OMNI-COWORK: FROM COLLABORATIVE AUDIO-VISUAL DISCUSSIONS TO WORKSPACE EDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents can substantially support collaborative work by assisting with coding, document editing, and content creation. However, when teams discuss changes around shared artifacts, users must still consolidate their evolving decisions into explicit instructions before delegating edits to agents, adding effort and delaying execution. This raises a key question: can agents infer and implement a group's final decisions directly from audio-visual discussions into workspace edits? To explore it, we introduce **Omni-CoWork**, a benchmark for translating collaborative audio-visual discussions into verifiable workspace edits. It contains 124 tasks with human-recorded views and 1,213 synthetic tasks, with executable checks assessing both required changes and preservation constraints. Across seven agents, task success reaches at most 4.0% and 24.3% on the two splits, respectively, whereas substantial gains with gold instructions point to intent understanding as a major bottleneck. Investigating this bottleneck, we find that richer audio-visual descriptions alone remain insufficient: task success improves when agents ground discussion references in artifacts and revisit evidence during execution. These findings highlight the need for *grounded collective reasoning*: recovering final collective decisions, grounding them in artifacts, and verifying their realization as the workspace evolves. We translate these requirements into an Omni-CoWork skill that yields consistent gains on two agents but leaves most tasks unsolved, underscoring the need for more effective methods to bridge the gap to reliable practical use.

1 INTRODUCTION

Large language model (LLM) agents increasingly support coding (Wang et al., 2025a), document editing (Zhang et al., 2025), and content creation (Zheng et al., 2025), but still rely on explicit user instructions (Ma et al., 2025). In collaborative work, however, requirements often emerge through discussions around shared artifacts (Ju & Aral, 2025). Users must therefore translate evolving decisions into explicit requests specifying what to change and what to preserve before agents can act, adding manual effort and delaying execution.

Removing this manual step requires more than transcription: agents must distinguish final decisions from discarded proposals (da Silva & Gama, 2026) and ground discussion references in editable artifacts (Wang et al., 2025b), while identifying what must remain unchanged. This ability to recover actionable intent is not directly tested by benchmarks such as SWE-bench Pro and TheAgentCompany, which begin with explicit task descriptions (Deng et al., 2026; Xu et al., 2025) and thus bypass requirement recovery. This raises a key question: *can agents infer and implement a group's final decisions directly from collaborative audio-visual discussions into workspace edits?*

To investigate this question, we introduce **Omni-CoWork**, a benchmark for translating audio-visual discussions into workspace edits. Each task provides an initial workspace and a discussion recording, requiring the agent to infer and implement the final decisions without a separate textual edit specification. We instantiate this setting in 124 human-recorded and 1,213 synthetic tasks spanning code, documents, slides, spreadsheets, images, and media, with controlled variation in discussion complexity and editing requirements. Since correct execution must respect both what to change and what to preserve, we evaluate final workspaces using executable property checks, allowing multiple valid implementations rather than requiring an exact match to a reference 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.

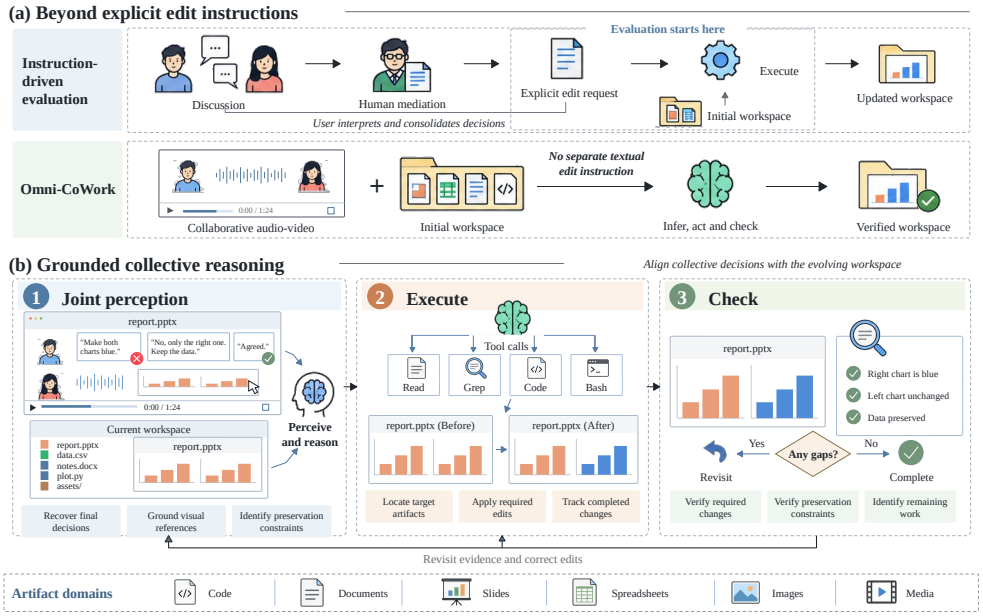


Figure 1: Overview of Omni-CoWork. (a) Agents infer and implement edits from collaborative audio-visual discussions and an initial workspace, without explicit edit instructions. (b) Agents recover and ground decisions, execute edits, and verify required changes and preservation constraints.

Across seven agents, task success reaches at most 4.0% on human-recorded tasks and 24.3% on synthetic tasks. Giving Claude gold instructions substantially improves success with the same executor, identifying intent understanding as a major bottleneck. We therefore examine how agents use discussion evidence, finding that visual and spoken inputs are complementary but that richer descriptions alone remain insufficient for reliable task completion. Beyond perceptual detail, agents benefit from grounding references in the workspace and revisiting them during execution, yet struggle to recognize incorrect implementations. Recovering a decision thus does not ensure that an agent can verify whether it has been correctly carried out.

Taken together, these findings motivate *grounded collective reasoning*: recovering the group's final decisions, grounding them in specific artifacts, and verifying their realization as the workspace evolves. To probe how far explicit guidance can address these requirements, we distill them into an Omni-CoWork skill and evaluate it on two agents. The skill improves task success on both splits, but most tasks remain unsolved, leaving a substantial gap to reliable practical use and motivating more effective methods for grounded collective reasoning.

Our contributions are threefold:

- We introduce Omni-CoWork, a benchmark for translating audio-visual discussions into workspace edits, with executable checks of required changes and preservation constraints.
- We evaluate seven agents, identifying intent understanding as a major bottleneck and diagnosing the roles of multimodal evidence, artifact grounding, and execution verification.
- We frame *grounded collective reasoning* as a research challenge and distill a skill, finding consistent gains but a substantial remaining gap to reliable practical use.

2 Related Work

2.1 Workspace Agents and Collaboration

Agent benchmarks commonly evaluate execution from explicit task descriptions (Liu et al., 2024). SWE-bench (Jimenez et al., 2024) tests repository-level issue resolution, while WebArena (Zhou et al., 2024), OSWorld (Xie et al., 2024), and WorkArena (Drouin et al., 2024) cover web, desktop,

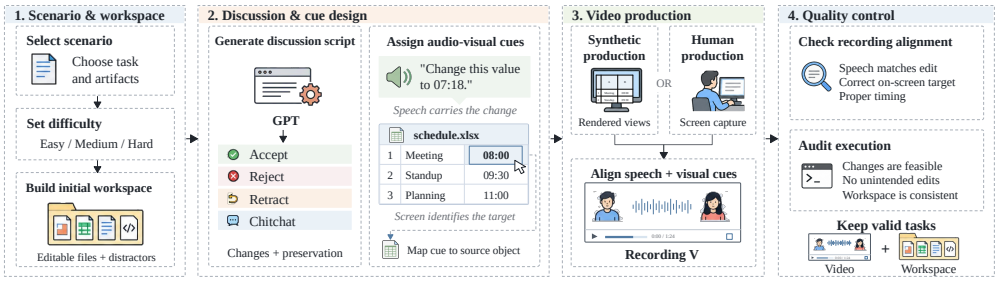


Figure 2: Omni-CoWork construction: (1) create workspaces, (2) script decisions and visual cues, (3) produce synthetic or human-recorded views, and (4) validate alignment and executability. Only the recording and initial workspace are agent-visible.

and enterprise workflows. Human–AI collaboration research studies interactive writing in CoAuthor (Lee et al., 2022) and shared task environments in Collaborative Gym (Shao et al., 2026). Omni-CoWork focuses on the step between discussion and execution: recovering the group's final decisions before acting. Its requirements are conveyed through an audio-visual discussion rather than a separate edit specification, so agents must distinguish accepted changes from discarded proposals and ground them in the workspace.

2.2 Audio-Visual Benchmarks

OmniVideoBench (Li et al., 2026), WorldSense (Hong et al., 2026), and Daily-Omni (Zhou et al., 2026) assess audio-visual understanding through question answering. Other benchmarks target cross-modal hallucination (Kim et al., 2025), event captioning, and temporal grounding (Geng et al., 2025). Their outputs describe or interpret recordings. Omni-CoWork instead evaluates the resulting workspace: understanding must lead to correct edits while preserving protected content. This outcome-based evaluation connects audio-visual interpretation to artifact grounding, execution, and verification, rather than treating perception as the final objective.

3 Omni-CoWork Benchmark

3.1 Task Formulation

As illustrated by the example in Figure 1, the agent receives a collaborative audio-visual recording and an initial workspace, and must infer and implement the group's final decisions without a separate textual edit specification. To determine both what was finally agreed and where it applies, the agent may revisit the recording, read and render artifacts, and edit files through multi-round tool interactions. The reference solution and verifier remain hidden throughout this process.

Task success requires the final workspace to satisfy both the required changes and preservation constraints specified by the group's final decisions. We assess these requirements with executable property checks, allowing valid implementations to differ from the reference workspace.

3.2 Data Construction

We construct each task from the workspace outward: create editable files, script a discussion about them, assign speech and visual cues, and produce the recording. The group's final decisions define both the reference solution and executable checks. As Figure 2 illustrates, this shared specification aligns the recording, editable artifacts, and evaluation criteria.

3.2.1 Scenario and Workspace Construction

Each scenario specifies artifact families and feasible operations, while a separate difficulty profile controls speaker count, discussion length, edit count, and distractor content. Table 1 summarizes

Table 1: Artifact families and discussion item types in Omni-CoWork. Appendix A provides details.

Artifact families		
Family	**Types**	**Representative operations**
Structured data	CSV, JSON, YAML, INI, ENV	Update, add, remove, normalize entries
Documents	Markdown, minutes	Revise text and structure; record decisions
Code	Python	Modify logic, insert guards, replace blocks
Web pages	HTML (CSS, JS)	Edit DOM elements, attributes, styles, and animations
Office files	DOCX, PPTX, XLSX	Edit text, slides, tables, formulas, and charts
Images	PNG	Adjust appearance, crop regions
Audio/video	MP4, M4A	Trim, concatenate, retime, edit audio and subtitles
Discussion item types		
Type	**Status**	**Required behavior**
ACCEPT	Accepted	Apply the edit
REJECT	Rejected	Do not apply the proposal
RETRACT	Withdrawn	Do not apply the withdrawn proposal
CHITCHAT	Non-task	No edit required

the artifact families, representative operations, and discussion item types. Based on these specifications, we instantiate an initial workspace with editable artifacts and distractor files, supporting either single-artifact tasks or coordinated changes across files. Creating the workspace first ensures that subsequent discussion references point to concrete, editable content.

3.2.2 DISCUSSION AND CUE DESIGN

Building on the initial workspace, we script discussions with accepted, rejected, and withdrawn proposals alongside non-task dialogue, corresponding respectively to the ACCEPT, REJECT, RETRACT, and CHITCHAT types in Table 1. Each proposal specifies a target and an intended modification, while final decisions determine which changes to apply and what to preserve. These decisions in turn define the reference workspace and the editing and preservation checks.

We then ground the scripted discussion in the workspace by linking each visual cue to its source object, such as a cell, text block, slide element, or media interval. For edits requiring both modalities, speech expresses the change while pointing or selection identifies the target; for example, "change this value to 07:18" accompanies a highlighted cell without naming its address.

3.2.3 VIDEO PRODUCTION

We combine synthetic rendering for scalable construction with human-operated screen recording for real application interfaces. In both pipelines, GPT-5.5 (OpenAI, 2026) generates scripted Chinese dialogue and Seed-TTS-2.0 (BytePlus, 2026) synthesizes multi-speaker speech, as detailed in Appendix B.4. Synthetic videos are generated using renderers, while human-recorded videos are captured by doctoral students operating real applications.

3.2.4 QUALITY CONTROL

We validate that the recording, workspace, and verifier consistently reflect the group's final decisions. Recording checks confirm that decisions are conveyed, visual cues identify the correct targets at the relevant moments, and private construction metadata is absent. We then require the reference solution to pass all checks and the initial workspace to fail required-change checks. Properties already satisfied initially are treated as preservation constraints, preventing unchanged content from counting as completed edits.

To further assess executability, we provide an agent with gold instructions specifying accepted edits and their locations. When execution fails, we inspect the instructions, workspace, verifier, and execution trace to determine whether it stems from a model error or a task defect. We then make targeted corrections to defective tasks or remove them.

Table 2: Dataset scale and coverage. Items include accepted, rejected, and withdrawn proposals and non-task dialogue. Detailed composition is in Appendix A.

	Total			Per task (avg.)		Coverage diversity		
Dataset	Tasks	Items	Video (h)	Items	Time (min)	Speakers (range)	File types	Op. types
Human-recorded	124	3,370	35.6	27.2	17.2	3–6	7	13
Synthetic	1,213	11,964	189.2	9.9	9.4	2–6	15	20

3.3 Dataset Statistics

The benchmark comprises 1,337 task instances across the seven artifact categories summarized in Table 1, including 124 human-recorded tasks and 1,213 synthetic tasks. The synthetic split spans six difficulty tiers, with shared concepts supporting comparisons across tiers. As shown in Table 2, synthetic tasks expand coverage from 7 to 15 file types and from 13 to 20 edit operations relative to the human-recorded split. Additional statistics on task composition, discussion duration, and accepted-change counts are provided in Appendix A.4.

3.4 Evaluation Metrics

Our primary metric, *Task success*, is the fraction of tasks whose final workspaces pass all required-change and preservation checks. To complement this all-or-nothing assessment, we also report *E-Rate* and *P-Rate*, the fractions of required edits and preservation constraints satisfied, respectively. An edit or constraint is satisfied only if all associated checks pass. We average these component rates across tasks with equal weight, excluding tasks without preservation constraints from P-Rate. Together, these metrics distinguish incomplete editing from damage to protected content, but neither alone captures whether a task has been completed successfully.

4 Experimental Results

4.1 Models and Environment

Model coverage. We evaluate the seven agents listed in Table 3: five LLM-based agents and two Omni-based agents with native audio-visual perception. This coverage allows comparison of complete agent configurations for decision recovery, artifact grounding, and edit execution.

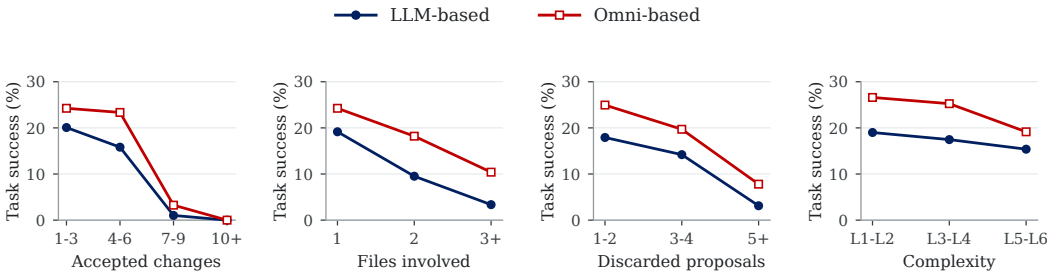
Agent framework. We implement our evaluation agents using `Qwen-Agent` to manage model–tool interactions. The framework connects the evaluated models to the workspace operations and audio-visual tools described below. Each task runs in a separate sandbox initialized with the workspace and recording. The shared initial prompt provides workspace and recording paths and a file inventory, without preloaded media, accepted changes, or target locations.

Tool setup. All models operate in a shared tool-enabled workspace environment. The tool suite includes `Bash`, `code`, `Edit`, `Glob`, `Grep`, `Read`, and `Write` for workspace operations. Audio-visual evidence is accessed through two tools. `read_media`, available only to Omni-based agents, places multimedia content directly in the evaluated model's context. `omni_av_tool`, available to all agents, uses a fixed auxiliary Gemini-3.5-Flash model to answer questions about the recording in text. LLM-based agents therefore obtain recording evidence through this auxiliary model rather than perceiving it directly. Appendix C details the tool interfaces and standard evaluation protocol.

Experimental setup. In the standard evaluation, agents have up to 100 turns per task, with a maximum of 16,000 tokens per generation. A run ends when the agent stops or reaches the turn limit, after which executable verifiers score the final workspace without an LLM judge. To diagnose the role of intent understanding, we additionally evaluate Claude with gold instructions specifying accepted edits and target locations, while keeping the executor unchanged.

Table 3: Agent performance (%). Task success requires all checks to pass; E-Rate and P-Rate measure required-change completion and preservation.

Model	Human-Recorded			Synthetic		
	Task success↑	E-Rate↑	P-Rate↑	Task success↑	E-Rate↑	P-Rate↑
LLM-based agents						
Claude Opus 4.8 (Anthropic, 2026)	0.0	43.3	97.2	19.4	54.5	94.1
DeepSeek V4 Pro (DeepSeek, 2026)	4.0	25.8	98.4	18.1	49.9	92.0
Grok 4.6 (xAI, 2026)	4.0	36.0	97.9	16.8	47.7	94.7
Qwen3.7 Max (Qwen, 2026)	0.0	25.3	89.8	8.2	33.6	87.3
Kimi K3 (Kimi, 2026)	1.6	14.4	95.2	9.7	33.9	94.9
Omni-based agents						
Gemini 3.8 Flash (Google, 2026)	4.0	53.0	93.1	24.3	60.1	94.3
Seed 2.1 Pro (ByteDance, 2026)	1.6	28.2	98.2	21.3	50.3	95.3
Gold-instruction reference						
Opus 4.8 + Gold Instructions	61.3	94.0	99.6	56.8	83.9	98.3



(a) Intent load (b) Files involved (c) Decision dynamics (d) Task complexity

Figure 3: Task success on synthetic tasks. Discarded proposals are rejected or withdrawn. Complexity bands average adjacent presets from Table 8.

4.2 Main Results

Table 3 shows that connecting an LLM executor to an Omni perception tool is far from sufficient for reliable collaborative editing. Despite access to audio-visual evidence and workspace operations, the five LLM-based agents achieve at most 4.0% task success on human-recorded tasks and 19.4% on synthetic tasks. The challenge is not confined to this configuration: even with native audio-visual perception, the best synthetic success reaches only 24.3%, while the human-recorded maximum remains 4.0%. These low success rates indicate a substantial gap between accessing multimodal evidence and reliably carrying out the group's intended changes.

Partial progress does not resolve this difficulty. On human-recorded tasks, Claude attains a 43.3% E-Rate and a 97.2% P-Rate but solves no task fully. Completing some edits and preserving most protected content thus falls short of satisfying the group's decisions and preservation constraints together. These results establish collaborative editing as a challenging task for current agents and motivate further analysis of its bottlenecks.

4.3 Performance Breakdown

Intent understanding is a major bottleneck. Table 3 shows that gold instructions raise Claude's task success from 0.0% to 61.3% on human-recorded tasks and from 19.4% to 56.8% on synthetic tasks with the same executor. This gap points to intent understanding as a major bottleneck: recovering final decisions, resolving their targets, and translating them into actionable requirements. The remaining failures also reveal substantial execution difficulty even with explicit requirements.

Intent load and cross-file scope. Consistent with this diagnosis, Figure 3a shows that, as accepted changes increase from 1–3 to 7–9, task success drops from 20.1% to 1.0% for LLM-based agents and from 24.3% to 3.3% for Omni-based agents, reaching 0.0% for both at 10+. This decline suggests a growing burden of understanding and tracking the group's intended changes. For cross-file grounding, Figure 3b shows that success falls from 19.2% to 3.4% and from 24.3% to 10.4%, respectively, as the number of files involved increases from one to three or more. This trend suggests greater difficulty in matching decisions to the correct artifacts and coordinating the intended edits across files. Although the all-checks criterion also becomes harder to satisfy as changes accumulate, these trends highlight a central challenge of intent understanding and grounding: determining what the group wants changed and in which file.

Decision dynamics and task complexity. Beyond intent load and file scope, Figure 3c shows that, as discarded proposals increase from 1–2 to 5+, task success drops from 17.9% to 3.1% for LLM-based agents and from 25.0% to 7.8% for Omni-based agents. More rejected or withdrawn proposals make it harder to distinguish final requirements from discarded alternatives. At the broader task level, Figure 3d shows that success falls from 19.0% to 15.4% and from 26.6% to 19.2%, respectively, from L1–L2 to L5–L6. As Table 8 shows, higher presets increase recording length, speaker count, and distractor files, while mean accepted-change counts remain within 4.6–4.9. This requires integrating evidence across speakers and time while locating relevant artifacts. Together, these trends reinforce the difficulty of recovering final intent and grounding it in the correct files.

5 Further Analysis

The preceding results reveal a substantial gap between executing explicit instructions and recovering intended edits from collaborative discussions. To understand the challenges underlying this gap, we investigate how agents integrate audio-visual evidence, ground collective decisions in workspace artifacts, and verify their realization during execution. We evaluate Claude-Opus-4.8 (Claude) and Gemini-3.8-Flash (Gemini) on 268 synthetic tasks previously solved fully by Claude with gold instructions, selecting tasks with demonstrated executability to reduce execution-related confounding.

Table 4: Perceptual access and evidence richness (%). (a) Task success (T-Rate), E-Rate, and P-Rate with upfront evidence. (b) Task success with agent-retrieved evidence. Gold-instruction scores in (b) are obtained from a separate evaluation on the selected analysis tasks.

(a) Input modality

Input	Claude			Gemini		
	T-Rate	E-Rate	P-Rate	T-Rate	E-Rate	P-Rate
Vision	2.6	14.2	98.2	2.2	9.1	99.0
Audio	6.0	33.1	95.3	0.7	2.8	96.6
AV	32.1	65.3	98.9	25.0	52.1	97.5

(b) Task success ↑

Evidence setting	Claude	Gemini
Frames	4.9	4.1
Frames + transcript	33.6	23.1
Frames + annotations	34.3	31.7
Gold Instructions	96.3	90.3

5.1 RQ1: Are Visual and Spoken Evidence Complementary?

Evidence about intended edits can be distributed across spoken discussion and visual context. We therefore compare Vision, Audio, and Audio-Visual inputs to assess their individual and joint contributions to intent recovery. The selected evidence is supplied upfront, rather than retrieved on demand, to reduce information loss from incomplete agent-driven inspection of the recording. Across conditions, agents retain workspace access but cannot access unselected channels.

Table 4a shows that joint input raises task success to 32.1% for Claude and 25.0% for Gemini, compared with 0.7–6.0% across single-channel conditions. E-Rate also rises to 65.3% and 52.1%, while P-Rate stays high across all conditions. These results suggest that recovering intended edits requires integrating spoken decisions with visual context, since either modality alone provides incomplete evidence. Yet most tasks remain unsolved with joint input, motivating further investigation into whether richer audio-visual evidence can improve intent understanding.

Finding 1. Visual and spoken evidence complement each other, improving task success and required-change completion, but joint access alone remains insufficient.

5.2 RQ2: Is Richer Audio-Visual Evidence Sufficient?

We next examine whether more explicit discussion evidence improves task completion. Agents retain the same workspace tools and retrieve evidence on demand: frames, frames with transcripts, or frames with structured discussion annotations that include speaker information and decision status. Table 4b shows that adding transcripts raises task success from 4.9% to 33.6% for Claude and from 4.1% to 23.1% for Gemini. Replacing transcripts with annotations provides a further 0.7 and 8.6 percentage points, reaching 34.3% and 31.7%, respectively. Despite these improvements, both agents remain far below their gold-instruction performance of 96.3% and 90.3%. These results suggest that simply enriching audio-visual descriptions does not bridge the gap between discussion evidence and actionable workspace edits.

Finding 2. Richer audio-visual descriptions improve task success, but greater perceptual detail alone remains insufficient for reliable completion, motivating stronger grounding of discussion decisions in workspace artifacts.

5.3 RQ3: Does Joint AV–Workspace Perception Improve Grounding?

Richer descriptions still require grounding in concrete artifacts, prompting us to test joint perception of the recording and workspace. To isolate the effect of perceptual input, we use a fixed auxiliary perceiver, Gemini-3.5-Flash, to generate a single plan from the recording, workspace, or both. Claude and Gemini then execute the plan with the same workspace access and tools across conditions, without recording access or further perception calls. This separates the effect of the perceiver's input from differences in workspace access or repeated consultation during execution.

Figure 4 shows that joint input raises success from 20.9% to 31.0% for Claude and from 19.0% to 29.1% for Gemini relative to recording-only plans, a gain of 10.1 percentage points for each. Workspace-only plans achieve 0.0% for both. Thus, discussion evidence and workspace context play complementary roles: the former conveys the group's intended changes, while the latter helps connect them to concrete editing targets. Joint perception thus improves task completion under fixed-plan execution, highlighting the value of connecting collective decisions to artifacts during planning.

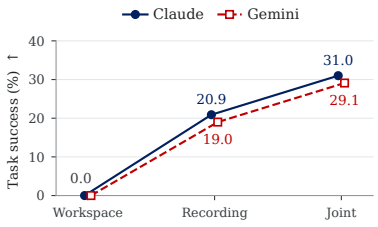


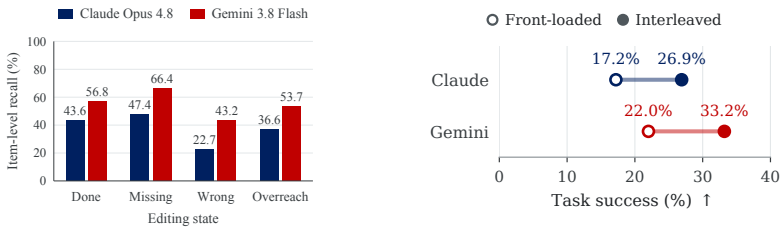
Figure 4: Task success with fixed plans.

Finding 3. Workspace context benefits not only execution but also intent understanding: grounding discussion evidence in artifacts during planning improves task completion.

5.4 RQ4: Does Continued Grounding Improve Task Completion?

Beyond identifying what to change, agents must track whether the intended edits are correctly applied as the workspace evolves. We first examine their ability to recognize the editing state, then test whether continued joint perception of the discussion and workspace improves task completion.

Recognizing the editing state. Given partially edited workspaces, agents classify edits as Done (correct), Missing (not applied), Wrong (incorrect), or Overreach (changes to files not mentioned) by reconciling discussion decisions with artifact contents and metadata. Figure 5a shows a consistent gap between detecting Missing edits and recognizing Wrong ones: recall falls from 47.4% to 22.7% for Claude and from 66.4% to 43.2% for Gemini. This gap highlights the difficulty of judging whether an applied edit matches the collective decision, motivating continued comparison between the intended changes and the evolving workspace.



(a) Editing-state recognition (b) Interleaved perception

Figure 5: Grounding during execution on the synthetic analysis pool.

Finding 4a. Both agents detect missing edits more reliably than incorrect ones, highlighting the difficulty of verifying whether actual workspace changes match collective decisions.

Consulting perception during execution. Motivated by the difficulty of recognizing incorrect edits, we test whether continued joint perception improves end-to-end task completion by comparing front-loaded and interleaved consultation. Both conditions use the auxiliary perceiver from RQ3 to inspect the recording and workspace, with up to six consultations allowed. FRONT-LOADED access ends at the first workspace-tool call, including read-only calls; INTERLEAVED access remains available throughout execution. Figure 5b shows that interleaving raises task success from 17.2% to 26.9% for Claude and from 22.0% to 33.2% for Gemini. Continued joint perception thus improves completion even when the allowed number of consultations is unchanged.

Finding 4b. Interleaving joint perception with execution improves task success under matched consultation limits, supporting iterative grounding against the workspace as editing proceeds.

SUMMARY: GROUNDED COLLECTIVE REASONING

These analyses suggest that reliable collaborative editing requires *grounded collective reasoning*, not merely detailed perception: agents must recover final collective decisions, ground them in editable artifacts, and verify their realization as the workspace evolves. Integrating these capabilities remains challenging on Omni-CoWork, motivating methods that connect collective intent with execution.

5.5 OMNI-COWORK SKILL

We distill these findings into an Omni-CoWork skill: prompt guidance to recover final decisions from audio-visual evidence, ground them in exact file locations and values, and verify on-disk edits while preserving unrelated content. Figure 6 shows task success gains of 4.1 and 1.6 percentage points on human-recorded tasks and 3.6 and 2.4 on synthetic tasks for Gemini and Seed, respectively. These consistent gains leave most tasks unsolved and a substantial gap to reliable practical use, motivating more effective methods for grounded collective reasoning.

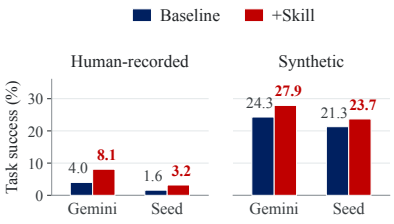


Figure 6: Task success with the skill.

6 CONCLUSIONS

We introduce Omni-CoWork, a benchmark for translating collaborative audio-visual discussions into verifiable workspace edits. Low success across seven agents and the gold-instruction gap identify intent recovery as a major bottleneck. Further analyses show that richer descriptions leave this gap unresolved, whereas artifact grounding and continued perception improve task completion. A skill distilled from these findings yields consistent gains but leaves most tasks unsolved and performance far from practical reliability. These results motivate more effective methods for grounded collective reasoning to bridge the gap between collective intent and reliable workspace execution.

AI USE STATEMENT

We used generative AI for benchmark data construction, analysis assistance, and language polishing. Language models assisted in generating scenario concepts, discussion scripts, initial and reference workspaces, and executable verifiers, as well as in media inspection and task repair. Text-to-speech models produced the dialogue tracks used in both recording pipelines. The model roles, construction procedures, and selected prompts are documented in Sections B and D.2. All AI-assisted text, analyses, and benchmark artifacts underwent manual review. Benchmark scores are computed by executable artifact checks rather than an LLM judge.

ETHICS STATEMENT

All task scenarios, workspace contents, and discussion scripts in Omni-CoWork are AI-generated and fictional; they do not contain real user data or private workplace information. The human screen recordings were produced with the participation of doctoral students, who provided informed consent and received no compensation.

REPRODUCIBILITY STATEMENT

To support reproducibility, we document benchmark construction and evaluation in the main text and Appendices A–D, including agent configurations, prompts, and the complete skill and its injection protocol. We plan to publicly release the benchmark dataset for independent evaluation.

ACKNOWLEDGMENTS

We thank the anonymous reviewers, area chairs, and program chairs for their time and effort in evaluating this work and coordinating the review process.

REFERENCES

Anthropic. Claude Opus 4.8 system card. System card, Anthropic, 2026. URL https://www.anthropic.com/claude-opus-4-8-system-card.

ByteDance. Seed2.1 model card: Agentic intelligence for productivity. Model card, ByteDance, 2026. URL https://lf3-static.bytednsdoc.com/obj/eden-cn/lapzild-tss/ljhwZthlaukjlkulzlp/seed2.1/Seed2_1_Model_Card.pdf.

BytePlus. TTS 2.0. Seed Speech documentation, July 2026. URL https://docs.byteplus.com/en/docs/byteplusvoice/texttospeechv2. Updated July 29, 2026.

Lucas Daniel Costa da Silva and Kiev Gama. GADR: Gathering architecture decision records from meeting transcriptions, 2026. URL https://arxiv.org/abs/2608.17694. Preprint.

DeepSeek. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348. Technical report.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Chetan Rane, Karmini Sampath, Maya Krishnan, Srivatsa Kundurthy, Sean Hendryx, Zifan Wang, Chen Bo Calvin Zhang, Noah Jacobson, Bing Liu, and Brad Kenstler. SWE-Bench Pro: Can AI agents solve long-horizon software engineering tasks? In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/61047.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11642–11662. PMLR, 2024. URL https://proceedings.mlr.press/v235/drouin24a.html.

Tiantian Geng, Jinrui Zhang, Qingni Wang, Teng Wang, Jinming Duan, and Feng Zheng. LongVALE: Vision-audio-language-event benchmark towards time-aware omni-modal perception of long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Geng_LongVALE_Vision-Audio-Language-Event_Benchmark_Towards_Time-Aware_Omni-Modal_Perception_of_Long_Videos_CVPR_2025_paper.html.

Google. Gemini 3.8 Flash: Model card. Model card, Google, September 2026. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-8-Flash-Model-Card.pdf.

Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. WorldSense: Evaluating real-world omnimodal understanding for multimodal LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=YxsfxAvJv4.

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.

Harang Ju and Sinan Aral. Collaborating with AI agents: Field experiments on teamwork, productivity, and performance, 2025. URL https://arxiv.org/abs/2503.18238. Revised in 2026.

Sung-Bin Kim, Hyun-Bin Oh, JungMok Lee, Arda Senocak, Joon Son Chung, and Tae-Hyun Oh. AVHBench: A cross-modal hallucination benchmark for audio-visual large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jTEKTdI3K9.

Kimi. Kimi K3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653. Technical report.

Mina Lee, Percy Liang, and Qian Yang. CoAuthor: Designing a human-AI collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2022. doi: 10.1145/3491102.3502030. URL https://doi.org/10.1145/3491102.3502030.

Caorui Li, Yu Chen, Yiyan Ji, Jin Xu, Zhenyu Cui, Shihao Li, Yuanxing Zhang, Zhenghao Song, Dingling Zhang, Ying He, et al. Omnivideobench: Towards audio-visual understanding evaluation for omni mllms. In *International Conference on Learning Representations*, volume 2026, pp. 138214–138236, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Qianou Ma, Weirui Peng, Chenyang Yang, Hua Shen, Ken Koedinger, and Tongshuang Wu. What should we engineer in prompts? training humans in requirement-driven LLM use. *ACM Transactions on Computer-Human Interaction*, 32(4):1–27, 2025. doi: 10.1145/3731756. URL https://doi.org/10.1145/3731756.

OpenAI. Introducing GPT-5.5. OpenAI, April 2026. URL https://openai.com/index/introducing-gpt-5-5/.

Qwen. Qwen3.7: The agent frontier. Qwen Blog, May 2026. URL https://qwen.ai/blog?id=qwen3.7.

Yijia Shao, Vinay Samuel, Yucheng Jiang, John Yang, and Diyi Yang. Collaborative Gym: A framework for enabling and evaluating human-agent collaboration. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a1c8a68e52499c9396854e3f967e37c0-Abstract-Conference.html.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Daniel Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An open platform for AI software developers as generalist agents. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a4b6ad6b48850c0c331d1259fc66a69c-Abstract-Conference.html.

Zhengxiang Wang, Weiling Li, Panagiotis Kaliosis, Owen Rambow, and Susan Brennan. LVLMs are bad at overhearing human referential communication. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16758–16782. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.emnlp-main.849. URL https://aclanthology.org/2025.emnlp-main.849/.

xAI. Model card: Grok 4.6. Model card, xAI, 2026. URL https://media.x.ai/v1/website/card-4p6-4cd2dc57.pdf.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Frank (Fangzheng) Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0315. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0d744742f6fac4d1134c019b7cef3c8a-Abstract-Datasets_and_Benchmarks_Track.html.

Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. UFO: A UI-focused agent for windows OS interaction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 597–622. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.26. URL https://aclanthology.org/2025.naacl-long.26/.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. PPTAgent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14402–14418. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.728. URL https://aclanthology.org/2025.emnlp-main.728/.

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.

Ziwei Zhou, Rui Wang, Zuxuan Wu, and Yu-Gang Jiang. Daily-Omni: Towards audio-visual reasoning with temporal alignment across modalities, 2026. URL https://arxiv.org/abs/2505.17862v2.

A Dataset Details

We provide additional details on the 1,213 synthetic tasks and 124 human-recorded tasks introduced in Section 3, including scenario coverage, change types, and task composition. Throughout, we distinguish all discussion items from accepted changes that the agent must carry out.

A.1 Concrete Task Example

The agent receives `query_clip.mp4` with its audio track and an initial workspace. Figure 7 shows a percentage-formatting example from an L6 synthetic anti-doping review task, including the proposal at 11:15 and the later final decision.



Figure 7: Perceptual inputs for a concrete task: a full video frame at 11:15, speech from the same discussion, and an abbreviated view of the 50-file initial workspace. The first utterance accompanies the shown frame; the later excerpt gives the final decision, with intervening turns omitted.

Figure 8 shows the visual cues that locate the edit. The path rendered in the video identifies the workspace file; the worksheet tab and highlighted range specify where the spoken reference to *this column* applies.



Figure 8: Localizing the instruction in Figure 7. Numbered panels highlight (1) the video path and its corresponding workspace file, (2) the selected worksheet, and (3) the selected range `G2:G9`; panel (4) describes the formatting operation.

A.2 Scenario Coverage

Scenario domains describe the professional setting, whereas artifact families describe the files being edited. A scenario may involve several artifact families, and the same family can appear across domains. Table 5 compares the two splits across 13 professional sectors. Synthetic multi-subtask tasks may span several sectors and are counted in each. Each human-recorded meeting belongs to one sector, and the 124 meetings cover 44 fine-grained domains.

Table 5: Professional-sector coverage. Synthetic tasks may span multiple sectors; each human-recorded meeting belongs to one sector.

Sector	Synthetic	Human	Example domains
Finance and insurance	169	7	Asset securitization; budget review
Manufacturing and industry	155	8	PCB assembly; production-line takt
Retail, food, and commerce	154	35	Menu engineering; store display
Healthcare and medicine	144	9	Clinical trials; outpatient appointments
Transport and logistics	128	13	Cold-chain control; fleet fuel management
Education	106	11	Academic competitions; summer-camp intake
Construction and property	104	9	Elevator retrofit; campus access control
Culture, sports, and tourism	96	5	Exhibition layout; community fitness
Government and public services	87	3	Paper-archive digitization
Energy and utilities	86	6	Remote meter reading
Agriculture and environment	72	7	Agricultural procurement; scrap recycling
HR and corporate operations	58	5	Meal-subsidy schemes; supplier review
IT, data, and telecommunications	42	6	Website accessibility; mobile payment

Table 6: Change categories among 6,009 annotated accepted changes in the synthetic split. Percentages are rounded.

Category	Changes	Share (%)	Example actions
Content	1,602	26.7	Set a value; rewrite, add, or remove text
Data	1,262	21.0	Set a value; export, transform, or filter data
Style	1,134	18.9	Add interaction; reformat; recolor
Parameter	739	12.3	Set a parameter value
Media	383	6.4	Trim; concatenate; retime; edit audio or subtitles
Structure	317	5.3	Export; change a chart; add or remove elements
Behavior	303	5.0	Set a behavior-related value
Cross-file	269	4.5	Copy information from a source file
Total	6,009	100.0	

A.3 Discussion Items and Accepted Changes

Each *discussion item* is a proposal or non-task utterance labeled ACCEPT, REJECT, RETRACT, or CHITCHAT. Only proposals labeled ACCEPT define required changes; rejected or withdrawn proposals must not be executed, and non-task dialogue adds no editing requirement. We refer to these required modifications as *accepted changes*, distinguishing them from all discussion items and from the individual tool calls used to implement them. Preservation constraints protect unrelated content and workspace structure. We categorize accepted changes by the aspect of an artifact being changed (*change category*) and the specific modification (*action*).

Synthetic change distribution. Table 6 summarizes the 6,009 synthetic accepted changes with category annotations; this category-level subset is distinct from the full discussion-item totals. Content (26.7%), data (21.0%), and style (18.9%) are the largest categories, covering text revision, data transformation, and visual formatting. The remaining categories cover parameter, media, structural, and behavioral changes. Cross-file edits (4.5%) require consulting a source file to determine the change to another artifact.

Human-recorded change distribution. The 124 human-recorded meetings contain 1,486 accepted changes among 3,370 discussion items (44.1%). Table 7 reports their distribution, including 23 accepted changes without a category label. Content, media, and parameter changes predominate. Media editing accounts for 31.7% of accepted changes, compared with 6.4% in the annotated synthetic subset, and comprises 264 trims, 88 speed changes, 78 concatenations, and 41 audio edits.

Table 7: Change categories among the 1,486 accepted changes in the human-recorded split, including 23 without a category label. Percentages are rounded.

Category	Changes	Share (%)	Example actions
Content	499	33.6	Modify content
Media	471	31.7	Trim; retime; concatenate; edit audio
Parameter	289	19.4	Change a parameter
Style	184	12.4	Recolor
Data	20	1.3	Modify data
Unlabeled	23	1.5	No category label
Total	1,486	100.0	

Table 8: Synthetic-task composition. Duration, Items, and Changes are per-task means. Distr. denotes distractor files. The single-focus dialogue presets are flat for L1, negotiated for L2, and contested for L3–L6. Multi-subtask presets use contested dialogue. Means are rounded.

Preset	**Min.**	**Speakers**	**Distr.**	**Items**	**Changes**	**Tasks**
			Single-focus tasks			
L1	1.4	2	4	8.4	4.6	182
L2	2.8	3	12	8.3	4.6	187
L3	4.4	3	12	8.4	4.6	192
L4	8.6	4	24	8.6	4.7	179
L5	15.7	4	32	8.5	4.9	184
L6	23.5	5	48	8.7	4.8	158
			Multi-subtask tasks			
X2	8.8	4	24	14.6	7.6	64
X3	13.1	4	32	23.2	11.7	37
X4	17.2	4	32	30.7	16.0	27
X8	25.6	5	48	65.0	32.0	3
Total						**1,213**

A.4 Difficulty Dimensions and Task Composition

Synthetic task composition. Synthetic tasks vary in speaker count, dialogue complexity, distractor files, recording duration, discussion-item count, document count, decision mix, source location, and intent abstraction.

Dialogue ranges from direct decisions (*flat*), through proposals followed by resolutions (*negotiated*), to multi-round exchanges with reversals (*contested*). Source-location labels distinguish values stated in speech (S1), values retrieved from another workspace file (S2), and requirements that combine the recording with a document (S3). Intent-abstraction labels distinguish direct requests (L1), descriptions of a problem (L2), and commentary that must be translated into an edit (L3). These abstraction labels are distinct from the L1–L6 task presets below.

Table 8 summarizes 1,082 single-focus tasks across six presets and 131 multi-subtask tasks involving two, three, four, or eight documents. It reports mean recording durations and item counts. "Items" counts all discussion items, whereas "Changes" counts only accepted changes.

Within the single-focus presets, mean accepted-change counts remain between 4.6 and 4.9, while duration, speaker count, distractors, and dialogue complexity vary. The multi-subtask presets additionally increase document and editing load.

Human-recorded task composition. Human-recorded tasks comprise multi-subtask meetings rather than the synthetic L/X presets. Each meeting contains 3–6 subtasks (mean 4.0), involves 3–6

speaker roles, and includes an average of 27.2 discussion items. Recordings average 17.2 minutes (median 14.0; range 11.1–42.2). Accepted decisions account for 44.1% of discussion items, compared with 54.4% in the synthetic split. The remaining human-recorded items are rejected proposals (21.7%), non-task dialogue (21.9%), or withdrawals (12.3%).

B Construction Details

The construction pipeline in Section 3.2 uses a shared task specification to align the discussion, reference workspace, and executable checks with the same intended edits. Selected construction prompts are provided in Section D.2.

B.1 Scenario and Workspace Construction

Each task begins with a professional scenario specifying the artifact types, file inventory, and proposed changes. We instantiate the scenario as an initial workspace containing editable artifacts and distractor files. The task brief describes the workspace without revealing the intended edits, and the initial files contain no comments or placeholders that disclose the solution.

A private specification associates each proposal with its decision status, operation, target file, intended outcome, and level of intent abstraction. Target locators, such as text anchors, code symbols, cell coordinates, and interface elements, connect the editable content to the visual cues in the recording. Applying only the accepted changes yields the reference workspace, with unrelated content left unchanged.

Executable checks assess the requested artifact properties rather than requiring an exact match to the reference workspace. For structured data, they inspect parsed properties or query results; for protected content, they compare against the initial workspace. This allows different implementations of the same intended change.

B.2 Discussion Generation

We turn the task specification into multi-speaker Chinese dialogue with domain-appropriate participant roles and conversational exchanges. Accepted proposals, rejections, withdrawals, and non-task utterances are interleaved, so that the agent must track how decisions develop over the meeting. Withdrawals include both an initial proposal and a subsequent retraction. Brief opening and closing exchanges frame the discussion without summarizing all decisions or repeating the target values.

Dialogue turns are annotated with speaker identities, discussion-item links, decision statuses, and pauses. These private annotations guide speech synthesis and the timing of visual cues. References to visual actions are restricted to those supported by the corresponding recording pipeline.

B.3 Audio-Visual Cue Design

We separate what the dialogue conveys from what the agent must recover visually. Speech communicates the requested operation and new values, while visual cues identify the target. Visible identifiers, such as cell coordinates, text anchors, filenames, and timeline positions, are reserved for the visual channel rather than read aloud. The dialogue instead uses references such as *this cell* or *this part*, grounded in the corresponding visual cue.

For cross-file edits, speech identifies the source document and the relevant portion to consult, but the agent must retrieve the value or formula from that document. For temporal media edits, the visual cue identifies the interval, while speech specifies the operation and any new parameter that is not visible on screen.

B.4 Synthetic Recording and Synchronization

Synthetic videos are rendered at 1280×720 pixels and 12 frames per second. A PIL-based renderer produces editor, table, preview, image, and logic views. Office and HTML artifacts are laid out with WeasyPrint and rasterized with PyMuPDF before composition. The cursor indicates where an edit should be made, while the dialogue conveys the requested change.



Figure 9: Operator-side recording console, shown in bilingual form. It provides the application window, target, visual action, dialogue, and segment timing. The console and its private annotations are excluded from the recording presented to the agent.

Speech is synthesized utterance by utterance as 24 kHz MP3 audio, using voice-pool cloning or Seed-TTS 2.0 with a consistent voice assignment for each speaker. We use `gpt-5.5` for scenario, dialogue, and workspace generation, and `gemini-3.5-flash` for construction-time media inspection. Repair stages use `gpt-6-astra` and `gpt-5.5`.

The duration of each synthesized utterance determines its interval on a shared audio-video timeline. We insert a base gap of 0.5 seconds between utterances and 1.2 seconds for annotated long pauses, with additional pauses for speaker changes and non-task exchanges. Video frames and the assembled audio track use the same intervals, making synchronization part of the rendering process.

B.5 Human Recording Protocol

Recording setup. Operators follow the scripted dialogue while interacting with real applications. Figure 9 shows the operator console on one display, which provides the application window, target, visual action, dialogue, and segment timing. Only the separate display containing the application windows is recorded, and dialogue is monitored through headphones. Application windows are maximized to retain file and interface context. Recordings are typically captured at widths of 1280–1920 pixels and scaled down toward 1280 pixels without upsampling.

Visual interactions. Each segment contains one action in one application window, allowing unsuccessful takes to be re-recorded independently. Operators may select cells or text, point to image regions, scroll, and play or pause media, but leave the workspace unchanged. Workspace files are marked read-only. During references such as *here*, operators hold the native system cursor on the target; window changes occur at segment boundaries.

Assembly and alignment. In the primary browser-based workflow, paired start and end signals align each recorded segment with the dialogue. These offsets are cross-checked against the

recording-page timestamps, with discrepancies above 150 ms flagged for inspection. Segments are then trimmed and concatenated, and any captured audio is replaced with the clean synthesized dialogue track.

Target calibration and quality checks. Before recording, operators can calibrate image regions or media intervals to match the displayed targets. Corrections are propagated to the task specification, verifier, reference workspace, and gold instructions. Recording checks cover technical validity, cursor behavior, and visible task metadata. Pointer-placement and holding issues are flagged for inspection, while recordings exposing task identifiers are rejected. The script and operator console are excluded from the captured view.

B.6 Validation and Information Separation

The reference workspace must pass all checks, while the unchanged initial workspace must fail the required-change checks. Properties already satisfied in the initial workspace, such as preserved structure or an unchanged file inventory, are evaluated as preservation constraints rather than completed changes.

Additional checks assess target locators, the scope of reference changes, and the separation of spoken information from visual identifiers. The recording-alignment and execution audits in Section 3.2.4 assess whether the intended decisions are recoverable and executable. Agents receive the initial workspace and recording; private task specifications, reference workspaces, and verifier code are withheld.

C Experimental Details

C.1 Standard Evaluation Protocol

Each task runs in a separate `e2b` sandbox containing the initial workspace and meeting recording. A shared prompt gives the working directory, file inventory, and recording path, and asks the agent to implement the team's final decisions. It does not reveal the accepted changes, target locations, or reference values.

As described in Section 4.1, agents have up to 100 turns to inspect the available evidence, execute code, and modify files. Each generation is limited to 16,000 tokens. A run ends when the agent stops or reaches the turn limit. Agents are instructed to store intermediate media extracts outside the project directory to avoid unintended workspace changes.

C.2 Workspace and Perception Tools

Table 9 summarizes the evaluation tools. Workspace operations run inside the sandbox, while media-perception requests are handled by the host.

C.3 Artifact Verification

Each task has an executable Python verifier; multi-subtask tasks aggregate their constituent verifiers. Scoring does not use an LLM judge. Required-change checks evaluate the requested artifact properties, and preservation checks compare protected content with the initial workspace. In particular, unrelated files are protected against byte-level modification.

We use the metrics defined in Section 3.4. An accepted change or preservation constraint is satisfied only when all checks associated with it pass. *Task success* requires all required-change and preservation checks to pass. *E-Rate* and *P-Rate* are computed per task and then averaged across tasks; P-Rate excludes tasks with no preservation constraints. This distinguishes completing some edits from delivering a fully correct workspace.

Table 9: Workspace and media-perception tools in the evaluation environment.

Tool	Interface and behavior
`Bash`	Executes a shell command; its timeout defaults to 300 seconds and is capped at 600 seconds.
`Edit`	Replaces an exact string in a file. A match must be unique unless replacement of all matches is requested.
`Glob`	Finds files matching a path pattern.
`Grep`	Searches file contents using a regular expression.
`Read`	Returns numbered text-file lines, optionally restricted by an offset and line count.
`Write`	Writes file contents, creating parent directories and replacing an existing file when necessary.
`code`	Runs Python in a fresh process with the workspace as its working directory; files persist between calls.
`read_media`	Accepts a media path, directory, or glob and returns evidence according to the perceptual profile.
`omni_av_tool`	Accepts a media path and a question and returns the auxiliary model's answer as text.

C.4 Omni-CoWork Skill

The following is the complete skill text used for Gemini and Seed in Section 5.5. We inject it directly into each agent's prompt, keeping tool access and resource budgets identical to the corresponding baseline. Its three steps cover final-decision recovery (INTENT), artifact grounding (GROUND), and edit verification (REALIZE).

Prompt C.1. Omni-CoWork Skill

`## How to work this task`

Three things decide success:

1. INTENT — the change to make lives in the recording's PICTURE (cursor / red dot showing which file / line / element) AND its AUDIO (the target value). Use both. Capture every decision that is the team's FINAL word; ignore anything overruled or withdrawn. Missing a requested change is the main way to fail, so be exhaustive.
2. GROUND — open the real files; pin each decision to an exact file, location, and new value. If you cannot pin it, look again — do not guess.
3. REALIZE — make the edits on disk (not just in your reply), then re-check each one: right value, nothing missed, nothing unauthorized changed.

Your output is the files themselves. Turns are limited — keep moving.

D Prompt Templates

D.1 Standard Evaluation Prompts

The standard edit-mode system prompt combines a common instruction with an LLM- or Omni-specific capability block. Both model types receive the same initial user-message template. These templates describe the standard evaluation; the diagnostic protocols in Section 5 explicitly modify the available evidence or interaction rules.

Prompt D.1. Common System Instruction

You are working inside a sandbox on a small project. Use the tools you are given. Be precise: change only what you were asked to change, and leave everything else exactly as it is.

Prompt D.2. LLM Capability Block

`## Your senses in this run`

- You CANNOT see the recording, and there is no route to its picture in this run.
- You CANNOT hear. You have no ability to process audio at all.
- The recording is a real file in this run (the message tells you where). Reading its bytes with code tells you nothing about what is in it; your tools do:
  - `omni_av_tool` answers a question about it, as text.

  Use them freely — a call costs you nothing but a turn.
- Keep anything you extract outside the project directory (e.g. under `/tmp`); the project files are the only thing you should be editing.

Prompt D.3. Omni Capability Block

`## Your senses in this run`

- You CAN see: pictures of the recording can be put into your context, and you perceive them yourself — but none are there yet; you have to go get them.
- You CAN hear: the audio can be put into your context and you perceive it yourself — but none is there yet; you have to go get it.
- The recording is a real file in this run (the message tells you where). Reading its bytes with code tells you nothing about what is in it; your tools do:
  - `read_media` puts the video itself into your own context, so you perceive it.
  - `omni_av_tool` answers a question about it, as text.

  Use them freely — a call costs you nothing but a turn.
- Keep anything you extract outside the project directory (e.g. under `/tmp`); the project files are the only thing you should be editing.

Prompt D.4. Initial User Message

Your working directory is {`workdir`} and has these files: {`files`}. The meeting recording is a file at {`recording_path`} — use your tools to read it. Based on what the team asks for in the meeting, edit the files in place (read a file, then write the modified file back). Change only what the team decides to change and leave everything else exactly as it is. When you are done, stop — you do not need to print the files.

D.2 Selected Construction Prompts

The following blocks present selected construction-prompt excerpts in English. Template variables are filled with task-specific values during construction; the dialogue-generation prompt requests Chinese dialogue.

Prompt D.5. Concept Design

System prompt.

You are a senior benchmark editor. Produce diverse, realistic, and verifiable task concepts. Output only a strict JSON array, without explanations or Markdown fences.

Task-template excerpt.

Design a batch of {`n`} concepts for {`label`} tasks in Omni-CoWork. Each task will provide a workspace containing real files and distractors, together with a multi-person review recording in which participants discuss and point to the requested changes. The evaluated model will edit the workspace based on the recording, and the result will be checked by code for both required changes and preservation.

Prompt D.5. Concept Design (continued)

Output a strict JSON array. Each concept specifies an identifier, artifact family, mode, concrete professional domain, brief, file inventory, and a sketch of discussion items. The brief describes the files and the kind of changes under discussion in one or two sentences, without revealing specific values or answers.

Use distinct domains across the batch and cover multiple file types. Mix direct requirements, symptoms, and commentary, using the L1, L2, and L3 abstraction labels. Each discussion sketch must contain at least two accepted proposals, one rejected proposal, one withdrawn proposal, and one non-task item. A withdrawal must represent a genuine change of mind.

Do not reveal answers in the brief or file descriptions. Store the final intended changes only in the private intent field.

Prompt D.6. Workspace and Verifier Materialization

System prompt.

You are a senior benchmark task engineer. Output only a strict JSON object, without explanations or Markdown code fences.

Task-template excerpt.

Instantiate the supplied task concept as an executable task. The meeting recording itself is the instruction: the evaluated model must infer the final decisions, locate the corresponding files, apply the accepted changes, and deliver the workspace. Scoring must use code to verify both that required changes were made and that protected content was preserved.

Return the initial workspace files, discussion items, reference files after accepted changes, and verifier source. Initial files must be realistic, self-consistent, and neutral, without comments, TODOs, or placeholders that hint at the requested answer. Include at least one distractor file.

Only accepted proposals may alter the reference files. Relative to the initial workspace, change only accepted targets while preserving unrelated content and required structure. When a format requires re-serialization or re-encoding, verify preservation of the relevant content and properties rather than byte identity. Provide target locators grounded in the actual initial contents.

At least 60% of accepted targets must lie within a naturally occurring group of at least three similar items, so that visual pointing determines which item is meant. Do not insert artificial duplicates merely to create ambiguity.

Make verification implementation-independent. For data operations, parse files and check properties or result sets rather than exact reference text. A required-change check must fail on the initial workspace and pass on a correct result. Properties that already hold in the initial workspace, including preserved structure and file inventory, are preservation checks, not required changes. Preservation checks compare the output with the initial workspace.

Prompt D.7. Allocation Between Speech and Visual Evidence

System prompt.

You are an information-channel auditor for evaluation data. Output only strict JSON, without explanations or Markdown fences.

Task-template excerpt.

The following is the private specification of a task in which a video serves as the instruction. Cursor pointing conveys where or which items to change, while dialogue conveys what to change them to. Split each requested modification into these two parts for dialogue generation.

For each item, produce `say`: the values or effects that must be spoken to specify the intended change; and `show_only`: target-identifying information conveyed by the cursor and visual display rather than read aloud. This includes pointed-to item names, visible anchors, timeline positions, cell coordinates, and target filenames or paths. Dialogue must refer to these targets deictically.

Apply `show_only` to information used to identify the target, not to all text visible on screen. New values that cannot be recovered from the visual evidence or a specified source document must be placed in `say`. Do not exclude a required spoken value merely because the same string appears elsewhere on screen.

Prompt D.7. Allocation Between Speech and Visual Evidence (continued)

For cross-document requirements, the source document and the relevant portion to consult must be expressible in `say`. Do not speak the source value or formula itself: the agent must open the source document to retrieve it.

For audio or video modifications, the visual timeline conveys the interval, while speech must state the action, such as trimming, concatenating, speeding up, slowing down, or muting, together with any new parameter that cannot be read from the screen. A selected interval without an action is insufficient.

The same separation applies to rejected and withdrawn proposals. Output a JSON object whose `splits` list contains an item identifier, `say`, and `show_only` for each item.

Prompt D.8. Discussion Generation

System prompt.

You are an experienced recorder of product and engineering review meetings. Output only strict JSON, without explanations or Markdown fences.

Task-template excerpt.

The following is private construction information for an Omni-CoWork task. A team is reviewing files in a workspace and discussing whether to change the indicated content. Turn the specified decisions into a natural multi-person review dialogue.

Use exactly {`n_speakers`} speakers with roles appropriate to the domain. Write natural, conversational Chinese with short utterances, exchanges, interruptions, agreement, and self-correction. Represent pauses through annotations rather than stage directions that would be read aloud.

Include one or two opening utterances introducing the review without announcing the number or details of the changes. Include one or two closing utterances without listing the decisions or repeating target values.

Do not assign one utterance mechanically to every item. Accepted proposals may develop through several exchanges. Interleave decisions rather than grouping all accepted proposals before rejections and withdrawals. At least one rejection or withdrawal must occur between two accepted proposals.

For an accepted proposal, preserve its assigned level of intent abstraction: a direct request, a description of a problem, or commentary implying an edit. Do not turn every proposal into an explicit instruction or provide code or implementation steps. Make final acceptance clear from the spoken exchange, without requiring a formulaic confirmation for every item. State any new value that cannot be recovered from the visual evidence or workspace. For a rejected proposal, make clear that the proposed change will not be applied. A withdrawn proposal requires an initial proposal and a later explicit withdrawal. Non-task dialogue must not introduce an editing requirement.

Never read aloud the target identifiers reserved for visual evidence. Use references such as "this part" or "the item the cursor points to." Match the reference to the artifact type. Mention only visual actions that the recording actually shows; do not claim a zoom, highlight, or annotation that the renderer does not perform.

Return strict JSON containing the speaker list and dialogue turns. Each turn specifies the speaker, spoken Chinese text, associated item identifier when applicable, a turn-status annotation, and an optional pause.

E FUTURE WORK

More effective methods. The Omni-CoWork skill improves task success, but performance remains far from the reliability required for practical deployment, motivating more effective methods for grounded collective reasoning. Promising directions include maintaining structured representations of accepted decisions and preservation constraints, linking each decision to supporting discussion evidence and concrete artifact locations, and updating these links as execution changes the workspace. Verification should distinguish missing changes from incorrect implementations and changes to files not mentioned, and use detected discrepancies to guide further perception and correction. The current skill encourages these behaviors through instructions; it does not establish that agents can carry them out reliably. Comparisons should control tool access and resource budgets to distinguish better grounding from additional computation.

Broader task scope. Future work can also extend the benchmark to more diverse scenarios, larger and more heterogeneous file systems, and richer operations, including artifact creation, reorganization, and coordinated cross-application updates. Longer collaborations would further require handling revised decisions, cross-file dependencies, and concurrent human edits, broadening the study of sustained alignment between collective intent and workspace changes.