPT-5.5 (lower is better). Slide 2 shown.

B EXPERIMENTAL PROTOCOL AND RESULTS

B.1 ANALYSIS CONTRACT AND EXCLUSIONS

The observational unit is one tenant × case task evaluated under a specific model × harness configuration. Every evaluation task remains in the task-completion denominator. A task without a gradeable response or artifact contributes a non-completion. When a gradeable output is retained, the assertion judgment determines the content outcome. The main comparison is the 11 model settings represented across all three harnesses. Twp addtional settings, GPT-6-Astra low/high, are benchmarked only on RefAgent and GHCP, as the requested effort could not be enforced on OpenCode.

B.2 RUN ISOLATION AND TRACE VALIDATION

Each system is evaluated as configured, including its own planning loop, turn budget, and built-in behaviors. Every subprocess receives a fresh MCP configuration bound to exactly one tenant and its users and launches from a temporary directory outside the repository. No mutable registration, working-directory output, or conversation state is shared across runs. Before aggregation, every trace is checked against its intended tenant, acting identity, and ego-centric slice; traces that do not satisfy this identity contract are quarantined rather than scored.

B.3 COMPLETE RESULT TABLES

Table B.1 reports the complete per-task assertion table for all 11 main model settings. Table B.2 reports the corresponding overall completion counts, and Table B.3 reports the complete paired GHCP–OpenCode analysis. Together these tables are the complete numerical counterpart to the frontier-only Table 2 and Figure 3 in the main paper.

Table B.1: Complete per-task *assertion pass rate*, mean ± standard deviation for all 11 model, evaluated at the highest reasoning effort across three agent harnesses.

		Chatbot		Artifact		
Arm	Model	Retr.	Multi.	Word	Excel	PPTX
BioNova (pharma R&D)						
RefAgent	Claude-Opus-4.7	0.70 ± 0.43	0.43 ± 0.37	0.89 ± 0.08	0.64 ± 0.37	0.66 ± 0.31
RefAgent	Claude-Opus-4.8	0.76 ± 0.39	0.45 ± 0.38	0.87 ± 0.24	0.68 ± 0.37	0.67 ± 0.31
RefAgent	Claude-Opus-5	$\mathbf{0.80 \pm 0.38}$	0.50 ± 0.43	0.96 ± 0.05	0.77 ± 0.30	$\mathbf{0.82 \pm 0.27}$
RefAgent	GPT-5.4	0.48 ± 0.40	0.50 ± 0.34	0.77 ± 0.20	0.51 ± 0.30	0.59 ± 0.31
RefAgent	GPT-5.5	0.51 ± 0.39	0.53 ± 0.30	0.88 ± 0.11	0.62 ± 0.30	0.67 ± 0.29
RefAgent	GPT-5.6-Luna	0.55 ± 0.39	0.57 ± 0.35	0.89 ± 0.12	0.61 ± 0.30	0.68 ± 0.29
RefAgent	GPT-5.6-Sol	0.53 ± 0.40	0.61 ± 0.33	0.82 ± 0.31	0.59 ± 0.35	0.72 ± 0.28
RefAgent	GPT-5.6-Terra	0.58 ± 0.39	0.57 ± 0.33	0.89 ± 0.10	0.56 ± 0.29	0.68 ± 0.30
RefAgent	GPT-6-Astra	0.62 ± 0.39	$\mathbf{0.71 \pm 0.30}$	0.95 ± 0.06	$\mathbf{0.79 \pm 0.33}$	$\mathbf{0.82 \pm 0.27}$
RefAgent	Grok-4.5 (high)	0.73 ± 0.40	0.59 ± 0.38	0.89 ± 0.24	0.64 ± 0.35	0.71 ± 0.28
RefAgent	Grok-4.6 (xhigh)	$\mathbf{0.80 \pm 0.36}$	0.65 ± 0.34	$\mathbf{0.99 \pm 0.03}$	0.70 ± 0.36	0.75 ± 0.30
GHCP	Claude-Opus-4.7	0.71 ± 0.43	0.43 ± 0.39	0.93 ± 0.07	0.71 ± 0.35	0.73 ± 0.29
GHCP	Claude-Opus-4.8	0.73 ± 0.42	0.41 ± 0.40	0.90 ± 0.09	0.66 ± 0.38	0.70 ± 0.32
GHCP	Claude-Opus-5	$\mathbf{0.83 \pm 0.35}$	0.48 ± 0.44	0.97 ± 0.05	0.75 ± 0.35	0.77 ± 0.33
GHCP	GPT-5.4	0.60 ± 0.41	0.48 ± 0.35	0.88 ± 0.21	0.60 ± 0.28	0.72 ± 0.25
GHCP	GPT-5.5	0.58 ± 0.39	0.53 ± 0.33	0.85 ± 0.24	0.70 ± 0.27	0.72 ± 0.26
GHCP	GPT-5.6-Luna	0.62 ± 0.40	0.51 ± 0.35	0.91 ± 0.07	0.55 ± 0.32	0.73 ± 0.29
GHCP	GPT-5.6-Sol	0.65 ± 0.44	0.53 ± 0.37	0.95 ± 0.07	0.65 ± 0.35	0.76 ± 0.28
GHCP	GPT-5.6-Terra	0.66 ± 0.40	0.50 ± 0.37	0.89 ± 0.08	0.58 ± 0.35	0.40 ± 0.41
GHCP	GPT-6-Astra	0.75 ± 0.36	$\mathbf{0.74 \pm 0.30}$	0.97 ± 0.04	$\mathbf{0.79 \pm 0.30}$	$\mathbf{0.84 \pm 0.24}$
GHCP	Grok-4.5 (high)	0.71 ± 0.40	0.57 ± 0.37	0.96 ± 0.08	0.72 ± 0.32	0.71 ± 0.31
GHCP	Grok-4.6 (xhigh)	0.81 ± 0.36	0.55 ± 0.39	$\mathbf{0.98 \pm 0.04}$	0.73 ± 0.32	0.80 ± 0.26
OpenCode	Claude-Opus-4.7	0.69 ± 0.43	0.36 ± 0.38	0.94 ± 0.06	0.72 ± 0.35	0.73 ± 0.28
OpenCode	Claude-Opus-4.8	0.76 ± 0.40	0.40 ± 0.40	0.92 ± 0.10	0.70 ± 0.34	0.73 ± 0.30
OpenCode	Claude-Opus-5	$\mathbf{0.85 \pm 0.34}$	0.54 ± 0.44	$\mathbf{0.98 \pm 0.04}$	$\mathbf{0.76 \pm 0.37}$	0.76 ± 0.32
OpenCode	GPT-5.4	0.61 ± 0.40	0.52 ± 0.35	0.89 ± 0.11	0.57 ± 0.29	0.71 ± 0.25
OpenCode	GPT-5.5	0.51 ± 0.38	0.52 ± 0.33	0.89 ± 0.09	0.63 ± 0.30	0.74 ± 0.27
OpenCode	GPT-5.6-Luna	0.50 ± 0.39	0.46 ± 0.35	0.93 ± 0.05	0.56 ± 0.34	0.73 ± 0.23
OpenCode	GPT-5.6-Sol	0.57 ± 0.43	0.51 ± 0.34	0.95 ± 0.06	0.62 ± 0.36	0.73 ± 0.27
OpenCode	GPT-5.6-Terra	0.50 ± 0.40	0.49 ± 0.33	0.91 ± 0.08	0.58 ± 0.28	0.64 ± 0.34
OpenCode	GPT-6-Astra	0.74 ± 0.36	$\mathbf{0.70 \pm 0.31}$	0.97 ± 0.05	$\mathbf{0.76 \pm 0.36}$	0.79 ± 0.27
OpenCode	Grok-4.5 (high)	0.74 ± 0.39	0.61 ± 0.36	$\mathbf{0.98 \pm 0.06}$	0.73 ± 0.31	0.72 ± 0.28
OpenCode	Grok-4.6 (xhigh)	0.78 ± 0.38	0.49 ± 0.39	$\mathbf{0.98 \pm 0.07}$	0.71 ± 0.39	$\mathbf{0.80 \pm 0.26}$

Table B.1 (continued). Complete per-task assertion pass rate: Ashcroft (financial advisory).

		Chatbot		Artifact		
Arm	Model	Retr.	Multi.	Word	Excel	PPTX
Ashcroft (financial advisory)						
RefAgent	Claude-Opus-4.7	0.81 ± 0.33	0.63 ± 0.37	0.85 ± 0.13	0.72 ± 0.30	0.78 ± 0.24
RefAgent	Claude-Opus-4.8	0.88 ± 0.28	0.62 ± 0.32	0.86 ± 0.15	0.62 ± 0.35	0.73 ± 0.28
RefAgent	Claude-Opus-5	$\mathbf{0.96 \pm 0.16}$	$\mathbf{0.78 \pm 0.28}$	$\mathbf{0.98 \pm 0.04}$	$\mathbf{0.89 \pm 0.19}$	0.83 ± 0.22
RefAgent	GPT-5.4	0.60 ± 0.35	0.52 ± 0.34	0.77 ± 0.26	0.48 ± 0.27	0.57 ± 0.24
RefAgent	GPT-5.5	0.58 ± 0.38	0.58 ± 0.32	0.79 ± 0.26	0.62 ± 0.28	0.67 ± 0.29
RefAgent	GPT-5.6-Luna	0.58 ± 0.39	0.63 ± 0.30	0.88 ± 0.13	0.60 ± 0.29	0.72 ± 0.28
RefAgent	GPT-5.6-Sol	0.56 ± 0.38	0.60 ± 0.32	0.79 ± 0.31	0.69 ± 0.28	0.72 ± 0.26
RefAgent	GPT-5.6-Terra	0.58 ± 0.38	0.58 ± 0.30	0.83 ± 0.12	0.55 ± 0.28	0.65 ± 0.24
RefAgent	GPT-6-Astra	0.63 ± 0.37	0.77 ± 0.26	0.95 ± 0.07	0.85 ± 0.24	$\mathbf{0.86 \pm 0.23}$
RefAgent	Grok-4.5 (high)	0.82 ± 0.32	0.66 ± 0.29	0.91 ± 0.10	0.71 ± 0.33	0.76 ± 0.24
RefAgent	Grok-4.6 (xhigh)	0.87 ± 0.30	0.70 ± 0.34	0.88 ± 0.24	0.71 ± 0.41	0.80 ± 0.28
GHCP	Claude-Opus-4.7	0.86 ± 0.30	0.59 ± 0.32	0.93 ± 0.11	0.76 ± 0.30	0.80 ± 0.25
GHCP	Claude-Opus-4.8	0.89 ± 0.28	0.59 ± 0.33	0.89 ± 0.11	0.76 ± 0.29	0.80 ± 0.23
GHCP	Claude-Opus-5	0.91 ± 0.26	0.73 ± 0.29	$\mathbf{0.97 \pm 0.05}$	0.81 ± 0.31	0.85 ± 0.22
GHCP	GPT-5.4	0.70 ± 0.34	0.43 ± 0.35	0.78 ± 0.24	0.59 ± 0.27	0.68 ± 0.35
GHCP	GPT-5.5	0.68 ± 0.34	0.54 ± 0.34	0.92 ± 0.08	0.66 ± 0.33	0.75 ± 0.21
GHCP	GPT-5.6-Luna	0.64 ± 0.38	0.55 ± 0.36	0.91 ± 0.10	0.61 ± 0.31	0.70 ± 0.23
GHCP	GPT-5.6-Sol	0.75 ± 0.36	0.56 ± 0.33	0.93 ± 0.09	0.66 ± 0.35	0.83 ± 0.23
GHCP	GPT-5.6-Terra	0.75 ± 0.36	0.49 ± 0.35	0.91 ± 0.11	0.50 ± 0.36	0.54 ± 0.42
GHCP	GPT-6-Astra	0.74 ± 0.36	$\mathbf{0.81 \pm 0.26}$	$\mathbf{0.97 \pm 0.08}$	$\mathbf{0.90 \pm 0.20}$	$\mathbf{0.87 \pm 0.22}$
GHCP	Grok-4.5 (high)	0.83 ± 0.32	0.61 ± 0.35	0.95 ± 0.09	0.75 ± 0.31	0.78 ± 0.29
GHCP	Grok-4.6 (xhigh)	$\mathbf{0.93 \pm 0.25}$	0.63 ± 0.36	0.96 ± 0.07	0.78 ± 0.31	0.80 ± 0.28
OpenCode	Claude-Opus-4.7	0.83 ± 0.35	0.57 ± 0.34	0.91 ± 0.08	0.74 ± 0.32	0.80 ± 0.24
OpenCode	Claude-Opus-4.8	0.87 ± 0.30	0.64 ± 0.30	0.89 ± 0.12	0.68 ± 0.34	0.78 ± 0.27
OpenCode	Claude-Opus-5	$\mathbf{0.93 \pm 0.24}$	0.74 ± 0.26	$\mathbf{0.99 \pm 0.03}$	$\mathbf{0.84 \pm 0.25}$	$\mathbf{0.88 \pm 0.21}$
OpenCode	GPT-5.4	0.67 ± 0.36	0.51 ± 0.35	0.85 ± 0.11	0.55 ± 0.23	0.73 ± 0.23
OpenCode	GPT-5.5	0.52 ± 0.34	0.54 ± 0.30	0.88 ± 0.10	0.66 ± 0.30	0.77 ± 0.24
OpenCode	GPT-5.6-Luna	0.54 ± 0.38	0.49 ± 0.30	0.95 ± 0.07	0.61 ± 0.30	0.76 ± 0.24
OpenCode	GPT-5.6-Sol	0.67 ± 0.38	0.46 ± 0.33	0.93 ± 0.08	0.61 ± 0.38	0.77 ± 0.24
OpenCode	GPT-5.6-Terra	0.64 ± 0.38	0.51 ± 0.35	0.90 ± 0.11	0.57 ± 0.34	0.72 ± 0.22
OpenCode	GPT-6-Astra	0.79 ± 0.32	$\mathbf{0.78 \pm 0.23}$	0.95 ± 0.08	$\mathbf{0.84 \pm 0.27}$	0.84 ± 0.25
OpenCode	Grok-4.5 (high)	0.86 ± 0.32	0.62 ± 0.31	0.91 ± 0.09	0.81 ± 0.27	0.78 ± 0.27
OpenCode	Grok-4.6 (xhigh)	0.92 ± 0.25	0.57 ± 0.37	0.91 ± 0.24	0.72 ± 0.36	0.76 ± 0.31

Table B.2: Task-completion-rate results for 11 model represented in all three harnesses. The Claude-Opus-4.8 reference-agent cell has 473 executions; all other cells have 474. Astra low/high effort runs are reported in the Appendix case study.

Model	Effort	RefAgent	GHCP	OpenCode
Claude-Opus-4.7	max	286/474 (60.3%)	298/474 (62.9%)	283/474 (59.7%)
Claude-Opus-4.8	max	304/473 (64.3%)	298/474 (62.9%)	304/474 (64.1%)
Claude-Opus-5	max	**365/474 (77.0%)**	**352/474 (74.3%)**	**363/474 (76.6%)**
GPT-5.4	max	174/474 (36.7%)	216/474 (45.6%)	228/474 (48.1%)
GPT-5.5	max	206/474 (43.5%)	245/474 (51.7%)	208/474 (43.9%)
GPT-5.6-Luna	max	229/474 (48.3%)	245/474 (51.7%)	206/474 (43.5%)
GPT-5.6-Sol	max	234/474 (49.4%)	290/474 (61.2%)	246/474 (51.9%)
GPT-5.6-Terra	max	221/474 (46.6%)	244/474 (51.5%)	217/474 (45.8%)
GPT-6-Astra	max	307/474 (64.8%)	339/474 (71.5%)	332/474 (70.0%)
Grok-4.5	high	300/474 (63.3%)	303/474 (63.9%)	315/474 (66.5%)
Grok-4.6	xhigh	333/474 (70.2%)	333/474 (70.2%)	324/474 (68.3%)

Table B.3: Paired comparison of GHCP (B) and OpenCode (C) over all tasks. The effect is the tenant-fixed paired risk difference in percentage points; intervals use 10,000 source-cluster bootstrap draws. B-only/C-only gives discordant outcomes. Exact paired test p-values are Benjamini–Hochberg corrected across the 11 complete model–effort comparisons.

Model	Effort	Pairs	B completion	C completion	B–C (pp)	95% CI	B-only/C-only	BH q
GPT-5.6-Sol	max	474	61.2%	51.9%	**+9.2**	[+5.7, +12.8]	63/19	< 0.001
GPT-5.6-Luna	max	474	51.7%	43.5%	+8.2	[+3.6, +12.8]	81/42	0.003
GPT-5.5	max	474	51.7%	43.9%	+7.8	[+3.3, +12.0]	77/40	0.003
GPT-5.6-Terra	max	474	51.5%	45.8%	+5.7	[+1.3, +10.1]	71/44	0.041
Claude-Opus-4.7	max	474	62.9%	59.7%	+3.2	[−0.4, +6.8]	45/30	0.232
Grok-4.6	xhigh	474	70.3%	68.4%	+1.9	[−1.5, +5.3]	38/29	0.401
GPT-6-Astra	max	474	71.5%	70.0%	+1.5	[−2.1, +5.3]	45/38	0.561
Claude-Opus-4.8	max	474	62.9%	64.1%	-1.2	[−4.8, +2.3]	34/40	0.561
Claude-Opus-5	max	474	**74.3%**	**76.6%**	-2.3	[−5.5, +0.9]	25/36	0.314
Grok-4.5	high	474	63.9%	66.5%	-2.5	[−5.7, +0.6]	25/37	0.297
GPT-5.4	max	474	45.6%	48.1%	-2.6	[−7.0, +1.9]	51/63	0.401

B.4 Scaling and effort analyses

GPT-6-Astra reasoning effort. The low and high Astra runs are not part of the main results because OpenCode could not enforce those requested effort levels. They remain useful as a case study in arms A and B. Task completion rate rises from 54.9% to 62.5% to 64.8% in RefAgent and from 64.3% to 71.3% to 71.5% in GHCP for low, high, and max effort. Paired source-cluster intervals show clear low-to-high gains (+7.6 points in RefAgent and +7.0 in GHCP) but little high-to-max gain (+2.3 and +0.2).

Table B.4: Astra reasoning-effort case study. Low and high OpenCode cells are unavailable because the requested effort could not be enforced; only the max setting enters the main three-arm comparison.

Effort	RefAgent	GHCP	OpenCode
high	296/474 (62.5%)	338/474 (71.3%)	*not available*
low	260/474 (54.9%)	305/474 (64.3%)	*not available*
max	**307/474 (64.8%)**	**339/474 (71.5%)**	**332/474 (70.0%)**

Figure 4c visualizes the tenant-specific effort trajectories in the main paper; the table above provides the corresponding pooled counts and rates.

Model-series contrasts. Figure B.1 displays the tenant-specific GPT flagship trajectory, while Table B.5 reports paired within-arm contrasts for the Claude Opus and Grok version series, the GPT flagship and GPT-5.6 size series, and the Astra effort case study.

Table B.5: Paired model-scaling contrasts within each arm. Positive values favor the model or effort in the left column. Intervals use the same source-cluster bootstrap as the GHCP–OpenCode analysis.

Arm	Higher setting	Lower setting	Pairs	Difference (pp)	95% CI
RefAgent	Opus-4.8 (max)	Opus-4.7 (max)	473	+3.8	[−0.2, +7.9]
RefAgent	Opus-5 (max)	Opus-4.8 (max)	473	**+12.6**	[**+8.3**, **+17.0**]
RefAgent	Grok-4.6 (xhigh)	Grok-4.5 (high)	474	**+7.0**	[**+3.2**, **+10.6**]
RefAgent	GPT-5.5 (max)	GPT-5.4 (max)	474	**+6.7**	[**+2.5**, **+10.9**]
RefAgent	GPT-5.6-Sol (max)	GPT-5.5 (max)	474	**+5.9**	[**+1.7**, **+10.3**]
RefAgent	GPT-5.6-Terra (max)	GPT-5.6-Luna (max)	474	-1.7	[−5.9, +2.6]
RefAgent	GPT-5.6-Sol (max)	GPT-5.6-Terra (max)	474	+2.7	[−1.7, +7.0]
GHCP	Opus-4.8 (max)	Opus-4.7 (max)	474	+0.0	[−3.6, +3.6]
GHCP	Opus-5 (max)	Opus-4.8 (max)	474	**+11.4**	[**+7.4**, **+15.4**]
GHCP	Grok-4.6 (xhigh)	Grok-4.5 (high)	474	**+6.3**	[**+2.5**, **+10.1**]
GHCP	GPT-5.5 (max)	GPT-5.4 (max)	474	**+6.1**	[**+1.5**, **+10.6**]
GHCP	GPT-5.6-Sol (max)	GPT-5.5 (max)	474	**+9.5**	[**+5.1**, **+13.9**]
GHCP	GPT-5.6-Terra (max)	GPT-5.6-Luna (max)	474	-0.2	[−4.5, +4.0]
GHCP	GPT-5.6-Sol (max)	GPT-5.6-Terra (max)	474	**+9.7**	[**+5.7**, **+13.9**]
OpenCode	Opus-4.8 (max)	Opus-4.7 (max)	474	**+4.4**	[**+0.8**, **+8.0**]
OpenCode	Opus-5 (max)	Opus-4.8 (max)	474	**+12.5**	[**+8.7**, **+16.3**]
OpenCode	Grok-4.6 (xhigh)	Grok-4.5 (high)	474	+1.9	[−1.9, +5.7]
OpenCode	GPT-5.5 (max)	GPT-5.4 (max)	474	-4.2	[−8.9, +0.4]
OpenCode	GPT-5.6-Sol (max)	GPT-5.5 (max)	474	**+8.0**	[**+3.5**, **+12.4**]
OpenCode	GPT-5.6-Terra (max)	GPT-5.6-Luna (max)	474	+2.3	[−2.0, +6.5]
OpenCode	GPT-5.6-Sol (max)	GPT-5.6-Terra (max)	474	**+6.2**	[**+1.8**, **+10.4**]
RefAgent	GPT-6-Astra (high)	GPT-6-Astra (low)	474	**+7.6**	[**+3.6**, **+11.5**]
RefAgent	GPT-6-Astra (max)	GPT-6-Astra (high)	474	+2.3	[−1.3, +5.9]
GHCP	GPT-6-Astra (high)	GPT-6-Astra (low)	474	**+7.0**	[**+3.2**, **+11.0**]
GHCP	GPT-6-Astra (max)	GPT-6-Astra (high)	474	+0.2	[−3.4, +3.8]

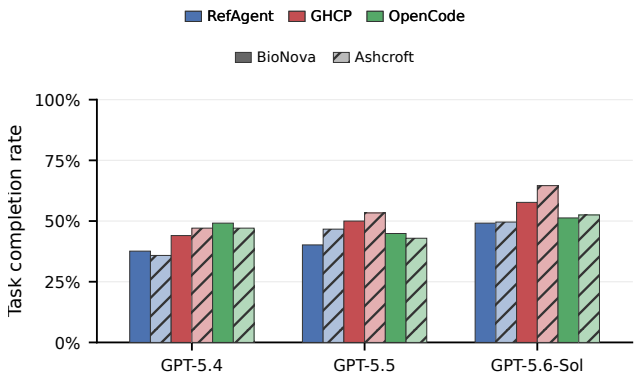


Figure B.1: GPT flagship version scaling across RefAgent, GHCP, and OpenCode. Grouped bars compare GPT-5.4, GPT-5.5, and GPT-5.6-Sol. Each model group orders harness pairs as RefAgent, GHCP, and OpenCode; BioNova uses solid bars and Ashcroft uses hatched bars. The task-completion-rate axis is zero-based from 0–100%.

C MECHANISM AND OPERATIONAL BEHAVIOR

This section separates execution-process evidence from outcome comparisons: first the observed tool-use profile, then recorded execution events included in the denominator.

C.1 TOOL-USE PROFILE AND TRACE DEPTH

Figure 5 summarizes total observed tool calls and task completion rate by trace depth. The top panels compare mean trace depth, and the bottom panels stratify task completion rate by call-count bin. These summaries use the main three-arm matrix. They do not identify a causal effect of calling more tools: difficult and retried tasks accumulate calls, especially in the 25+ bin. MCP-only counts are available for the CLI arms; the complete trace count is available for all three arms.

Table C.1: Mean calls per task by functional category in the 11 model results. Columns use the search/artifact split and report all three arms.

	Search			Artifact		
Category	A	B	C	A	B	C
Read	4.26	**4.53**	4.37	7.37	**7.57**	7.37
File	1.59	**2.05**	1.93	3.80	4.09	**4.13**
Email	2.18	2.59	**2.60**	**1.50**	1.33	1.27
Meeting	1.61	**1.98**	1.91	1.16	**1.34**	1.26
Chat	2.46	**3.15**	3.06	2.27	2.44	**2.45**
People	0.46	**0.63**	0.59	1.46	**1.52**	1.36
Authoring	**0.01**	**0.01**	**0.01**	2.35	4.37	**5.46**
Total calls	12.6	15.0	14.5	19.9	22.7	23.3
Task completion rate	53.3%	**57.0%**	53.0%	63.0%	**67.3%**	67.1%

C.2 PROVIDER CONTENT-FILTER INTERCEPTION

The experiment records 196 provider content-filter events. They are concentrated in BioNova Claude-family runs in GHCP and OpenCode, plus one Ashcroft Opus 4.8 record in each production arm. The benchmark treats the provider policy as part of the as-configured system, so every affected execution remains in the denominator.

Table C.2: Provider content-filter events in affected cells. Every affected execution remains in the task-completion denominator; scoring is retained when a gradeable output is available.

Tenant	Model	Arm	Blocked	Rate
Ashcroft	Opus-4.8	GHCP	**1/240**	**0.4%**
Ashcroft	Opus-4.8	OpenCode	**1/240**	**0.4%**
BioNova	Opus-4.7	GHCP	23/234	9.8%
BioNova	Opus-4.8	GHCP	32/234	13.7%
BioNova	Opus-4.7	OpenCode	32/234	13.7%
BioNova	Opus-4.8	OpenCode	37/234	15.8%
BioNova	Opus-5	GHCP	30/234	12.8%
BioNova	Opus-5	OpenCode	40/234	17.1%

C.3 EXECUTION-EVENT CENSUS

Table C.3 reports all classified execution and judging events in the analysable set. Every affected execution remains in the task-completion denominator, and instrument events remain explicitly marked.

Table C.3: Recorded execution and judging events in the 17,537-record analysable set. Every affected execution remains in the task-completion denominator.

Event category	Records
Provider content filter	**196**
Agent timeout	43
No answer delivered	10
Harness error	5
Judge failure	3
Agent process failure	2
Empty judge response	1

D Robustness and sensitivity analyses

This section tests whether the reported ordering persists under the reevaluation results and under alternative completion thresholds.

D.1 Evaluator-model sensitivity

Intra-model re-judge results The intra-model evaluator-sensitivity analysis re-judges four model–arm cells using the same GPT-5.4-mini model. Table D.1 reports judge and re-judge completion counts, changed outcomes, and assertion decisions.

Inter-model re-judge results We additionally draw a fixed-seed stratified sample of 1,000 records from 1,884 eligible records across task class, tenant, tested model, and arm. The GPT-5.4-mini judge and Claude Haiku-4.5 re-judge agree on 92.2% of task-completion outcomes ($\kappa = 0.82$). The GPT-5.4-mini judge and both re-judges agree on 899/1,000 outcomes.

Table D.1: Sensitivity to the re-judge results.

Model	Arm	Records	Re-judge Pass	Judge Pass	Outcome changes	Assertion changes
Opus-5	GHCP	474	347	**352**	9	43
GPT-5.6-Sol	GHCP	474	280	**290**	24	101
Opus-5	OpenCode	474	361	**363**	20	45
GPT-5.6-Sol	OpenCode	474	**247**	246	27	104

D.2 Task-completion-threshold sensitivity

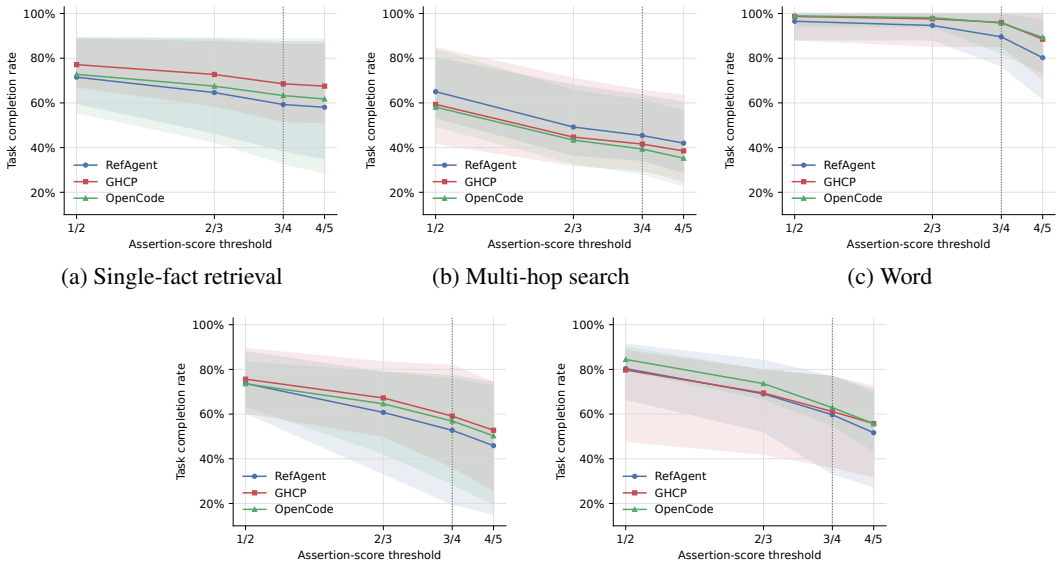
The assertion threshold for task completion is varied independently over every available main-matrix cell. Tables D.3 and D.2 provide the complete configuration and task-family sweeps. Rank correlations with the $3/4$ baseline remain at least 0.945, indicating that the broad ordering remains similar across the selected thresholds.

Table D.2: Task-completion-rate sensitivity for the balanced 11-setting matrix across four assertion thresholds. Every task assigned for evaluation remains in the denominator.

Arm	Task family	$1/2$	$2/3$	$3/4$	$4/5$
RefAgent	Artifact	81.0%	71.0%	63.0%	55.1%
GHCP	Artifact	81.9%	74.2%	**67.3%**	**61.2%**
OpenCode	Artifact	**83.1%**	**75.0%**	67.1%	60.4%
RefAgent	Search	68.7%	58.0%	53.3%	51.1%
GHCP	Search	**69.5%**	**60.7%**	**57.0%**	**55.1%**
OpenCode	Search	66.5%	57.1%	53.0%	50.4%

Table D.3: Model–arm task-completion-rate sensitivity for the 11 settings in the main three-arm comparison.

Model	Effort	1/2	2/3	3/4	4/5
RefAgent					
Opus-4.7	max	72.8%	64.3%	60.3%	57.0%
Opus-4.8	max	75.9%	68.3%	64.3%	59.0%
Opus-5	max	**84.4%**	**79.1%**	**77.0%**	**75.3%**
GPT-5.4	max	61.4%	46.2%	36.7%	31.4%
GPT-5.5	max	68.1%	52.3%	43.5%	38.2%
GPT-5.6-Luna	max	69.8%	55.7%	48.3%	43.0%
GPT-5.6-Sol	max	67.1%	55.7%	49.4%	44.1%
GPT-5.6-Terra	max	69.2%	55.5%	46.6%	39.7%
GPT-6-Astra	max	79.3%	69.6%	64.8%	62.0%
Grok-4.5	high	76.4%	68.1%	63.3%	59.9%
Grok-4.6	xhigh	79.7%	73.8%	70.3%	68.6%
GHCP					
Opus-4.7	max	74.5%	67.7%	62.9%	60.3%
Opus-4.8	max	75.5%	68.1%	62.9%	59.7%
Opus-5	max	80.4%	76.2%	**74.3%**	**72.8%**
GPT-5.4	max	64.8%	52.7%	45.6%	40.9%
GPT-5.5	max	71.5%	58.6%	51.7%	47.3%
GPT-5.6-Luna	max	69.6%	57.6%	51.7%	46.0%
GPT-5.6-Sol	max	72.8%	64.3%	61.2%	57.6%
GPT-5.6-Terra	max	64.8%	56.5%	51.5%	47.0%
GPT-6-Astra	max	**84.4%**	**76.8%**	71.5%	69.4%
Grok-4.5	high	74.9%	68.1%	63.9%	60.5%
Grok-4.6	xhigh	80.0%	73.8%	70.3%	67.9%
OpenCode					
Opus-4.7	max	71.5%	64.8%	59.7%	58.4%
Opus-4.8	max	75.3%	69.8%	64.1%	60.5%
Opus-5	max	81.9%	**77.6%**	**76.6%**	**73.4%**
GPT-5.4	max	69.6%	57.6%	48.1%	41.8%
GPT-5.5	max	67.1%	51.7%	43.9%	38.0%
GPT-5.6-Luna	max	64.1%	51.1%	43.5%	38.2%
GPT-5.6-Sol	max	66.5%	57.0%	51.9%	47.3%
GPT-5.6-Terra	max	64.1%	52.5%	45.8%	40.3%
GPT-6-Astra	max	**83.3%**	74.7%	70.0%	67.5%
Grok-4.5	high	78.9%	71.5%	66.5%	62.7%
Grok-4.6	xhigh	74.1%	70.5%	68.4%	65.2%



(a) Single-fact retrieval (b) Multi-hop search (c) Word

(d) Excel (e) PowerPoint

Figure D.1: Task-type-specific task-completion sensitivity using the same encoding as Figure 3(c). In each panel, lines show the mean configuration task completion rate for RefAgent, GHCP, and OpenCode; shaded ribbons span the model range across the 11 settings represented in all three arms. Results pool both tenants; all panels share the same 10–103% y-scale, and the vertical reference marks the reported $3/4$ assertion threshold.

E Extended descriptions of main figures

The main-text captions identify the panels, encodings, and essential denominator rules without repeating the surrounding argument. For completeness, this section provides the fuller panel-by-panel descriptions and interpretive context, including aggregation choices, exclusions, and the conclusions supported by each display.

Figure 1: enterprise search-then-produce workflow. *(1) Enterprise work scenario:* the benchmark setup creates an access-safe employee task and source-derived assertions from a synthetic tenant. The employee operates over an identity-bound view of email, files, meetings, and chat; tasks cover retrieval-augmented generation (RAG), including single-fact and multi-hop synthesis, plus editable Word, Excel, and PowerPoint deliverables. *(2) Digital-worker execution:* the task reaches the agent, which searches and reads employee-visible evidence through the shared MCP interface and uses a task-local workspace to reconcile, synthesize, and write. *(3) Deliverable and evaluation:* the final work product is normalized to text: response text is used directly, whereas Office artifact content is extracted before entering the same assertion judge. The dashed path shows that assertions bypass the agent; the resulting pass rate c defines task completion when it meets the chosen threshold (0.75 in this study). Office artifacts also branch to native rendering, and their page screenshots enter visual judges that count failures of five presentation rules. The resulting 0–5 visual score is lower-is-better.

Figure 2: tenant heterogeneity, dynamics, and complexity. *(a)* Per-tenant complexity statistics on a log scale show that both tenants span thousands of content units and graph nodes and edges over a ∼100-day horizon. *(b)* Content is distributed across five content modalities. *(c)* The tenants contain 15–16 file types with no dominant format. *(d)* Content accrues in a bursty, event-driven stream, dispersing a task's evidence across non-adjacent days and increasing retrieval difficulty. Each acting identity sees only $8.1\%/13.3\%$ of the communications corpus; the complete tenant statistics are reported in Appendix A.2.

Figure 3: capability and robustness profile. *(a,b)* Task completion rate at the 0.75 threshold is shown for Opus-5, GPT-6-Astra, and Grok-4.6 on each tenant. Bars are grouped into RefAgent, GHCP, and OpenCode blocks and use within-harness color gradients; the value above each bar is the task completion rate. Table 2 reports mean assertion pass rate for the same frontier-model subset. Appendix Tables B.1, B.2, and B.3 provide complete per-task, overall-completion, and paired results for all 11 settings. Astra low/high/max effort is shown in Figure 4c, with detailed counts in Appendix B.4. *(c)* For each arm, the line is the mean configuration task completion rate and the shaded ribbon spans the model range as the assertion threshold varies. The vertical reference marks the reported $3/4$ threshold.

Figure 4: model $\times$ harness analysis. *(a)* Lines and points compare Opus-4.7, Opus-4.8, and Opus-5 within each harness. Agent is encoded by marker shape and redundant color; tenant is encoded only by line style, with solid BioNova and dashed Ashcroft trajectories. *(b)* The same encoding compares GPT-5.6 Luna, Terra, and Sol. *(c)* Low, high, and max requested effort are compared for GPT-6-Astra; OpenCode low/high cells are unavailable, so only its two tenant-specific max markers appear, each with a short solid or dashed tenant cue. GPT flagship version scaling is reported separately in Appendix Figure B.1.

Figure 5: agentic-trace analysis. *(a,b)* Mean calls are stacked by functional category for search and artifact tasks. *(c,d)* Grouped bars show task completion rate by total-call bin for RefAgent, GHCP, and OpenCode. Every subfigure contains its applicable legend. The search/artifact call axes are tightened to 0–16 and 0–26 calls, respectively, and call-bin panels remove unused horizontal margins. The 26+ bin combines hard tasks, retries, and timeouts and therefore has lower task completion rate.