OFFICEBENCH-PRO: BENCHMARKING AGENTS FOR PROFESSIONAL OFFICE WORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Professional Office work requires evidence, decisions, and editable files to remain consistent through inspection and reuse. We introduce OFFICEBENCH-PRO, comprising 130 expert-authored and 47 synthesized tasks across professional domains and Office formats. The tasks couple factual, reasoning, and aesthetic requirements, with native structures essential to analysis or delivery. Source-grounded synthesis and evolution develop business demands under independent task and rubric review. An Office Agent-as-a-Judge traces evidence across files, tests required native behavior, and inspects rendered presentation. Across six models under a shared execution protocol, the highest overall scores are 51.3% on expert tasks and 50.8% on synthesized tasks. In a separate fourteen-task archival cohort, only 18 of 84 submissions reach a 50% diagnostic threshold in all three dimensions without a business veto. File inspection reveals unsupported source values, missing calculation dependencies, and layout failures. These results expose limitations in how the evaluated models ground decisions in evidence, maintain native functionality, and coordinate content with visual presentation.

1 INTRODUCTION

Large language model applications are expanding from conversational assistance and software development to broader professional work. Claude Cowork[1] and ChatGPT Work[2] allow users to delegate multistep tasks involving files and applications. Much of this work is carried out through documents, spreadsheets, and presentations that support analysis and decision making. Producing these deliverables requires agents to reconcile source materials, draw justified conclusions, and organize the results for their intended audience. The value of this delegation depends on whether the resulting Office files can support informed decisions and remain usable during subsequent review and revision.

Recent benchmarks increasingly evaluate agents on professional work. One line focuses on individual Office formats, with SpreadsheetBench (Ma et al., 2024) and MBABench (Yen et al., 2026) assessing spreadsheet manipulation and financial modeling, and PresentBench (Chen et al., 2026) evaluating slide generation. Another line extends evaluation to broader professional workflows. GDPval (Patwardhan et al., 2025) compares economically valuable deliverables against expert work. JobBench (Li et al., 2026) selects tasks that professionals want to delegate and checks professional reasoning through fact-anchored rubric chains. Workspace-Bench (Tang et al., 2026) examines agents' ability to discover and use file dependencies in large, heterogeneous workspaces. However, these benchmarks only partially capture the interdependent demands of professional Office work, where source-grounded analysis, native file functionality, and professional presentation must jointly serve the same business objective. A correct spreadsheet result can still yield a misleading or unusable work product if the accompanying report draws an unsupported conclusion or the presentation obscures the comparison needed for a decision.

We introduce OFFICEBENCH-PRO to evaluate the coordinated quality of professional Office deliverables. Each task comprises a workspace of related source files, a workplace instruction, and a private rubric. Given the workspace and instruction, the agent creates or revises Office files that are

[1] https://claude.com/product/cowork
[2] https://chatgpt.com/work

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.

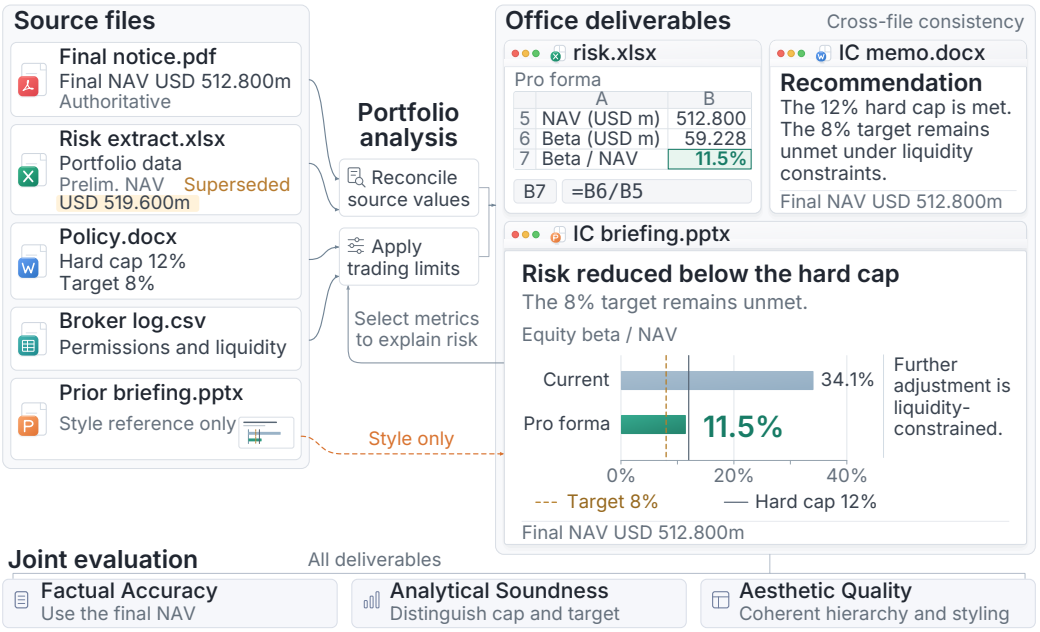


Figure 1: An illustrative OFFICEBENCH-PRO task connecting source evidence, native calculations, and coordinated Office deliverables, jointly evaluated for factual accuracy, analytical soundness, and aesthetic quality. NAV denotes net asset value.

evaluated for *Factual Accuracy*, *Analytical Soundness*, and *Aesthetic Quality*. These are coequal requirements that constrain one another. In the portfolio review in Figure 1, authoritative source values determine risk estimates, and the workbook, memo, and slides must distinguish satisfying a hard risk limit from attaining a lower target. That comparison determines which quantities must be computed and how they should be organized and presented. Native calculations preserve the analytical dependencies, while the visual hierarchy and style of tables and charts must support a coherent, professional presentation. Every task requires native Office structures to play a substantive role in analysis or delivery.

We recruit domain experts to construct 130 tasks from their professional experience. In parallel, we develop a framework that synthesizes tasks and aligned rubrics from existing workspace files. The framework combines source-grounded design, execution-guided evolution, and independent review, contributing 47 tasks to the benchmark. We then adapt Agent-as-a-Judge (Zhuge et al., 2024) to trace claims in submissions to source evidence, test required native behavior, and inspect rendered outputs. Across six models under a shared execution protocol, the best overall scores are 51.3% on expert tasks and 50.8% on synthesized tasks. No model leads all three quality dimensions in either collection. Further analysis reveals weaknesses in how the evaluated models ground professional judgments in source evidence, implement required native functionality, and coordinate content with visual presentation.

Our contributions are summarized as follows.

- We develop a task design framework that couples factual accuracy, analytical soundness, and aesthetic quality around a shared workplace objective, with native Office structures essential to analysis or delivery.
- We construct OFFICEBENCH-PRO with 130 expert-authored and 47 synthesized tasks across professional domains and Office formats. Our synthesis framework grounds task design and evolution in workspace evidence, with independent reviews of task validity and rubric alignment.

- We establish an evaluation protocol that combines source tracing, tests of required native behavior, and inspection of rendered outputs. Model comparisons and selected file analyses reveal shortcomings in evidence use, native functionality, and professional presentation.

2 Related Work

Table 1: Documented task coverage and released evaluation protocols. Attributes may describe different tasks or evaluation settings.

	Input		Output		Evaluation			
Benchmark	Multi-file	Cross-file	Formats	Multi-type	Rubrics	Aesth.	Native	Judge
Office benchmarks								
DocFormBench (2026)	—	—	W · ·	—	—	Fmt	S	R
SpreadsheetBench (2024)	—	—	· X ·	—	—	—	—	R
MBABench (2026)	✓	✓	· X ·	—	Shared	✓	S	M
Finch (2026)	✓	✓	W X ·	—	Shared	Fmt	S	M/H
SlidesBench (2025)	✓	—	· · P	—	Shared	✓	S	R+M
PPTArena (2025)	—	—	· · P	—	Task	✓	S	M
SlidesGen-Bench (2026)	—	—	· · P	—	—	✓	S	R+M
PresentBench (2026)	✓	✓	· · P	—	Task	✓	—	M
OfficeBench (2024)	✓	✓	W X ·	✓	—	—	—	R
DocOps (2026)	✓	✓	W X P	—	—	Fmt	S	R
OmegaUse-OfficeVal (2026)	✓	✓	W X P	—	Task	Fmt	S	R
General professional work benchmarks								
GDPval (2025)	✓	✓	W X P	✓	Task	✓	S	H
APEX (extended) (2025)	Text	—	· · ·	—	Task	—	—	M
APEX-Agents (2026)	✓	✓	W X P	—	Task	—	—	M
JobBench (2026)	✓	✓	W X P	✓	Task	Fmt	S	M
Workspace-Bench (2026)	✓	✓	W X P	✓	Task	✓	S	A
Expert consulting (2026)	✓	✓	· · P	—	Shared	Fmt	—	H
OfficeBench-Pro	✓	✓	W X P	✓	Task	✓	S+B	A

Aesth.: design quality (✓), formatting/readability (Fmt). Native: structural checks (S), input-change probes (B). R: programmatic; M: fixed model pipeline; A: active agent judge; H: human.

Office benchmarks. Office evaluation has expanded from testing isolated file operations to judging the quality of complete deliverables. Existing suites cover document formatting and spreadsheet transformations (Rao et al., 2026; Ma et al., 2024), slide generation, understanding, and editing (Ge et al., 2025; Huang et al., 2025; Ofengenden et al., 2025; Yang et al., 2026; Jang et al., 2026), and application workflows or native-state preservation (Wang et al., 2024; Jiang et al., 2026; Zhou et al., 2026). MBABench evaluates financial accuracy, formula robustness, and presentation (Yen et al., 2026); Finch checks financial workflows through formula, structural, and rendered evidence (Dong et al., 2026); PresentBench judges source fidelity and slide design (Chen et al., 2026). In OfficeBench-Pro, each task ties factual, analytical, and aesthetic requirements to one workplace objective: source evidence grounds the analysis, while visual goals shape the comparisons the analysis must support.

General professional work benchmarks. In occupational assignments, agents work toward a specified goal while choosing how to interpret the evidence and carry out the work (Patwardhan et al., 2025; Vidgen et al., 2025; 2026; Asthana et al., 2026). JobBench awards rubric credit only when all checks along a fact-anchored reasoning chain pass (Li et al., 2026). Workspace-Bench pairs rubric assessment with dependency-recognition metrics from execution traces, using an active judge (Tang et al., 2026). OfficeBench-Pro admits tasks only when removing native Office structure, while retaining task-relevant content, would substantially weaken a key challenge or leave an essential requirement unmet. Its judge traces sources, inspects native and rendered evidence, and probes update behavior when the task requires it. Table 1 summarizes coverage and evaluation protocols.

3 THE OFFICEBENCH-PRO BENCHMARK

Task formulation. OFFICEBENCH-PRO evaluates agents on professional assignments that require creating or revising Office files from heterogeneous source materials. Agents must select relevant evidence, reconcile information across sources, and turn their analysis into the requested files. A task $\tau = (\mathcal{W}, \mathcal{I}, \mathcal{R})$ comprises a workspace, an instruction, and a private rubric. The agent receives $(\mathcal{W}, \mathcal{I})$ and submits the required files O. The instruction specifies the work objective, audience, essential constraints, and deliverables, leaving evidence selection, analysis, and implementation to the agent. The rubric translates requirements grounded in the instruction and source materials into independently decidable binary criteria.

Coupled quality requirements. Each task defines three coequal quality requirements for one workplace objective. *Factual Accuracy* concerns correct content and consistency with sources and instructions. *Analytical Soundness* concerns valid calculations and inferences, and professional judgments grounded in evidence and consistent with task constraints. *Aesthetic Quality* concerns whether information hierarchy, layout, typography, color, and chart design form a coherent, professional presentation suited to the audience and purpose. Source evidence constrains the analysis, analytical results guide content and presentation, and visual goals and constraints shape information selection and the comparisons the analysis must support.

Native Office structures. Tasks must rely on native structure in at least one key stage of interpreting inputs, performing the work, or delivering results. Formulas and references preserve calculation dependencies, chart bindings connect charts to their data, and document styles and slide layouts support structured revision and consistent presentation. We use a substitution test for inclusion: replacing the relevant files with text, CSV, or static images that retain task-relevant content but remove native structure must substantially weaken a key challenge or leave an essential requirement unmet. The task determines the required editability and update behavior.

Benchmark composition. OFFICEBENCH-PRO contains 177 tasks: 130 authored by domain experts and 47 synthesized from existing workspace files. The collection spans finance, law, healthcare, education, arts and humanities, and technical domains. Required outputs include Word documents, Excel workbooks, and PowerPoint presentations; 147 tasks require more than one of these formats. Section 4 describes how both task collections are constructed and reviewed.

4 BENCHMARK CONSTRUCTION

4.1 EXPERT-AUTHORED TASKS

We include 130 tasks authored by 56 domain professionals in clinical practice, law, teaching and research, finance, engineering, and design. Authors draw on their professional experience to define work objectives, intended audiences, and business constraints. They organize supporting input files, write instructions specifying the required deliverables, and translate these requirements into rubrics covering Factual Accuracy, Analytical Soundness, and Aesthetic Quality. They also provide reference deliverables for review.

Quality review combines author self-checks with independent spot checks by a quality-control team, examining consistency among the instructions, input files, and rubrics. Outputs from model trials provide concrete examples for authors and reviewers to check whether individual criteria are clear and verifiable. Issues identified in the tasks or rubrics are returned to the authors for revision and rechecked.

4.2 WORKSPACE-BASED SYNTHESIS

We design a multi-agent framework for task construction and evolution that transforms workspace files into candidate task packages comprising an instruction, an accompanying workspace, and a rubric. We use JobBench workspace files and public Zenodo collections, without the original JobBench task descriptions. As shown in Figure 2, the framework combines workspace screening, task and rubric

construction, and automated evolution guided by execution feedback, with independent reviewers for tasks and rubrics.

Workspace screening and task synthesis. This stage determines whether a source workspace can support realistic, feasible Office work and constructs the instruction and task workspace from accepted materials. The Task Designer checks file readability, source traceability, and input sufficiency, as well as whether the file relationships can support tasks that depend on native Office structures and couple factual, reasoning, and aesthetic requirements. Workspaces that fail these admission checks are excluded before synthesis.

For accepted workspaces, the designer uses source precedence and business constraints to define objectives, audiences, and deliverables. It selects and adapts input files, writes the instruction, and records sources and derivations for critical claims. Targeted recalculation and trials of essential Office operations check feasibility, while an independent Task Reviewer revisits the sources to verify task grounding and solvability.

Rubric design and review. This stage converts task completion and quality requirements into explicit, verifiable assessment criteria. The Rubric Designer uses the instruction, source materials, and professional context to construct criteria across the three quality dimensions, specifying evidence sources, decision boundaries, and acceptable alternatives for each. Aesthetic criteria follow the audience, purpose, and applicable design references. An independent Rubric Reviewer checks coverage, evidential support, and fairness, including whether the criteria impose unjustified restrictions or duplicate penalties. Execution starts only after the rubric is accepted.

Execution feedback and automated evolution. Evolution combines cross-module diagnosis with analysis of task redesign opportunities to generate objectives and constraints grounded in source evidence. The Execution Agent runs the task with a model distinct from the design model, and the Evaluation Agent provides criterion-level judgments and evidence. The Evolution Analyst cross-checks the instruction, workspace, rubric, and execution and evaluation evidence to locate construction defects, runtime failures, and judgment errors, prioritizing upstream issues that would invalidate subsequent difficulty assessment.

Once quality and measurement validity are established, the analyst contrasts the work demanded by the current task with the work that the source workspace can support. It analyzes dependencies among evidence, business constraints, key decisions, and native Office operations to identify work requirements not yet captured by the task. It also examines whether recombining existing information, constraints, and file relationships can introduce new independent judgments, coordinated operations, or trade-offs between analysis and presentation. These findings inform an assessment of remaining scope for evolution and source limitations, from which the analyst automatically proposes revision objectives constrained by source support, business necessity, and feasibility.

The findings are translated into structured design guidance specifying the business objectives to extend, the supporting source files, the permitted scope of changes, and the factual and professional constraints to preserve. Model identities, scores, and previous answers are withheld from designers. The Task Designer automatically reformulates the instruction and workspace, while the Rubric Designer rebuilds the corresponding criteria for the revised requirements. Each revision must justify the added work by the business function lost if it were removed and pass independent task and rubric review. Execution and evaluation evidence from the new task version returns to the analyst to reassess remaining opportunities and guide the next iteration. Before automated acceptance, the Final Auditor independently checks professional plausibility, solvability, and assessment fairness without access to execution outputs, scores, or earlier review opinions. Appendix B specifies the difficulty criterion and revision budgets.

All synthesized tasks undergo expert review before inclusion in the benchmark.

5 Agent-as-a-Judge for Office Deliverables

We evaluate factual accuracy, analytical soundness, and aesthetic quality against fixed task criteria. A model judge given fixed evidence cannot investigate omitted native relationships; a general-purpose agent judge may rely on property queries or cached values without Office-specific checks. We adapt

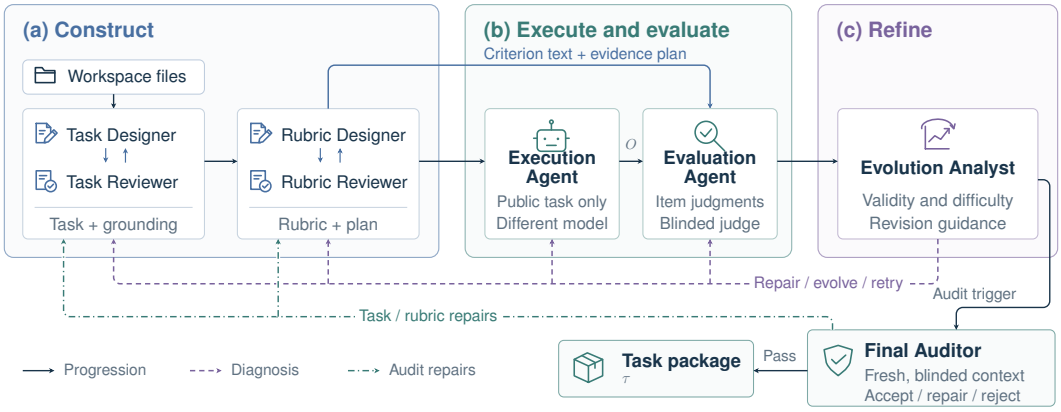


Figure 2: Task synthesis through nested multi-agent loops. Separate Task and Rubric Designer–Reviewer pairs construct and validate tasks and assessment criteria. Dark arrows show progression; the blue path carries private assessment inputs to the Evaluation Agent. Feedback directs repairs, difficulty evolution, or retries. The Final Auditor independently checks task quality.

Agent-as-a-Judge (Zhuge et al., 2024) through an Office evaluation environment, criterion-guided verification, and blinded judgments with evidence review.

An environment for Office evaluation. The judge receives the instruction, the same workspace used for execution, submitted Office files, and criterion texts. OfficeCLI[3] and complementary parsing, rendering, and recalculation tools support joint inspection of sources, native structure, and presentation.

Criterion-guided verification. Prompts specify the evidence and checks for each criterion. The judge traces source records and calculation rules, checks derivations, and tests conclusions against evidence and task constraints. Native checks follow formulas, references, and chart bindings through upstream calculations to verify data selection and source support. Reproducibility criteria require forced recalculation on scratch copies and comparison with saved values. For update-dependent criteria, the judge changes an admissible input and checks downstream values and charts against the response required by the task and source data.

For visual criteria, format compliance does not establish layout or chart quality. Prompts explicitly require active OfficeCLI rendering and visual inspection of the resulting views; property queries or preview generation alone are insufficient. The judge assesses hierarchy, layout, color coordination, and chart design against the task's purpose and applicable references, covers each criterion's scope, and retains the views used. Calculation or rendering anomalies require independent verification before attribution to distinguish tool limitations from submission defects.

Blinded judgments and evidence review. The controller hides the evaluated model's identity and scoring metadata, including criterion weights, to limit irrelevant influences. Prompts prohibit new requirements and accept equivalent formulas or visual designs that meet the fixed criteria. The judge investigates insufficient or conflicting evidence and reviews decisions in the same context before returning a Boolean verdict and evidence per criterion. The controller then aggregates these judgments deterministically (Appendix A.1).

6 Experiments

6.1 Data and Experimental Configuration

OfficeBench-Pro contains 130 expert-authored and 47 synthesized tasks spanning finance, law, healthcare, education, arts and humanities, and technical work in Chinese and English. The

[3] https://github.com/iOfficeAI/OfficeCLI

workspaces contain 786 input files, or 4.4 per task; 147 tasks require multiple Office formats. Appendix A specifies the counting and scoring conventions.

The headline comparison evaluates six models once per task through Claude Code. GPT-5.5 in Codex judges the outputs. Execution and evaluation use separate containers, with OfficeCLI, Python, compatible renderers, and Chinese and English fonts; each has a two-hour limit and no additional token or tool-call cap. Scores are task-macro averages within each source group. Separate archival analyses use 120 submissions on 20 tasks, with the original rubric versions; their results do not reconstruct the headline table. Fourteen archival tasks support joint dimension scores, and targeted assistance uses these same fourteen tasks (Appendix F).

6.2 Overall scores conceal different delivery profiles

Table 2: Scores by task source (%; higher is better). Overall scores include task-specific weights and business vetoes; dimension scores are normalized separately. Bold and underlining identify the highest and second-highest entries. Mean averages the six models.

	Expert tasks				Synthesized tasks			
Model	Score	Facts	Reasoning	Aesthetics	Score	Facts	Reasoning	Aesthetics
Claude Fable 5	**51.3**	**50.1**	45.7	<u>68.0</u>	**50.8**	**49.8**	<u>45.1</u>	**67.4**
GPT-5.6-Sol	<u>49.2</u>	<u>49.4</u>	<u>47.6</u>	54.0	45.4	46.0	43.7	51.1
Claude Opus 5	48.3	48.5	**47.9**	55.8	<u>48.0</u>	<u>47.9</u>	**47.6**	55.2
Qwen3.8-Max	47.0	46.3	45.0	58.5	42.2	42.4	40.7	52.9
Kimi-K3	44.8	42.0	45.8	**69.4**	38.6	36.7	38.8	<u>61.0</u>
Seed2.1-Pro	27.4	27.0	33.5	45.3	25.8	25.8	31.4	42.5
Mean	**44.7**	**43.9**	**44.3**	**58.5**	**41.8**	**41.4**	**41.2**	**55.0**

The best overall scores are 51.3% on expert tasks and 50.8% on synthesized tasks (Table 2). The six-model means are 44.7% and 41.8%. All six models score lower on the synthesized collection, by 0.3–6.2 percentage points; this is a between-collection comparison, not an estimate of the effect of synthesis.

No model leads all three dimensions. Claude Fable 5 leads facts; Claude Opus 5 leads reasoning at 47.9% and 47.6%. Kimi-K3 ranks fifth overall in both collections but first and second in aesthetics, while its factual scores are second lowest. A model's stronger presentation can therefore coexist with less reliable content. Dimension means measure attainment against different requirements; they are not interchangeable scales of intrinsic ability.

6.3 Construction must separate harder work from a changed assessment

Four sampled synthesis trajectories reach scores below 60% after two to four revisions. Their mean score declines from 80.3% to 55.0%, with temporary increases in two trajectories (Figure 3, Appendix B). Applying a transported initial assessment contract to the final outputs attributes 16.2 points of the 25.3-point decline to output changes on shared requirements and 9.1 points to assessment changes (Table 4). The full decline should therefore not be read as a gain in difficulty under fixed requirements.

A separate biostatistics design comparison shows the intended additional work: source precedence determines the event rate, which affects sample size, recruitment feasibility, budget, and the committee recommendation. The workbook must preserve this chain and the briefing must communicate its consequences. Expert review supplies an independent quality check: eleven experts reviewed all 47 synthesized tasks; 16 were adopted unchanged and 31 after minor expert repairs. These are adoption paths for the reviewed set, not an automatic-generation yield estimate.

6.4 Validation of the Office Judge

We evaluate whether the Office Agent-as-a-Judge identifies defects in submitted files consistently with human assessment. Three submissions from each of twenty tasks yield 60 submissions and 2,589

submission–criterion pairs. Two assessors independently inspect the instructions, source materials, native files, and renderings without seeing automated judgments; disagreements are adjudicated to form a common reference. We compare no rendered views, optional rendering, and required visual inspection using GPT-5.5 in Codex, with identical task and rubric versions, submissions, native inspection tools, and time budgets.

Table 3: Agreement with the human reference on 2,589 submission–criterion pairs. Error rates concern aesthetic criteria. Values are percentages except for Cohen's κ; bold marks the best value in each column.

Rendering condition	Overall agreement ↑	Aesthetic agreement ↑	Aesthetic false acceptance ↓	Aesthetic false rejection ↓	Overall κ ↑
No rendered views	86.0	73.9	56.4	9.4	0.71
Optional rendering	89.6	84.5	31.4	6.8	0.79
Required inspection	**92.9**	**93.4**	**9.6**	**5.0**	**0.85**

False acceptance labels an unmet requirement as satisfied; false rejection labels a satisfied requirement as unmet. Their human-reference denominators are 188 and 340 aesthetic criteria, respectively.

Aggregate agreement can conceal missed defects. Without rendered views, the judge achieves 86.0% agreement overall yet falsely accepts 56.4% of aesthetic requirements judged unmet by humans (Table 3). With optional rendering, false acceptance remains at 31.4%. Requiring visual inspection raises overall agreement from 89.6% to 92.9% and aesthetic agreement from 84.5% to 93.4%. Aesthetic false acceptance falls to 9.6%, while false rejection also decreases from 6.8% to 5.0%. Reductions in both error directions indicate better discrimination between satisfied and unmet requirements.

Initial appearance does not establish update behavior. We examine six controlled workbook implementations at two input states (Appendix C). A linked implementation and a constant-expression implementation both display 18.0% at a net asset value (NAV) of 100, with identical renderings; each workbook contains two formulas and one chart. After NAV changes to 125, the linked ratio correctly becomes 14.4%, while the constant expression remains at 18.0%. An offset formula responds to the change but returns an incorrect 16.4%; an equivalent formula remains correct at both states. These controls distinguish object presence, input responsiveness, and correct calculation. For update-dependent requirements, the judge must verify the expected numerical and chart responses while accepting valid equivalent implementations.

7 Limitations

The benchmark covers selected domains, languages, and workplace conventions; it does not estimate labor-market effects or all capabilities of a deployed assistant. Public files may occur in pretraining. The headline uses one execution per model and task, and the archival diagnostic cohort is purposive. Its heterogeneous historical rubrics limit cross-task comparability; missing dimension denominators are excluded explicitly. Native-check rates compare requirement families, not randomized task difficulty. Assistance combines changes in available structure and execution strategy, so it identifies intervention effects under that configuration rather than a unique cognitive cause. Task bootstrap intervals do not measure run-to-run variation. Four evolution trajectories and one workbook setting do not establish population-wide construction yield or judge reliability.

8 Conclusion

OfficeBench-Pro evaluates coordinated professional Office delivery: source-grounded decisions, consistent expression across files, professional presentation, and the native structures needed for further use. It complements broad professional-work evaluations and specialized file benchmarks. The experiments show why this scope matters: strengths on separate dimensions need not coincide, editable objects can lack working dependencies, and improving average scores can leave joint attainment unchanged. These findings favor development and evaluation that trace the complete chain from admissible evidence through decisions to reusable, audience-appropriate deliverables.

AI USE STATEMENT

Generative AI supported task and experimental design, analysis code, literature retrieval, figure preparation, and manuscript writing. Agents also perform synthesis, execution, and evaluation in the benchmark framework. The authors are responsible for the study design, evidence, analysis, and scientific claims.

ETHICS STATEMENT

Professional materials can contain personal or commercial information. Their use and release require source-level permissions, de-identification, and data minimization; public availability does not itself establish redistribution rights. Expert participation requires informed agreement and appropriate compensation. Cases describe task requirements and file behavior without identifying reviewers or individuals represented in source records. Outputs are intended for research evaluation.

REPRODUCIBILITY STATEMENT

The appendices specify scoring, task cohorts, criterion tags, intervention controls, bootstrap units, native probes, and harness accounting. Deterministic analysis code binds numerical tables and plots to their input records. Artifact checks preserve original submissions and use scratch copies for recalculation and rendering. Headline, archival, controlled, and external comparisons retain separate populations and protocols; their scores are not pooled into a new leaderboard.

REFERENCES

Tanmay Asthana, Aman Saksena, and Divyansh Sahu. Evaluating Deep Research Agents on Expert Consulting Work: A Benchmark with Verifiers, Rubrics, and Cognitive Traps, 2026. URL https://arxiv.org/abs/2605.17554.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D. Hwang, Peter Jansen, Varsha Kishore, Bodhisattwa Prasad Majumder, Aakanksha Naik, Sigal Rahamimov, Kyle Richardson, Amanpreet Singh, Harshit Surana, Aryeh Tiktinsky, Rosni Vasu, Guy Wiener, Chloe Anastasiades, Stefan Candra, Jason Dunkelberger, Dan Emery, Rob Evans, Malachi Hamada, Regan Huff, Rodney Kinney, Matt Latzke, Jaron Lochner, Ruben Lozano-Aguilera, Cecile Nguyen, Smita Rao, Amber Tanaka, Brooke Vlahos, Peter Clark, Doug Downey, Yoav Goldberg, Ashish Sabharwal, and Daniel S. Weld. AstaBench: Rigorous Benchmarking of AI Agents with a Scientific Research Suite, 2026. URL https://arxiv.org/abs/2510.21652.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering, 2025. URL https://arxiv.org/abs/2410.07095.

Xin-Sheng Chen, Jiayu Zhu, Pei lin Li, Hanzheng Wang, Shuojin Yang, and Meng-Hao Guo. PresentBench: A Fine-Grained Rubric-Based Benchmark for Slide Generation, 2026. URL https://arxiv.org/abs/2603.07244.

Haoyu Dong, Pengkun Zhang, Yan Gao, Xuanyu Dong, Yilin Cheng, Mingzhe Lu, Zikun Zhu, Adina Yakefu, and Shuxin Zheng. Finch: Benchmarking Finance & Accounting across Spreadsheet-Centric Enterprise Workflows, 2026. URL https://arxiv.org/abs/2512.13168.

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, 2025. URL https://arxiv.org/abs/2501.00912.

Zheng Huang, Xukai Liu, Tianyu Hu, Kai Zhang, and Ye Liu. PPTBench: Towards Holistic Evaluation of Large Language Models for PowerPoint Layout and Design Understanding, 2025. URL https://arxiv.org/abs/2512.02624.

Daesik Jang, Morgan Lindsay Heisler, Linzi Xing, Yifei Li, Edward Wang, Ying Xiong, Yong Zhang, and Zhenan Fan. DECKBench: Benchmarking Multi-Agent Frameworks for Academic Slide Generation and Editing, 2026. URL https://arxiv.org/abs/2602.13318.

Jiazhen Jiang, Boxi Cao, Lingyong Yan, Yaojie Lu, Hongyu Lin, Shuaiqiang Wang, Dawei Yin, Xianpei Han, and Le Sun. DocOps: A Verifiable Benchmark for Autonomous Agents in Complex Document Operations, 2026. URL https://arxiv.org/abs/2607.19865.

Yuetai Li, Yichen Feng, Zhangchen Xu, Zixian Ma, Kaiyuan Zheng, Fengqing Jiang, Xinghua Sun, Rulin Shao, Zichen Chen, Yue Huang, Xinyang Han, Brian Lee, Kayla Xu, Shenglai Zeng, Hang Hua, Xiangliang Zhang, Basel Alomair, Ranjay Krishna, Luke Zettlemoyer, Pang Wei Koh, Bhaskar Ramasubramanian, Luyao Niu, Xiang Yue, and Radha Poovendran. JobBench: Aligning Agent Work With Human Will, 2026. URL https://arxiv.org/abs/2605.26329.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. SpreadsheetBench: Towards Challenging Real World Spreadsheet Manipulation, 2024. URL https://arxiv.org/abs/2406.14991.

Michael Ofengenden, Yunze Man, Ziqi Pang, Liang-Yan Gui, and Yu-Xiong Wang. PPTArena: A Benchmark for PowerPoint Editing, 2025. URL https://arxiv.org/abs/2512.03042.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie S. Kim, Patrick Chao, Samuel Miserendino, Gildas Chabot, David Li, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating AI Model Performance on Real-World Economically Valuable Tasks, 2025. URL https://arxiv.org/abs/2510.04374.

Shihao Rao, Liang Li, Jiapeng Liu, Tong Lin, Bing Li, Xiyan Gao, Peng Fu, Jing Huang, and Can Ma. What to Format and How: A Benchmark and Workflow Approach for Document Formatting, 2026. URL https://arxiv.org/abs/2606.01936.

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.

Bertie Vidgen, Abby Fennelly, Evan Pinnix, Julien Benchek, Daniyal Khan, Zach Richards, Austin Bridges, Calix Huang, Kanishka Sahu, Abhishek Kottamasu, Bo Ma, Ben Hunsberger, Isaac Robinson, Akul Datta, Chirag Mahapatra, Dominic Barton, Cass R. Sunstein, Eric Topol, Brendan Foody, and Osvald Nitski. The AI Productivity Index (APEX), 2025. URL https://arxiv.org/abs/2509.25721.

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

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.

Yunqiao Yang, Wenbo Li, Houxing Ren, Zimu Lu, Ke Wang, Zhiyuan Huang, Zhuofan Zong, Mingjie Zhan, and Hongsheng Li. SlidesGen-Bench: Evaluating Slides Generation via Computational and Quantitative Metrics, 2026. URL https://arxiv.org/abs/2601.09487.

Thomson Yen, Julian Poeltl, Harshith Srinivas Gear, Yilin Meng, Joshua Fan, Adam Shen, Yili Liu, Ali Bauyrzhan, Patrick Shea, Siri Du, Haoyang Liu, Daniel Guetta, and Hongseok Namkoong. MBABench: Evaluating LLM Agents on End-to-End Spreadsheet Tasks in Finance, 2026. URL https://arxiv.org/abs/2605.22664.

Jingbo Zhou, Yusai Zhao, Qi Bao, Jingjia Cao, Zhenghai Chen, Chang Gao, Kaiqi Guo, Muxin Guo, Mingxuan Li, Xinjiang Lu, Yanru Ma, Yixiong Xiao, Zenghui Zhang, Le Zhang, and Hua Wu. OmegaUse-OfficeVal: Benchmarking LLM Agents on Long-Horizon Office-Suite Tasks with Economic Grounding, 2026. URL https://arxiv.org/abs/2607.27155.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-Judge: Evaluate Agents with Agents, 2024. URL https://arxiv.org/abs/2410.10934.

A BENCHMARK AND EVALUATION PROTOCOL

Composition and unit of evaluation. The benchmark contains 130 expert-authored and 47 synthesized tasks. Its 786 input files yield the mean of 4.4 files reported in Section 6.1. File counts include contents from one ZIP level and exclude archive metadata. Word documents are required in 114 tasks, Excel workbooks in 156, and PowerPoint presentations in 140; 147 tasks require more than one of these formats. The overlapping counts describe required native deliverables, excluding intermediate files and additional PDF exports. Each evaluation unit comprises a task instruction, workspace, rubric, execution configuration, and submitted files. The twenty-task archive supplies qualitative cases, the native and joint diagnostics, the harness analysis, and the judge comparison. Its fourteen tasks with complete positive dimension denominators supply the assistance study. Table 2 reports the separate headline evaluation.

Configuration and separation of roles. Section 6.1 specifies the six execution models, Claude Code harness, GPT-5.5/Codex evaluator, and two-hour limits. The common Office environment provides OfficeCLI and its skills, Python libraries, renderers, and Chinese and English fonts. The execution agent receives the public task and workspace. The judge accesses the rubric and submitted files in a separate container and returns criterion-level Boolean decisions with supporting evidence. The controller applies criterion weights and business vetoes. Behavioral probes use working copies to preserve the original submission.

Criterion evidence and decision boundaries. The judge receives criterion IDs and texts, with weights, dimension labels, veto flags, the evaluated model's identity, and other judges' decisions withheld. Where provided, a frozen evaluation plan specifies source anchors, inspection methods, decision boundaries, tolerances, and acceptable alternatives; it cannot add task requirements. Existing style is normative only when the instruction or editing context requires it. The prompts require evidence for each condition within a criterion and review within the same continuing context. The controller validates criterion coverage and output format; these checks do not certify the semantic correctness of the evidence or judgments.

Visual evidence and independent checks. Visual judgments require actual inspection of rendered views. The judge retains the views used for decisions and records the inspected locations, methods, observations, and conclusions. Native queries and rendered views can share a calculation engine, so their agreement does not establish independent corroboration. Suspected tool errors require a separate verification method before a defect is attributed to the submission. Testing calculation reproducibility requires executing the formulas through forced recalculation or recalculation after clearing formula caches; opening and saving a file without executing its formulas is insufficient.

A.1 SCORING AND JOINT DIAGNOSIS

Each binary criterion has a dimension and a signed integer weight. Construction uses nonzero weights from -10 to 10 and separately declared business vetoes. A positive predicate rewards a requirement; a negative predicate identifies a penalized defect. With y_i denoting whether that predicate holds,

$$s^{\mathrm{pre}}(O) = \frac{\sum_i w_i y_i}{\sum_{i:w_i>0} w_i}, \qquad s(O) = \begin{cases} 0, & \text{a declared veto is triggered,} \\ s^{\mathrm{pre}}(O), & \text{otherwise.} \end{cases} \tag{1}$$

Dimension scores s_F, s_R, s_A apply the same normalization before the overall veto, whenever their positive-weight denominator exists. They are not averaged to recover s. To test whether strengths coincide in one submission, we additionally report

$$J_h = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \mathbf{1}\{s_F \geq h,\ s_R \geq h,\ s_A \geq h,\ \text{no veto}\}. \tag{2}$$

The common set $\mathcal{T}$ contains tasks with all three denominators. J_h is an auxiliary joint-attainment diagnostic, not a replacement score or a professional acceptance threshold.

Score interpretation. A declared veto is triggered when its positive requirement fails or its negative defect predicate holds. For task τ, the pre-veto range is $[L_\tau, 1]$, where $L_\tau = \sum_{w_i<0} w_i / \sum_{w_i>0} w_i$;

penalties can produce negative values. Multiplication by 100 gives the table units. Historical analyses preserve their recorded weights, including sentinel weights attached to explicit veto flags. These archival encodings are retained for recomputation rather than silently normalized to a new rubric. Scores are first averaged with equal task weights within each source group; the Mean row then averages the six model scores. The headline matrix reports the resulting macro-averages separately for the two task sources.

Outcome accounting and paired analysis. A missing deliverable from a valid execution is assessed by the rubric. Provider, environment, and evidence-acquisition failures are recorded separately from scored outcomes. A task receives a score once every criterion has a valid judgment. Comparisons retain the task, rubric, configuration, and output identities. Assistance analysis resamples tasks as paired blocks and retains all criteria and conditions within each sampled block (Appendix G); criteria and tool calls are not treated as independent observations.

B Construction Review and Evolution Controls

Automated synthesis settings. After task and rubric quality and the validity of execution and evaluation have been checked, construction uses the valid pre-veto score for the corresponding task version as its difficulty criterion, requiring a score strictly below 60%. Scores at or above 60% trigger further complexity evolution where the source materials support it. Complexity evolution permits at most four revisions, with a separate budget for quality repair. Independent final review starts only after the quality, validity, and difficulty conditions are met. Budget exhaustion or a lack of justified extensions stops automated revision without implying acceptance.

Expert task reviews. We analyzed task IDs, initial dispositions, and comments about tasks, rubrics, and attachments in the review records. The source workbook contains 47 records across three batches of 14, 10, and 23 tasks; respectively 2, 0, and 14 were accepted without changes. The experts completed minor repairs to the other 31 tasks, and the final benchmark uses the resulting versions. Eleven experts contributed reviews, with one initial disposition retained per task. The batches contain different tasks and review fields. The adoption statistics in Section 6.3 summarize these reviewed tasks.

Recorded evolution trajectories and design comparison. The four tasks in Figure 3 are `biostatisticians-task1`, `management_analysts-task2`, `reporters_and_correspondents-task1`, and `technical_writers-task2`. We analyzed their fifteen recorded execution-agent scores. A separate biostatistics design comparison examined an initial task and a regulatory-response design based on the same workspace materials. That analysis traced changes in source precedence, sample-size calculations, resource constraints, and Office deliverables. It is a separate design comparison, not an identified pair of rounds in the trajectory plot.

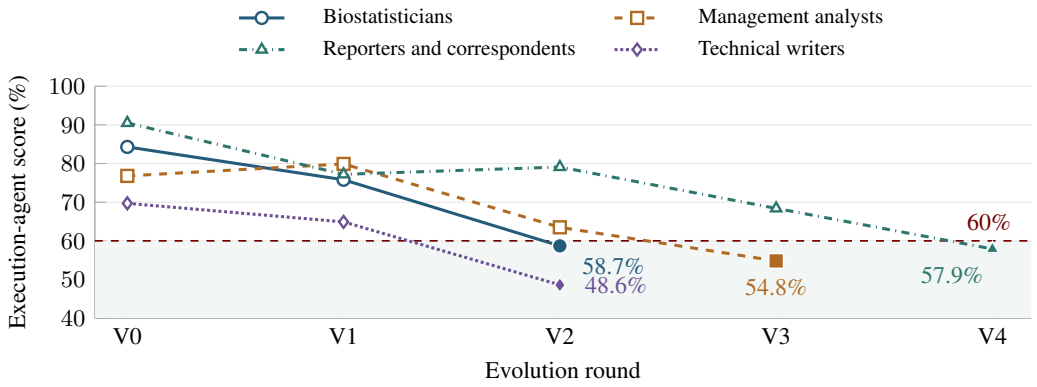


Figure 3: Execution scores for all four sampled tasks fall below the 60% threshold after two to four revisions. V0 denotes initial synthesis; V1–V4 denote subsequent revisions. The dashed line marks the threshold; filled markers identify each task's final reported version.

Separating output and assessment changes. Evolution changes the work assigned to an agent and may also change how it is assessed. Let q_0, q_1 score the initial and revised outputs O_0, O_1. Define $q_{0\to1}$ by applying the initial assessment contract to the revised task and updating only justified source-dependent bindings. When this transport preserves requirement meaning,

$$q_1(O_1) - q_0(O_0) = \underbrace{q_{0\to1}(O_1) - q_0(O_0)}_{\text{output change, common contract}} + \underbrace{q_1(O_1) - q_{0\to1}(O_1)}_{\text{assessment change}}. \tag{3}$$

This identity separates score changes under a common assessment contract from changes in the assessment itself. Comparable assessments preserve source authority, units, acceptable alternatives, veto semantics, and requirement meaning. Each task version binds its expected values to the corresponding admissible sources.

Common-contract rescoring. For the four trajectory tasks, we rescore the final output using the initial requirement meanings and update only justified source-dependent values. Table 4 applies Equation 3. The average final score on the transported initial contract is 64.1%, compared with 55.0% on the final rubric. The 16.2-point shared-contract decline and 9.1-point assessment contribution add to the 25.3-point total decline after rounding. The former measures changes on shared requirements; the latter includes added assessment burden. Both depend on valid transport of source authority, units, alternatives, and veto semantics. This four-task analysis does not establish broad construction yield or held-out transfer.

Table 4: Score changes decomposed using a common assessment contract (%). $q_0(O_0)$ evaluates the initial output, $q_{0\to1}(O_1)$ evaluates the final output on transported initial requirements, and $q_1(O_1)$ applies the final rubric. Drops are initial minus final, in percentage points.

Task	$q_0(O_0)$	$q_{0o1}(O_1)$	$q_1(O_1)$	Shared drop	Assessment drop
Biostatistics	84.3	68.4	58.7	15.9	9.7
Management analysis	76.8	63.9	54.8	12.9	9.1
Journalism	90.5	67.1	57.9	23.4	9.2
Technical writing	69.7	57.0	48.6	12.7	8.4
Mean	80.3	64.1	55.0	16.2	9.1

Executable comparability check. We executed a comparability check that preserved the declared goal, outputs, criterion templates, dimensions, weights, veto rules, and source-binding rules. Both source specifications passed the same contract, with exposure ratios of 18.0% and 14.4%. The checker rejected each mutation listed in Table 5. These checks verified the registered declarations and source bindings under the specified contract.

Table 5: Results of the comparability-contract checks. Both valid source versions passed; each isolated mutation was rejected for violating a registered condition.

Input specification	Outcome
Initial and updated source versions	Both accepted
Changed criterion weight	Rejected
Additional criterion	Rejected
Stale expected value	Rejected
Changed numerical unit	Rejected
Unregistered source	Rejected
Output-dependent binding	Rejected
Stricter visual requirement	Rejected

C Native Inspection and Judge Validation

Human reference and comparison unit. The judge study uses three submissions for each of the twenty selected tasks. Their rubrics contain 590 factual, 97 reasoning, and 176 aesthetic criteria, giving 1,770, 291, and 528 submission–criterion pairs, respectively. Two assessors independently

inspect the task, admissible sources, native submission, and rendered views without seeing the automated decisions. Disagreements are adjudicated to form a common reference. The eleven experts who review task construction assess task validity; their dispositions are separate from these labels on model submissions.

Rendering conditions. All three conditions use GPT-5.5 in Codex with the same task and rubric versions, submissions, native inspection tools, behavioral probes, and time allowance. Without rendered views, the judge can read text, formulas, object coordinates, and relationships but cannot observe page images. Optional rendering provides a renderer and image viewer without requiring their use for visual criteria. Required inspection requires the judge to observe the relevant rendered content before making a visual judgment. The task and rubric determine the inspection scope, with detailed views acquired as needed. Human labels and premarked defect locations are withheld from every condition. Each evaluation starts in a fresh context, and probes use scratch copies.

Agreement and error direction. Table 3 uses the same 2,589 complete reference–judge pairs in each condition. Agreement is the proportion of matching decisions across these pairs. The overall value is weighted by criterion counts, rather than averaging the three dimension percentages. Error analysis uses a common satisfaction orientation: a positive criterion is satisfied when its condition holds, and a negative criterion when its penalized defect is absent. With human labels as the reference, false acceptance is $\mathrm{FP}/(\mathrm{FP}+\mathrm{TN})$, and false rejection is $\mathrm{FN}/(\mathrm{TP}+\mathrm{FN})$. Cohen's $\kappa = (p_o - p_e)/(1-p_e)$ uses observed agreement p_o and chance agreement p_e computed from the two marginal label distributions. The pooled coefficient uses this normalized orientation across criteria.

Among aesthetic pairs, the reference contains 188 unmet and 340 met requirements. False acceptances decline from 106 to 59 to 18 across the three conditions; false rejections decline from 32 to 23 to 17. Higher agreement reflects both better defect detection and fewer rejections of satisfactory results. These pooled rates summarize the selected submission–criterion pairs, with tasks and files shared across rendering conditions.

A local rendering control. Task 74456 supplies the course-review briefing in Figure 4. Its negative criterion R39 assigns a ten-point penalty if crowding, overlap, or chart obstruction impairs information access on any slide. On slide 8, the R04 route label and description collide with the footer. The judge without rendered views does not trigger R39; the judge with required visual inspection and the expert reference do. The layout control moved the four row objects upward and the footer downward while preserving all text, styles, and other package members. The comparison examined the R04/footer region in both layouts under the same local overlap condition, separately from the 60-submission comparison. Native coordinates and rendered views both locate the original collision.

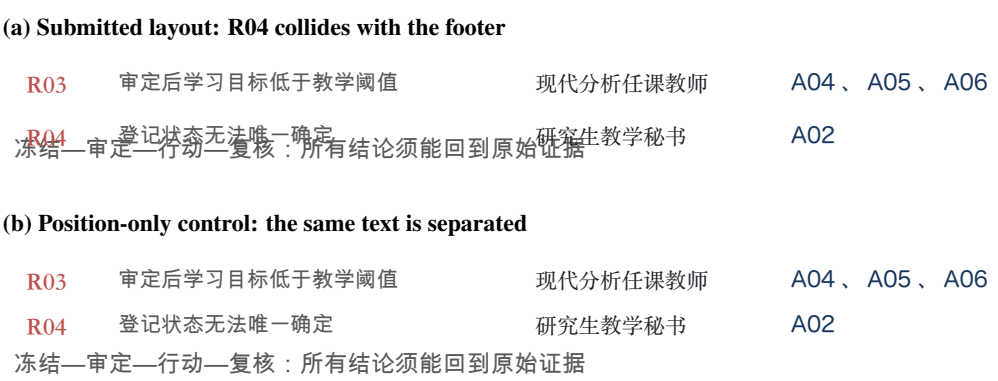
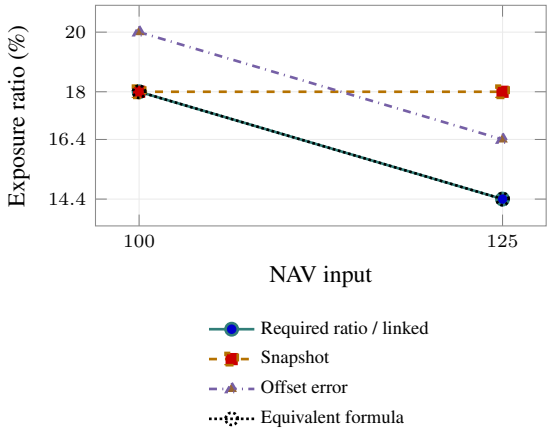


Figure 4: The same region of slide 8 before and after five objects are repositioned. R04 describes an unresolved registration status; the footer requires conclusions to remain traceable to source evidence. Without rendered views, the judge misses the collision; after visual inspection, it detects the collision, matching the expert reference. The control preserves text and styling and passes the local overlap check; it does not establish the quality of the whole presentation.

Workbook task and variants. We prepared six implementations of an exposure-monitoring workbook. The task specified an exposure of 18, net asset value (NAV) of 100, and a limit of 12%. The exposure ratio was required to equal exposure/NAV, remain editable, and update in both the summary and its native chart when NAV changed. Two coordinated palettes and equivalent formulas were permitted. The linked implementations used `B11/B10`; the offset implementations added `0.02`; the equivalent implementation used `SUM(B11)/B10`; the snapshot used `18/100`. Every implementation included the limit formula and the same chart references. The six implementations isolated formula linkage, an offset error, an equivalent calculation, and palette changes.

Independent recalculation and rendering. For each implementation, we set NAV to 100 or 125 on a scratch copy, cleared the tested formula caches, requested full recalculation, and opened and saved the workbook with LibreOfficeDev 26.8.0.0.alpha0. We compared the saved formula values and chart caches with reference ratios of 0.18 and 0.144 using a tolerance of 10^{-10}. Percentages were rounded for display. Source workbook hashes were verified before and after the probes.

We exported each recalculated workbook to PDF and rendered it at a maximum dimension of 1,800 pixels. Complete pages were compared under identical rendering settings. The linked and snapshot pages matched at NAV 100 and differed at NAV 125. The equivalent formula matched the linked implementation at both states. The navy and teal variants produced the same numerical results within each mechanism. All twelve saved workbooks retained two formulas and one chart, and the chart caches agreed with their respective summary values. The offset implementation showed that a chart can update faithfully from an incorrect upstream calculation.



Same initial appearance

Linked and snapshot renderings are pixel-identical at NAV 100. Both have two formulas and one chart.

Different update behavior

At NAV 125, the linked ratio is 14.4%; the snapshot remains 18.0%. The offset updates to an incorrect 16.4%.

A correct alternative

The equivalent formula matches the linked values and renderings in both states.

Figure 5: Native-behavior results from six workbook implementations at two input states. The plotted values were saved after independent LibreOffice recalculation; lines connect the two tested states. Numerical responses overlap for navy and teal variants, and for linked and equivalent formulas. Chart caches match each workbook's output, including the erroneous variants.

Native-behavior findings. The workbook experiment distinguished the behaviors shown in Figure 5. A snapshot containing a formula failed to update, an offset formula updated to a wrong value, and an equivalent formula remained valid. Recalculation and input changes exposed these differences within the six workbook implementations. Rendered views supported presentation checks, while formula values and chart caches recorded the corresponding calculation behavior.

For a complete evaluation with criterion-error fraction e, the correct, wrong, and unavailable shares are $(1-e, e, 0)$; a return lacking complete judgments contributes $(0, 0, 1)$. Averaging within task and then across tasks makes the three shares sum to one.

D HARNESS COMPARISON

Score and tool-use accounting. The harness study uses the same selected twenty-task set for both models. Overall scores average final task scores with equal task weights, including declared business vetoes. Tool use counts execution-stage requests after deduplication by tool-call identifier, including

failed requests and child-agent calls recorded in the trace. Evaluation calls are excluded. A turn with several requests contributes several calls; tool rounds count distinct assistant messages containing a request. One shell request may bundle several operations, and orchestration may add calls without editing an Office file. The retained Claude Code baselines use version 2.1.220, a 200-turn limit, and a bubblewrap sandbox. These settings apply to the twenty-task study and are recorded separately from the headline configuration.

The Claude Code traces contain 1,197 calls for Claude Opus 5 and 1,629 for GPT-5.6-Sol. Their combined `Bash`, `Read`, `Write`, and `Edit` counts are 1,160 and 1,158; other tools account for 37 and 471 calls. These other tools include task management, delegation, messaging, and web access. This breakdown reports execution requests by tool category. Relative changes use unrounded aggregates; Table 6 displays one decimal place.

Table 6: Mean overall score and execution tool calls on the same 20 selected tasks. Bold marks the highest score and fewest calls within each model.

Model	Harness	Score (%)	Tool calls / task
Claude Opus 5	Claude Code	**48.3**	**59.9**
Claude Opus 5	OpenClaw	46.3	71.8
GPT-5.6-Sol	Claude Code	49.2	81.5
GPT-5.6-Sol	OpenClaw	50.2	93.7
GPT-5.6-Sol	Codex	**51.2**	**69.2**

Harness effects are model-dependent. On the same 20 archival tasks, OpenClaw changes Claude Opus 5's score by -2.0 points and GPT-5.6-Sol's by $+1.0$ relative to Claude Code. GPT-5.6-Sol with Codex gains 2.0 points with 15.0% fewer logged tool calls (Table 6). Calls do not measure equivalent work across tools. Together with prior matched-model analyses (Chan et al., 2025; Bragg et al., 2026), this motivates reporting model, harness, tool access, and resource limits as the evaluated system.

Published matched-model evidence. Table 7 reproduces the GPT-4o comparison from Table 2 of MLE-bench (Chan et al., 2025). All three rows use `gpt-4o-2024-08-06`, a 24-hour attempt limit, and the benchmark's ML-engineering tasks. The reported repetitions differ: 36 for AIDE and three for each other scaffold. The table reports MLE-bench's any-medal rates with the original standard errors. The comparison shows how the execution configuration affects that benchmark's results.

Table 7: Published matched-model scaffold comparison on MLE-bench (Chan et al., 2025). Values reproduce its Table 2; $\pm$ denotes the original standard error. All rows use the same GPT-4o snapshot.

Scaffold	Any medal (%)	Repetitions
AIDE	8.7 ± 0.5	36
OpenHands	4.4 ± 1.4	3
MLAB	0.8 ± 0.5	3

E Inspected File-Level Cases

The qualitative review examined selected submissions from a twenty-task archive. Recorded judgments guided case selection, and we verified observations against the instruction, admissible sources, original submission, and compatible rendering. The analysis traced each failure to the affected file content or native structure. Appendix D describes the harness analysis of the same archive.

A factual discrepancy enters a lending recommendation. Task 74435 requests a loan-risk report and financial workbook. Three enforcement screenshots embedded in the company-information document show amounts of 27,981.4, 31,097, and 41,804.8. GPT-5.6-Sol's report instead gives 256,410, 2,868,804, and 8,849,342 yuan and uses their total of approximately 11.97 million yuan

in its lending assessment. The source images and report text establish both the discrepancy and its use as decision evidence. This case traces the source discrepancy from extraction into the lending assessment.

A statistical workbook omits its calculation chain. Task 74446 requires a traceable workbook calculation chain linking adopted item scores, statistical samples, equating parameters, and recalculated grades, and explicitly rejects hardcoded summaries. Qwen3.8-Max supplies 23 worksheets whose analytical results are constants. Package inspection finds no formula cells and no pivot, query, external-link, or macro components that provide an alternative calculation mechanism. The submission therefore lacks the required executable chain, regardless of calculations performed while generating it.

A routing instruction collides with a footer. Task 74456 requests an editable course-review briefing with controlled information density. On slide 8, the last responsibility-routing row in Claude Opus 5's submission overlaps the footer, making the route label and description difficult to read. A rendering with Chinese font support shows the collision, and native object coordinates independently confirm it. The observation documents a local legibility defect in the routing instruction.

F Archival Diagnostic Cohorts and Native Requirements

Population and judgments. The archival cohort contains one submission from each of six models for each of twenty tasks: 74417, 74418, 74421, 74428, 74430, 74432, 74435, 74441, 74442, 74446, 74448, 74452, 74456, 74463, 74464, 74467, 74468, 74470, 74549, and 74560. Each retained rubric is paired with its original Boolean decisions. This cohort supports diagnostics of those judged submissions, not a new estimate of the 177-task headline results. The analysis does not independently re-adjudicate every file; the three inspections in Appendix E provide separate direct evidence.

Dimension coverage. A dimension score needs a positive-weight denominator. Tasks 74418, 74421, 74441, 74452, 74463, and 74468 lack at least one such denominator in their historical rubrics. Some contain only negative criteria in that dimension. We exclude these six tasks from joint and assistance analyses instead of treating missing dimensions as zero or as satisfied. The remaining fourteen tasks yield 84 model–task observations. Table 8 uses $h = 0.5$; Table 9 varies the threshold on the same tasks. Counts retain negative contributions and require no declared veto.

Table 8: Joint attainment on the same 14 archival tasks per model. F, R, and A count scores of at least 50% in each dimension; joint also requires no business veto. These counts diagnose concurrent weaknesses and are not task acceptance rates.

Model	F	R	A	Joint	Joint (%)
Claude Fable 5	8/14	7/14	10/14	5/14	35.7
GPT-5.6-Sol	7/14	6/14	8/14	3/14	21.4
Claude Opus 5	8/14	6/14	9/14	3/14	21.4
Qwen3.8-Max	8/14	5/14	7/14	2/14	14.3
Kimi-K3	8/14	7/14	12/14	4/14	28.6
Seed2.1-Pro	2/14	5/14	6/14	1/14	7.1

Table 9: Joint attainment by diagnostic threshold (counts out of 14 tasks per model). Every column uses the same tasks and excludes vetoed submissions.

Model	40%	50%	60%	70%
Claude Fable 5	8	5	2	2
GPT-5.6-Sol	4	3	1	1
Claude Opus 5	7	3	1	1
Qwen3.8-Max	2	2	1	1
Kimi-K3	6	4	2	1
Seed2.1-Pro	3	1	0	0

Averaging the three marginal attainment rates gives 59.5% for Claude Fable 5 and 64.3% for Kimi-K3, while their joint rates are 35.7% and 28.6%. Marginals therefore do not identify their overlap. This comparison concerns coincidence within a submission; it does not establish a causal interaction between dimensions. Joint attainment falls to seven submissions at 60% and six at 70%, so the bottleneck is not confined to the 50% cutoff.

Native requirement selection. We code positive criteria using their retained statements. *Calculation and dependency* criteria require executable calculations, references, update behavior, or functional navigation. *Object and document structure* criteria require editable objects, document fields, or reusable master/style structures. Formula mathematics without an executable requirement, typography merely mentioning formulas, duplicate checks, and broad composite conditions are excluded. Some selected criteria combine native functionality with an exact location or implementation rule; the sensitivity row removes ten such restrictive criteria. The tags select existing decisions without rescoring the submission or adding a quality dimension.

Table 10: Fulfillment of tagged native requirements on the same five tasks (74442, 74446, 74464, 74470, 74549). Rates average criterion fractions within task, then equally over tasks. The pooled object and dependency counts are 53/90 and 14/60; pooled rates differ from task-macro means.

Model	Native objects (%)	Dependencies (%)
Claude Fable 5	56.7	10.0
GPT-5.6-Sol	66.7	10.0
Claude Opus 5	70.0	10.0
Qwen3.8-Max	63.3	30.0
Kimi-K3	83.3	10.0
Seed2.1-Pro	36.7	10.0
Mean	62.8	13.3

Table 11: All tagged archival requirements. Macro rates first average within task and model; micro rates pool criterion judgments. The sensitivity row removes ten criteria with restrictive implementation rules. Rows have different task coverage and are descriptive.

Requirement family	Tasks	Criteria	Judgments	Macro (%)	Micro (%)
Calculation / dependency	14	32	121/192	51.9	63.0
Object / document structure	8	19	74/114	72.6	64.9
Dependency sensitivity	13	22	79/132	57.4	59.8

The shared five-task cohort is determined by the availability of both tag families before computing rates. We average selected decisions within each task and model, then equally over five tasks and six models. Thus 62.8% versus 13.3% describes the same thirty model–task observations, despite different criterion counts. On all eligible tasks, the corresponding macro rates are 72.6% and 51.9%, with different populations. These all-task rates are not a matched difficulty effect. Removing the ten restrictive dependency criteria yields a 57.4% macro rate. The subset still contains unmet functional requirements; its changed task and criterion composition prevents interpreting this difference as the effect of implementation specificity alone.

A low dependency rate motivates tests that follow an input through its calculation and display paths. It does not prove that every failure arises from weak reasoning rather than an unsuitable tool, overspecified criterion, or incomplete inspection. Native controls distinguish these possibilities locally. Benchmark use should retain the cohort, rubric version, and evidence channel associated with each conclusion.

G Targeted Assistance: Controls and Detailed Results

Question and common unit. The study asks whether reducing obstacles in source use, native authoring, and final inspection improves coordinated delivery. It uses the fourteen joint-diagnostic tasks with Claude Opus 5 and GPT-5.6-Sol. Five conditions per model and task yield 140 submissions.

Table 12: Criterion IDs for the archival native diagnostic. IDs refer to the retained rubric version. Duplicate checks, pure formula descriptions, and broad composite conditions were excluded from the tag selection.

Task	Calculation / dependency	Object / document structure
74417	R21	–
74421	R5, R6, R7, R8, R62	–
74428	R5	–
74430	–	R62
74432	R4, R20, R21, R22, R23, R24, R27	–
74435	R21, R44	–
74441	R5, R8, R12	–
74442	R11, R15, R16, R19	R31
74446	R38	R6, R39, R40
74448	–	R30
74456	–	R28, R31
74463	R20	–
74464	R9, R60	R45, R48, R63
74468	R4	–
74470	R8	R5, R29
74549	R20, R26	R28, R29, R30, R38, R43, R44
74560	R5	–

The harness follows the archival Claude Code 2.1.220 configuration, with a 200-turn cap and bubblewrap sandbox; this setting is separate from the headline Docker configuration. Every condition retains the task, sources, criterion IDs, weights, veto semantics, evaluator, and resource allowance. Each is a separate execution with one sample per task and configuration; no output or private criterion judgment is shared across conditions.

Support conditions. The baseline exposes the normal tools. The *source map* adds an index of admissible files and precedence rules already stated in those files. It supplies no answer values, new business rules, computed results, or recommendation. The *native scaffold* supplies empty editable structures appropriate to the requested formats: input and calculation regions, chart/data containers, heading and caption fields, and reusable slide layouts. It supplies no completed calculations, final content, or private rubric. The *visual review* condition requires rendering and inspecting relevant final files and allows revision within the existing budget. It provides no human defect labels. The *combined* condition supplies all three supports. The requested work and scoring rule remain fixed.

The source map lowers evidence-location and precedence-management effort. The scaffold lowers mechanical authoring overhead while leaving calculations and decisions to the model. Visual review provides an opportunity to observe actual presentation. None guarantees success elsewhere: a scaffold can contain a wrong formula, and a visually improved file can preserve an unsupported conclusion.

Scores and uncertainty. Table 13 reports mean final scores, pre-veto dimension means, and $J_{0.5}$ on the same fourteen tasks in every row. Task-specific weights explain why the overall mean differs from the average dimension mean. We subtract baseline from supported scores within task and model. A paired nonparametric bootstrap samples fourteen task IDs with replacement, retaining every criterion and condition within each sampled task, over 10,000 draws. Percentile 95% intervals use seed 20260923. They describe task-panel variation under a task-independence assumption, not repeated-run or annotator variation.

Interpretation. Source support primarily improves factual and reasoning attainment; visual review mainly improves aesthetics. Combined support has the highest mean score for both models without improving joint attainment. For Claude Opus 5, 12 tasks improve in score, one is unchanged, and one worsens; for GPT-5.6-Sol, the counts are 11, one, and two. A positive mean therefore coexists with regressions. The reduction from three to two jointly attaining Claude Opus 5 submissions reflects a regression across a dimension boundary. Evaluation should inspect what improved, what regressed, and whether the full deliverable remained consistent.

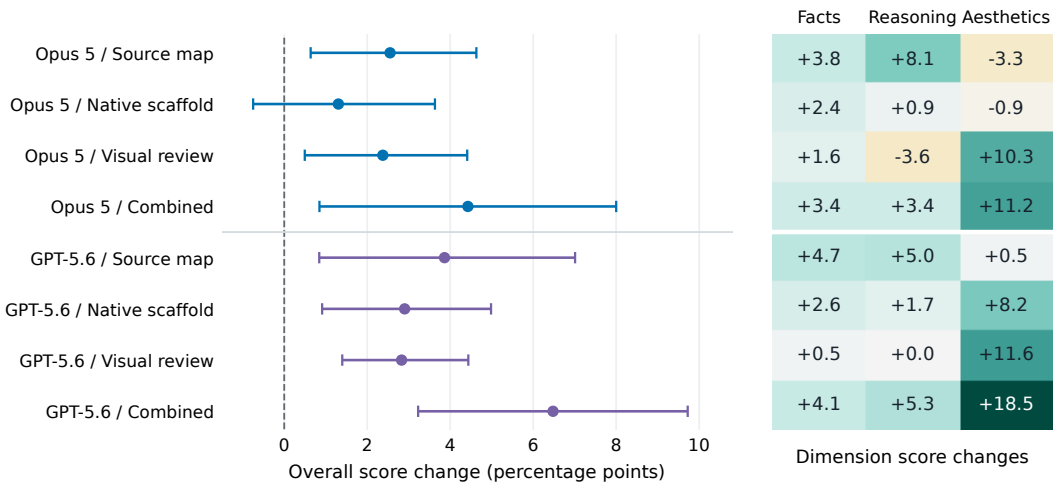


Figure 6: Effects of targeted support on 14 matched tasks per model. Left: change in overall score with 95% paired task-bootstrap intervals. Right: changes in factual, reasoning, and aesthetic scores (percentage points). Each support condition is compared with the same model's baseline; task resampling preserves paired outcomes.

Table 13: All assistance results on 14 tasks per model. Scores are percentages; Δs and its paired task-bootstrap 95% interval are percentage points. J counts simultaneous 50% attainment without a veto. Baseline rows use no additional support.

Model	Support	Score	F	R	A	J	Δs	95% interval
Opus 5	Baseline	48.0	46.5	51.3	57.9	3/14	+0.0	–
Opus 5	Source map	50.6	50.4	59.4	54.6	3/14	+2.6	[0.6, 4.6]
Opus 5	Native scaffold	49.3	48.9	52.2	57.0	3/14	+1.3	[-0.7, 3.6]
Opus 5	Visual review	50.4	48.1	47.7	68.3	3/14	+2.4	[0.5, 4.4]
Opus 5	Combined	52.4	49.9	54.7	69.2	2/14	+4.4	[0.8, 8.0]
GPT-5.6	Baseline	44.9	37.1	51.0	53.4	3/14	+0.0	–
GPT-5.6	Source map	48.8	41.8	56.0	53.9	3/14	+3.9	[0.8, 7.0]
GPT-5.6	Native scaffold	47.8	39.7	52.7	61.6	3/14	+2.9	[0.9, 5.0]
GPT-5.6	Visual review	47.8	37.6	51.0	65.0	3/14	+2.8	[1.4, 4.4]
GPT-5.6	Combined	51.4	41.2	56.3	71.9	3/14	+6.5	[3.2, 9.7]

These interventions do not isolate a unique cognitive faculty. They change practical support in an agent workflow. A null or uncertain effect can reflect the support's implementation, tasks, or model. The small purposive panel supports interpretations at this scope.

Table 14: Task-level variation in the assistance effect. Counts partition all 14 tasks for each row; unchanged includes outputs whose final score remains zero after a retained veto.

Model	Support	Improved	Unchanged	Worse
Claude Opus 5	Source map	8	4	2
Claude Opus 5	Native scaffold	7	4	3
Claude Opus 5	Visual review	9	4	1
Claude Opus 5	Combined	12	1	1
GPT-5.6-Sol	Source map	9	4	1
GPT-5.6-Sol	Native scaffold	7	5	2
GPT-5.6-Sol	Visual review	9	5	0
GPT-5.6-Sol	Combined	11	1	2