SIMULATING ORGANIZATIONAL DATA LINEAGE FOR REALISTIC AND CONTROLLABLE EVALUATION OF ENTERPRISE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous agents are increasingly being used to produce complex enterprise artifacts such as spreadsheets, documents, and presentations. However, generating these scenarios remains difficult since realistic benchmarks require substantial manual effort, while existing synthetic approaches offer limited control over the relevance, diversity, and complexity of generated tasks. Evaluating generated artifacts also remains a challenge since no single reference captures the requirements that is practically imposed by different organizational stakeholders. We introduce LINEAGEBENCH, a synthetic framework that systematically evaluates agents in realistic and diverse enterprise states through multi-persona data lineage. Rather than simulating organizational activity forward towards a potentially misaligned emergent state, LineageBench begins with a high-level deliverable and recursively works backward to identify the upstream artifacts required to produce it along with the organizational personas responsible for them. We instantiate LineageBench with 30 organizational graphs spanning multiple industries, grounded in the O*NET taxonomy and usage statistics. We also leverage the lineage structure to construct rubrics to evaluate each artifact by collating the expectations from each artifact's producer, data providers, and consumers. By varying the lineage distance between a target artifact and the ancestors supplied to an agent, we evaluate 12 frontier agents on more than 500 tasks with controlled compositional complexity. Results show that the strongest agents achieve nearly twice the score of the weakest, yet similarly capable agents can require up to twice as many tool calls. Performance also declines by 31% on average as lineage distance increases from one to five hops. Together, these findings expose novel gaps in the capability and efficiency of enterprise agents.

1 INTRODCUTION

Modern enterprises run on the knowledge their teams create, share, and build upon. To do so productively, they rely on structured and semi-structured documents: spreadsheets that track budgets and pipelines, reports and specifications written as text documents, and slide decks that summarize findings for decision makers [1]. These productivity artifacts are rarely authored in isolation and are often derived from one another. For instance, a quarterly review is assembled from sales reports and workbooks from several data analysts. Producing one artifact from others is often among the most common forms of knowledge work, and also among the most laborious as it requires locating the relevant upstream sources, reconciling stylistic and presentational differences across contributors, and synthesizing their content into a new format tailored to a different audience.

Recent advances in agentic artificial intelligence (AI) have shown great promise in their use in drafting documents, populating spreadsheets, and generating presentations from natural language instructions (Patwardhan et al., 2026). However, deploying them in enterprise settings demands a level of quality and factual fidelity that current systems do not reliably achieve. A useful artifact must be both faithful to its sources and compatible with downstream workflows, as failures in either can propagate through subsequent deliverables. Thus, we need benchmarks at scale that can measure

[1] https://www.microsoft.com/en-us/worklab/work-trend-index

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.

Benchmark	Artifact production	Automated task/ world synthesis	Multi-persona ownership	Complexity-controlled Task construction	Multi-persona evaluation
GDPval (Patwardhan et al., 2026)	✓	–	–	–	–
APEX-Agents (Vidgen et al., 2026)	✓	–	–	–	–
WorkArena++ (Boisvert et al., 2024)	–	✓	–	✓	–
OfficeBench (Wang et al., 2024)	✓	✓	–	–	–
Synthetic Computers (Ge et al., 2026)	✓	✓	✓	–	–
OmniBench (Bu et al., 2025)	✓	✓	–	✓	–
Workspace-Bench (Tang et al., 2026)	✓	✓	–	–	–
PPT-Eval (Gandhi et al., 2026)	✓	–	–	–	–
LINEAGEBENCH	✓	✓	✓	✓	✓

Table 1: Comparison of LineageBench with representative benchmarks, where a checkmark indicates that a functionality is provided and a dash indicates that it is not.

whether they faithfully compose information across multiple upstream artifacts to satisfy a goal while anticipating any downstream usage.

Such benchmarks remain scarce because enterprise data is largely proprietary, while public corpora does not capture the quality and dependency structure of real enterprise work. One line of work manually designs realistic scenarios and synthesizes the corresponding environment state in a single shot (Wang et al., 2024), which can lead to a lack of scenario diversity. To enhance diversity, real systems have been simulated that evolve the environment over time by continually generating new artifacts (Ge et al., 2026). But such simulations offer little control over the quality and complexity of the resulting state and may produce many artifacts of limited relevance. Moreover, both paradigms typically model artifacts independently or from a single persona's perspective, whereas organizational artifacts are produced across personas and functions, with each artifact transforming or contributing information to subsequent artifacts. This motivates a goal-directed generation mechanism organized around explicit data lineage, in which every artifact is grounded in identifiable upstream sources and serves a defined role in producing a downstream deliverable.

To address these limitations, we introduce LINEAGEBENCH that enables (1) scalable synthesis of realistic and diverse multi-persona enterprise states, (2) task construction with controllable compositional complexity, and (3) multi-persona evaluation of artifact quality. Table 1 summarizes our contributions compared with the literature. LineageBench achieves this by explicitly generating the lineage graph of organizational data within these enterprises. We use O*NET occupations and work activities to sample realistic goals that these personas may perform using AI. Then, the graph is recursively constructed by working *backward* to identify the required upstream artifacts and the organizational personas responsible for producing them. We then use the resulting graph structure to both (a) control the task difficulty by varying the lineage distance between the requested artifact and the provided inputs, and (b) generate artifact quality specifications by eliciting task success criteria from different stakeholders including ancestors, owner, and downstream consumers. Results on 30 generated organizations show that the Grok-4.5/Claude-Opus-4.6/GPT-5.6 generate high-quality artifacts but their performance declines as we provide inputs at higher hops, while similarly capable agents can be separated by their efficiency of the tool calls.

2 RELATED WORK

Realistic artifact-based benchmarks. Recent benchmarks have moved from short-form answers toward economically meaningful work products that includes creation of real work artifacts. GDPval (Patwardhan et al., 2026) contains tasks written by experienced professionals across occupations in economically important sectors and evaluates diverse deliverables. WindowsWorld uses a multi-agent framework steered by occupations to generate professional cross-application workflows (Li et al., 2026a), while APEX-Agents (Vidgen et al., 2026) provides expert-built project worlds, tasks, rubrics, and gold outputs for investment banking, consulting, and law. Benchmarks have also been created to focus on specific applications such as spreadsheet manipulation and end-to-end business workbooks (Ma et al., 2024; Zhu et al., 2026), powerpoint editing (Gandhi et al., 2026), and verifiable operations over documents, spreadsheets, slides, and PDFs (Jiang et al., 2026). These works establish the importance of occupational realism, native file formats, and artifact-quality evaluation,

but their environments and tasks generally require substantial expert curation and do not explicitly model how artifacts produced by different organizational roles contribute to one another.

Open-ended interactive enterprise environments. Computer-use benchmarks evaluate agents in open-ended environments, where they navigate multiple office and desktop applications, transfer information across interfaces, and modify persistent state. OfficeBench (Wang et al., 2024) and OSWorld (Xie et al., 2024) introduced realistic office and operating-system tasks with task-specific files and state-based checks. Subsequent works (Li et al., 2026a; Wang et al., 2025; Yuan et al., 2026; Li et al., 2026b) extend this to more professional and long-horizon workflows spanning multiple applications. Enterprise environments have also been specifically designed over automatically verifiable workplace operations (Styles et al., 2024; Trivedi et al., 2024; Malay et al., 2026), realistic CRM tasks (Huang et al., 2025; 2026), and workflows across the company or SaaS services (Xu et al., 2025; Shi et al., 2026). While these benchmarks evaluate the agents' ability to interact with multiple applications in open-ended environments, we focus on controlled evaluation of their ability to produce high-quality artifacts by composing upstream evidence from different applications.

Synthetic organizational simulation. Synthetic generation of realistic enterprise environments reduces the cost of manually constructing each task world. Synthetic Computers creates persona-conditioned computers with rich file systems and then runs long-horizon productivity simulations within them (Ge et al., 2026). MyPCBench evaluates personal assistants in a desktop and application state seeded for a canonical persona (Jang et al., 2026), while FileGram generates persona-driven file-system workflows and behavioral traces for evaluating agent memory and personalization (Liu et al., 2026). OmniBench synthesizes controllable task graphs by composing subtasks in a bottom up manner through typed input and output resources, but lacks realistic diversity (Bu et al., 2025). More broadly, multi-agent simulations show how role specialization can induce social and collaborative behavior (Park et al., 2023; Li et al., 2023; Hong et al., 2024). Thus, existing approaches either support high controllability at the cost of real-world diversity (through subtask composition), or support high-diversity forward simulation with limited control on the emergent state being desirable (like SyntheticComputers). LineageBench instead starts from a realistic target deliverable and expands backward across organizational roles to infer and materialize its upstream artifact lineage, generating diverse, realistic, and controllable tasks.

3 LINEAGEBENCH

Real enterprise states can be messy, where information flows in an unpredictable manner across different people within an organization. We model this state by abstracting out the dynamic process using stable versioned artifacts owned by one person and derived from other artifacts. Specifically, we obtain state $\mathcal{S} := (\mathcal{G}, \mathcal{O})$, where $\mathcal{G}$ is a graph of interdependent enterprise data and $\mathcal{O}$ maps each datum to the person who owns it. Furthermore, we assume that the dependencies across artifacts do not form a loop such that $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ forms a directed acyclic structure in which each node $v \in \mathcal{V}$ denotes a datum and each edge $(u, v) \in \mathcal{E}$ indicates that v is derived, at least in part, from u. We write $\mathcal{U}(v) := \{u \in \mathcal{V} : (u, v) \in \mathcal{E}\}$, $\mathcal{D}(v) := \{d \in \mathcal{V} : (v, d) \in \mathcal{E}\}$ for the immediate upstream dependencies and downstream consumers of v, respectively. We also assume that each datum is produced and owned by a single person, given by an ownership map $\mathcal{O} : \mathcal{V} \to \mathcal{P}$ over a set of personas $\mathcal{P}$, and use the shorthand $\mathcal{U}_{\mathcal{P}}(v) := \{\mathcal{O}(u) : u \in \mathcal{U}(v)\}$, $\mathcal{D}_{\mathcal{P}}(v) := \{\mathcal{O}(d) : d \in \mathcal{D}(v)\}$ for the immediate upstream and downstream stakeholders of v.

LineageBench explicitly generates enterprise states as multi-persona lineage graphs, where the graph helps in constructing the tasks with controlled complexities and multi-persona rubrics, while the ownership map helps in synthesizing realistic and diverse scenarios. Figure 1 illustrates the pipeline of LineageBench and we describe each step in detail below.

3.1 SYNTHETIC GENERATION

We generate the lineage graph by following a three stage process: *sampling* realistic telemetry-grounded goals (Figure 1a), *backward expansion* of the lineage structure (Figure 1b), followed by *forward materialization* of artifact content (Figure 1c). Algorithm 1 provides a detailed step-by-step algorithm for the generation procedure that we describe below.

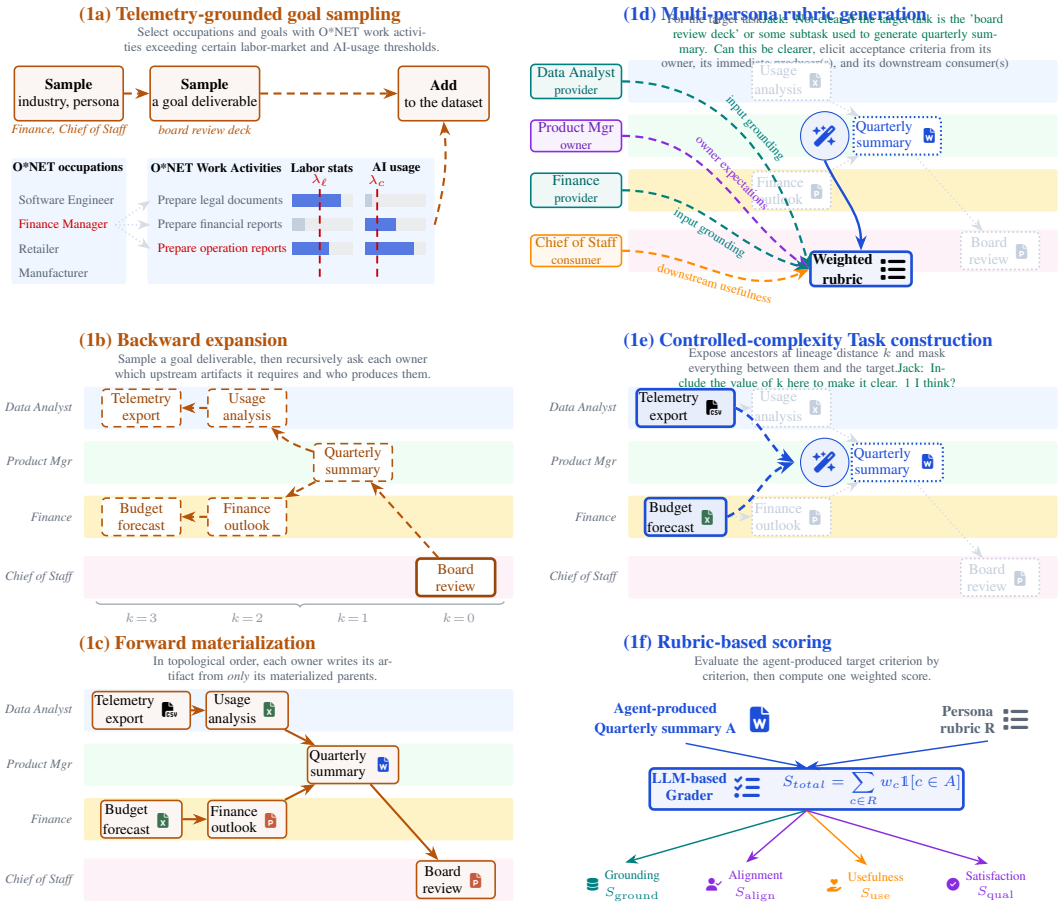


Figure 1: **Overview of the LineageBench framework.** The left column illustrates the ***synthetic generation pipeline*** of a graph from the goal of a board review deck: **(1a)** how goals are sampled grounded in telemetry, **(1b)** how each sampled goal is recursively expanded into persona-owned upstream dependencies; and **(1c)** how the resulting lineage is materialized in topological order. The right column illustrates the ***evaluation pipeline*** for a task of generating the Quarterly summary: **(1d)** how rubrics are generated from multiple stakeholders; **(1e)** how tasks are constructed by varying the given inputs using lineage distance k; and **(1f)** how agent-produced artifacts are evaluated against the rubrics using an LLM-based grader into different dimensions.

Telemetry-grounded goal sampling. We design our simulation such that its quality can be controlled from ground up by building organizations that represents realistic work in specific industries. To this end, we consider personas o based on the O*NET occupations (O*NET OnLine, 2026) and an LLM-generated high-level *goal* deliverable g that each such persona would plausibly be responsible for. For example, *a chief of staff in finance sector needs a board review deck*. Then, we classify whether these generated goals align with the persona and with the current usage of AI in the community. We leverage the labor expectations for the common work activities conducted by the O*NET occupation and then check if the corresponding work activity for the sampled goal exceeds a threshold λ_ℓ for this occupation. Additionally, we check if AI is commonly used to automate the activity described in the goal, using a threshold λ_c over the observed public usage statistics of Claude Anthropic (2026). By grounding our synthesis in these carefully crafted scenarios, we ensure that the synthesized data simulates a real world. We provide additional details and statistics in Appendix B.

Backward expansion. Starting from the sampled goal g, we recursively ask a large language model (LLM) to identify the upstream artifacts required to produce the requested datum, along with the persona who would own each required input. Formally, at each node v with owner p, an LLM-based procedure DERIVEUPSTREAMDEPENDENCIES is called that returns a set of dependency–owner pairs $\mathcal{R}_v := \{(u_1, p_{u_1}), \ldots, (u_m, p_{u_m})\}$, each of which is added to $\mathcal{G}$ as a node with owner

$\mathcal{O}(u_i) = p_{u_i}$ and an edge (u_i, v). When a proposed dependency corresponds to a datum already present in $\mathcal{V}$, we reuse the existing node and add only the edge as long as it does not cause a cycle to form, allowing artifacts to be shared across branches and across goals rather than duplicated (see App E for more details). Newly introduced nodes are then expanded recursively, unless the node is identified as a source/raw data that its owner would obtain directly rather than derive from other internal artifacts, or if it lies at a maximum lineage depth $H_{\max}$. Repeating this process for N_G sampled goals yields an organization in which multiple deliverables may draw on overlapping upstream work. At the end of this stage, nodes are *specifications* rather than content: each records what the datum represents, who owns it, and what it depends on, but not the information it contains.

Forward materialization. We then traverse $\mathcal{G}$ in topological order and materialize (i.e., fill with real content) the artifact $\mathcal{A}(v)$ for each node from its already-materialized parents and the owner/task descriptions using an LLM-based agent, while the source artifacts are generated from their specification alone. Each artifact is produced by an agent instantiated with the persona $\mathcal{O}(v)$ and written in a format appropriate to its role and request, so that tabular data becomes a spreadsheet, an analysis or memo becomes a document, and an executive-facing deliverable becomes a presentation. Materializing in this direction ensures that an artifact is generated from its upstream dependencies only, and never from its descendants. As a result, downstream conclusions cannot leak backward into the evidence that is supposed to support them, and values appearing in a derived artifact can be traced to the parents from which they were computed. The final materialized state $\mathcal{S}^* := (\mathcal{G}, \mathcal{O}, \mathcal{A})$ therefore provides not only a collection of realistic enterprise files, but a ground-truth record of who produced each artifact, which information it was derived from, and who eventually may use it.

3.2 Evaluation

Once we have a materialized enterprise state $\mathcal{S}^*$, we evaluate agents on their ability to produce a high-quality artifact corresponding to a given node t from the perspective of the owner $\mathcal{O}(t)$. For systematically evaluating the produced artifact, we follow a three-stage process: collecting success criteria from different stakeholders (Figure 1d), constructing tasks with controlled complexity (Figure 1e), and finally scoring the produced artifact against the rubrics (Figure 1f). We also validate the quality of the generated rubrics and scores using human annotation in Appendix D.5.

Multi-persona Rubric Generation. Given a target node (*Quarterly Summary.docx*) that we want to test agents on, we collect the success criteria from its immediate ancestor(s) providing the data directly (*Data Analyst*, *Finanace*), the owner of this artifact (*Product Manager*), and the direct consumer(s) of this artifact in the lineage graph (*Chief of Staff*). Each persona then provides a list of success criteria along with the importance weights (from 1–5, 5 = goal-critical fact the whole deliverable hinges on; 1 = minor polish), which are then merged and deduplicated into a consolidated weighted list. Additional details and complete prompts are provided in Appendix C.2.

Controlled-complexity Task Construction. The lineage graph further allows us to construct tasks with variable complexity by varying the set of *provided* input artifacts provided to produce the target node. Instead of giving the immediate parents of the target node, we can also give the set parents of each immediate parent (i.e., *Telemetry export.csv* instead of *Usage analysis.xlsx* and *Budget forecast.xlsx* instead of *Finance outlook.xlsx*) to produce the target artifact. Since information is fully derivable within the lineage graph, this keeps the task solvable while changing its complexity since more steps would be needed by the agent to transform the data to align it with the requirements. More generally, we define artifact inputs to lie at a lineage distance k to be $\mathcal{I}_k(t) := \{\mathcal{A}(u) : (\delta(u,t) = k) \vee ((\delta(u,t) \leq k) \wedge (\mathcal{U}(t) = \varnothing))\}$, where $\delta(u, v)$ denotes the distance of the shortest directed path between u and v. The resulting task for a target t is then formed by $\tau_k(t) := \big(t,\ \mathcal{I}_k(t),\ \mathcal{O}(t)\big)$. As k increases, the intermediate artifacts that ordinarily bridge the underlying data and the target are withheld, and the agent must reconstruct progressively deeper compositions of the same evidence.

Rubric-based scoring. Finally, we use different agents to solve each constructed task to produce a candidate artifact. Each candidate is then evaluated against the set of rubrics for the target artifact using a grader judge LLM $\mathcal{J}$ that checks whether each rubric item is satisfied (1), partially satisfied (0.5, or not satisfied (0). The ***Blended* (Bl.)** score of an artifact is then the weight-normalized sum over each item scores. Each item is also classified into one of the following dimensions and we also find the corresponding weighted-average over all the items belonging to each:

- **Visual structure (VS)**: check for presentation and layout (e.g., titled sections/banners, readable and well-formed tables, consistent styling, and content fit).
- **Goal satisfaction (GS)**: whether the artifact accomplishes the deliverable's stated purpose set out by the owner (e.g. "forecast demand per lane-week", "reconcile the two specimen extracts").
- **Owner alignment (OA)**: whether the content matches the role, scope, and intent that the owner's persona in their team would realistically make.
- **Completeness (CP)**: whether all required content is present, i.e., every mandated field, column, section, record, or case is populated rather than omitted or left as a placeholder.
- **Consumer usefulness (CU)**: whether the artifact gives its downstream consumers what they need to act, i.e., the information and structure to use in their workflow rather than merely correct.
- **Producer grounding (PG)**: whether every concrete claim, number, and figure is grounded and derivable from the supplied inputs.

4 EXPERIMENTAL SETUP & RESULTS

We instantiate LINEAGEBENCH with a corpus of 30 synthetic enterprises whose personas and goals are grounded in the O*NET occupation taxonomy (Section 3.1). We first generate 2,700 candidate goals across 540 O*NET occupations selected for technological exposure and organizational hierarchy ($\sim$ 5 goals per occupation). We then filter on two per-goal signals: a *labor-market weight* $\lambda_\ell = 2.0$ and a *Claude-usage* share $\lambda_c = 0.05\%$, giving us a set of 93 occupations that represents AI-relevant economically-meaningful work. From these, we sample 30 different organizations with goals that belong to different occupations and cover different work activities. We provide a detailed analysis of the coverage and filtering results in Appendix B. Graphs are then generated with a maximum depth $H_{max} = 6$, maximum goals $N_G = 5$ per enterprise, at most 4 direct dependencies per artifact, and at most 50 artifacts per goal such that the resulting enterprise graphs contain 54–250 artifacts (mean 138). Each node is materialized using GPT-5.6-sol [2] into specified artifact formats. For evaluation, we sample at most 10 non-source targets including the final goals from each graph, that gives us 236 total evaluation targets spanning terminal deliverables and intermediate nodes.

We evaluate a large scale of frontier systems as artifact-producing agents ranging from the OpenAI family [3] (GPT-5.4, GPT-5.5, and GPT-5.6 sol/terra/luna); Anthropic family [4] (Claude Opus 4.8, Opus 4.6, Sonnet 4.6, and Haiku 4.5); NVIDIA's Nemotron-3-Ultra [5]; to xAI's Grok-4.5 and 4.6 [6]. All models are run with a `high` reasoning effort. We use Claude Opus 4.8 to generate the rubrics for each target and each produced artifact is evaluated against these with different LLM graders. In particular, we use three independent judges: Claude Opus 4.8, Claude Sonnet 5, and GPT-6 and score a rubric item to be the average over all the judges. Unless stated otherwise, we report the immediate-parent setting $\tau_1(t)$, in which an agent receives the 1-hop ancestor frontier $\mathcal{I}_1(t)$ of the target together with the role context of its owner $\mathcal{O}(t)$ and must produce a good artifact $\mathcal{A}(t)$. More details of the task and owner context are provided in Appendix C.4 and E.

4.1 CAN FRONTIER AGENTS GENERATE DESIRED ARTIFACTS FROM DIRECT INPUTS?

Table 2 reports the blended and per-dimension scores for tasks with immediate parents, i.e., $\tau_1(t)$. We find that the strongest agents match or exceed the pipeline's own reference artifact: Grok-4.5 leads at 0.760 blended, Grok-4.6 follows at 0.744, and Claude Opus 4.6, the GPT-5.6 family, and Opus 4.8 all sit at or above the GPT-5.6 Reference (0.687). This indicates that our evaluation pipeline is not biased towards the reference artifact and current frontier agents can synthesize a more well-specified deliverable. Results also separate the agents into clear tiers—a top group (the Grok pair, Opus 4.6/4.8, the GPT-5.6 family, 0.69–0.76), a middle group (Sonnet 4.6 and Nemotron-3-Ultra, 0.57–0.65), and a markedly weaker group (Haiku 4.5 at 0.52 and GPT-5.4/5.5 at $\approx$ 0.39). This tiering is also found to be stable across the three judges that make up the weighted scores, as we show in per-judge leaderboards in Appendix D.1 and with a high model-level Spearman 0.75–0.92 between judges in Table 12. Across dimensions, *visual_structure* is the easiest (0.76–0.81 for the

[2] `https://developers.openai.com/api/docs/models/gpt-5.6-sol`
[3] `https://developers.openai.com/api/docs/models/`
[4] `https://platform.claude.com/docs/en/models/overview`
[5] `https://build.nvidia.com/nvidia/nemotron-3-ultra-550b-a55b`
[6] `https://docs.x.ai/developers/models`

Table 2: Blended and per-dimension scores at $\tau_1(t)$ on the 236 sampled targets. Each score is averaged over 3 judge outputs and each cell is the mean with its per-target standard deviation as a gray subscript. Bl.: blended; VS: visual_structure, GS: goal_satisfaction, OA: owner_alignment, CP: completeness, CU: consumer_usefulness, PG: producer_grounding. Best per column in **bold**.

Agent	VS	GS	OA	CP	CU	PG	Bl.
Grok-4.5	$0.805_{0.22}$	$\mathbf{0.710}_{0.24}$	$\mathbf{0.766}_{0.22}$	$\mathbf{0.724}_{0.23}$	$\mathbf{0.711}_{0.23}$	$\mathbf{0.796}_{0.16}$	$\mathbf{0.760}_{0.16}$
Grok-4.6	$0.780_{0.21}$	$0.688_{0.24}$	$0.759_{0.23}$	$0.705_{0.23}$	$0.660_{0.24}$	$0.792_{0.16}$	$0.744_{0.16}$
Claude Opus 4.6	$0.771_{0.20}$	$0.673_{0.23}$	$0.691_{0.23}$	$0.706_{0.21}$	$0.683_{0.21}$	$0.725_{0.16}$	$0.705_{0.15}$
GPT-5.6-luna	$0.778_{0.19}$	$0.642_{0.24}$	$0.688_{0.23}$	$0.668_{0.22}$	$0.655_{0.24}$	$0.736_{0.15}$	$0.698_{0.13}$
Claude Opus 4.8	$0.781_{0.19}$	$0.673_{0.23}$	$0.677_{0.23}$	$0.679_{0.21}$	$0.683_{0.23}$	$0.710_{0.16}$	$0.693_{0.14}$
GPT-5.6-sol	$\mathbf{0.809}_{0.18}$	$0.655_{0.23}$	$0.683_{0.23}$	$0.685_{0.20}$	$0.680_{0.24}$	$0.711_{0.15}$	$0.692_{0.13}$
Reference	$0.808_{0.19}$	$0.647_{0.25}$	$0.676_{0.23}$	$0.669_{0.22}$	$0.665_{0.25}$	$0.715_{0.14}$	$0.687_{0.13}$
Claude Sonnet 4.6	$0.715_{0.27}$	$0.590_{0.30}$	$0.642_{0.27}$	$0.625_{0.28}$	$0.618_{0.29}$	$0.690_{0.21}$	$0.653_{0.22}$
GPT-5.6-terra	$0.761_{0.21}$	$0.603_{0.24}$	$0.626_{0.23}$	$0.632_{0.23}$	$0.624_{0.24}$	$0.680_{0.15}$	$0.650_{0.14}$
Nemotron-3-Ultra	$0.639_{0.26}$	$0.534_{0.29}$	$0.549_{0.27}$	$0.562_{0.27}$	$0.599_{0.27}$	$0.587_{0.19}$	$0.571_{0.18}$
Claude Haiku 4.5	$0.596_{0.25}$	$0.502_{0.26}$	$0.506_{0.26}$	$0.530_{0.23}$	$0.545_{0.26}$	$0.523_{0.19}$	$0.521_{0.16}$
GPT-5.4	$0.484_{0.31}$	$0.285_{0.26}$	$0.405_{0.29}$	$0.310_{0.26}$	$0.321_{0.29}$	$0.455_{0.21}$	$0.387_{0.18}$
GPT-5.5	$0.468_{0.31}$	$0.281_{0.26}$	$0.403_{0.29}$	$0.314_{0.27}$	$0.317_{0.28}$	$0.454_{0.20}$	$0.385_{0.17}$

top tier) whereas *goal_satisfaction* and *consumer_usefulness* are the hardest and account for most of the gap between tiers. Per-target dispersion is substantial (SD 0.13–0.30), but each model is averaged over ≥ 235 targets, so the $\sim$0.01–0.02 gaps that separate adjacent models are supported by many paired observations.

4.2 How does the performance vary with varying lineage depth of the inputs?

We now hold the target t fixed and recede its input frontier from the immediate parents $\mathcal{I}_1(t)$ (τ_1) to k-hop ancestors $\mathcal{I}_k(t)$ (τ_k), forcing the agent to reconstruct the deliverable across intervening lineage rather than from its direct sources. Figure 2 traces the full $\tau_1 \to \tau_5$ curve (with standard-error bars) on the 38 sampled targets deep enough to be scored at every level. We find that the quality declines monotonically for every generator, but the slopes differ sharply: GPT-5.6-sol is markedly the most depth-robust, retaining 0.532 at τ_5 (a 0.152 drop from τ_1), whereas Sonnet 4.6 and Haiku 4.5 fall off fastest—to 0.420 and 0.296 (drops of 0.237 and 0.236)—and the Opus pair lands in between ($\approx$ 0.45). This is the compositional gap LineageBench is built to expose: frontier agents are strong single-hop synthesizers but lose fidelity as they must integrate evidence over longer lineage chains. This opens up novel ways to formulate tasks with a common objective but different complexities. We provide results for more targets and agents in Appendix D.

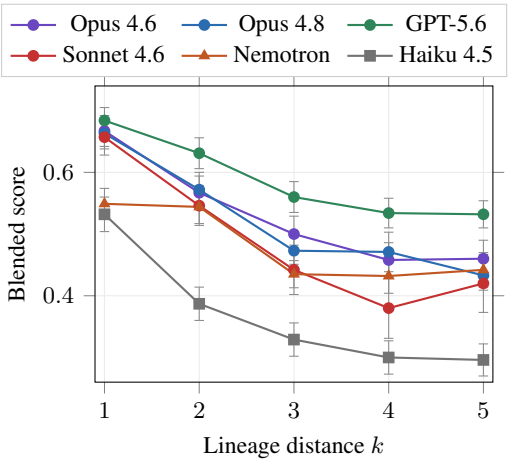


Figure 2: **Quality vs. lineage depth.** Mean blended score ($\pm$SEM) as the input frontier recedes from the immediate parents (τ_1) to 5-hop ancestors (τ_5), on the 38 common targets deep enough to be scored at every level.

4.3 Sensitivity analysis

Next, we analyze how well the agents perform on specific target artifact types, number of visible inputs, target's depth, and owner's industry. Table 3 shows the sensitivity of the results for the first three while Figure 3 illustrates the performance along the industry that the owner belongs to. In particular, we observe the following trends for these factors:

Table 3: Sensitivity of the τ_1 blended score (common set) to target artifact type, number of inputs, and target role. Row maximum in **bold**. Nemo.: Nemotron-3-Ultra.

Split	N	Grok-4.5	Opus 4.6	Opus 4.8	Sonnet 4.6	GPT-5.6-sol	GPT-5.6-luna	Nemo.	Haiku
Artifact type									
Word	73	**0.749**	0.737	0.729	0.681	0.716	0.741	0.592	0.549
Excel	113	**0.774**	0.680	0.681	0.605	0.678	0.688	0.537	0.482
PowerPoint	50	**0.725**	0.697	0.654	0.697	0.670	0.656	0.599	0.560
Number of visible inputs									
1	14	**0.791**	0.750	0.744	0.739	0.758	0.726	0.746	0.661
2–3	21	**0.793**	0.735	0.726	0.707	0.708	0.745	0.625	0.555
4+	201	**0.749**	0.694	0.683	0.636	0.681	0.690	0.549	0.505
Target role									
Intermediate (derived)	214	**0.767**	0.711	0.699	0.652	0.694	0.706	0.572	0.525
Terminal (goal)	22	**0.648**	0.607	0.606	0.609	0.633	0.618	0.520	0.465

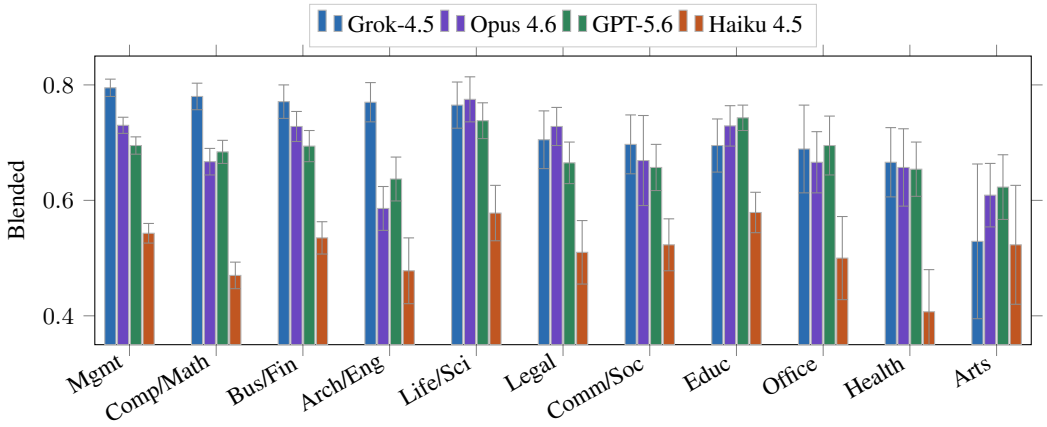


Figure 3: **Quality by industry (owner O*NET occupation industry).** Mean blended score ($\pm$SEM) for four agents spanning the tiers. Industries are ordered by decreasing Grok-4.5 score. Grok-4.5 leads in most functions; Opus 4.6 peaks on Life & Physical Science and Legal but dips on Architecture & Engineering, and Haiku is weakest throughout (worst on Healthcare). Small-N functions (Arts N=4, Community/Social N=6, Healthcare N=7) carry wide error bars.

Artifact type. Word documents are the easiest medium for most models, Excel and PowerPoint harder, but the tiering remains unchanged. We find that GPT-5.6-sol outperforms luna on decks but underperforms on Word documents, while claude models are found to be consistely better than the GPT models on *presentations*. However, Sonnet drops sharply (0.607) on spreadsheets, while the GPT-5.6 family holds up ($\approx$ 0.68).

Number of inputs. Blended quality rises as the input fan-in shrinks, but almost entirely for the *weaker* models: from 4+ inputs to a single input, Haiku climbs $0.507 \rightarrow 0.661$ and Nemotron $0.553 \rightarrow 0.746$, indicating that these models struggle disproportionately with multi-input synthesis. On the other hand, stronger models (Grok-4.5 and the Opus pair) are more stable and move less by changing the number of inputs. Thus, multi-input synthesis is thus a real difficulty axis that separates weak from strong generators, and since most targets (201/235) already carry 4+ inputs, the headline numbers above are dominated by the hard, high-fan-in regime for these weak models.

Target role and industry. Terminal deliverables (*goal* targets) are harder than intermediate (*derived*) nodes for every model as we observe that most agents fall 0.03–0.12—because terminal artifacts tend to represent the most complex synthesis tasks. Figure 3 further breaks the score down by the owner's O*NET occupation's industry. Grok-4.5 leads in most industries (peaking on Management, Computer & Mathematical, and Business & Financial at $\approx$ 0.77–0.80), with exceptions where Opus 4.6 (Life & Physical Science, Legal) or GPT-5.6 (Educational) edge ahead. The hardest cells are Grok-4.5 on Arts (0.529, N=4), Opus 4.6 on Architecture & Engineering (0.586), and Haiku

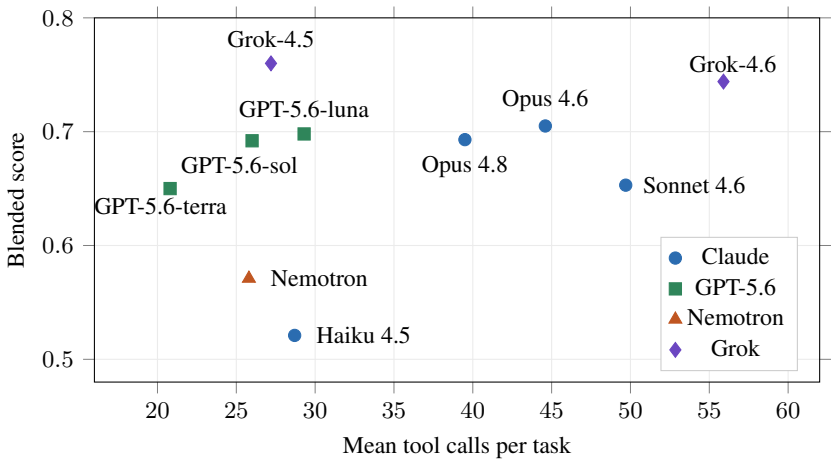


Figure 4: **Quality vs. number of tool calls for τ_1 tasks**. Each point is an agent's mean blended score against its mean number of tool calls per task; upper quadrant is desirable.

on Healthcare (0.407), but we note that these swings are model-specific rather than systematic, but leave the ranking largely intact.

4.4 How efficient are the agents?

Finally we investigate how much *effort* each agent spends to reach its quality, by analyzing the agent trajectories. For every τ_1 deliverable, we count the tool calls (tool invocations and their observations) and reasoning steps (assistant turns) in the trajectory, and average over each model. Figure 4 plots the mean blended score against the mean tool calls. We thus find that quality is *not* bought by more actions. Grok-4.5 defines a pareto frontier since it has the best quality on the board (0.760) at a modest 27.2 calls and the GPT-5.6 family sits alongside it: GPT-5.6-luna reaches 0.698 in 29.3 calls while GPT-5.6/terra deliver 0.65–0.69 in only 21–26. The Opus models buy their quality with far more actions (Opus 4.6 0.705 at 44.6 calls; Opus 4.8 0.693 at 39.5), while Sonnet 4.6 is the least efficient strong model, spending the most actions of any capable system (49.7 calls) yet trailing at 0.653. On the other hand, the later version Grok-4.6 is the most ineffecient as it expends the most tool calls of all (55.9) at a performance which is slightly worse than its predecessor. This indicates that the current frontier is often developed at the cost of efficiency, as achieving higher quality typically requires more tool calls and reasoning steps. Note that tool calls is only a proxy of the total cost as it does not consider the verbosity of the models caused by reasoning, so this analysis only reflects genuine differences in action-level efficiency. We also analyze the total cost based on current public API costs of some agents in Appendix D.

5 Conclusion

We present LineageBench, a framework that generates synthetic agent-evaluation data for realistic enterprise scenarios by simulating organizational data lineage. This lets us evaluate agents on realistic tasks with controlled compositional complexity, scored against automated rubrics elicited from multiple stakeholders. Our results show that frontier agents still have substantial room for improvement, particularly when intermediate compositional steps are masked out, and that stronger models often achieve their gains at the cost of efficiency. By exposing these gaps within a customizable synthetic framework, LineageBench can be used both to train better agents and to keep generating new, harder tasks for held-out evaluation. While O*NET and usage statistics helps us to ground our simulation in realistic goals for each occupation, it can itself be outdated and may not reflect the most current occupational work. Integrating real online telemetry within our framework, including extending it to the backward expansion itself, is thus an important future direction. Future work could leverage our multi-persona rubrics to train agents to internalize upstream and downstream stakeholder expectations, even when those expectations are not explicitly stated.

AI USAGE STATEMENT

Generative AI is an integral component of the LINEAGEBENCH methodology. As detailed in Sections 3 and 4, we used large language models to sample candidate goals, expand lineage graphs backward, materialize their artifacts forward, elicit multi-stakeholder evaluation rubrics, and score generated deliverables. We provide the full prompts and experimental setup details in Appendix E. We also used LLM-based agents to assist with experiment execution and used generative AI for limited writing support, including polishing prose, correcting grammar, and formatting LaTeX. We did not use generative AI to originate the research questions, formulate mathematical claims, design the methodology or experiments, or interpret the reported results. The authors validated the experimental pipeline and reported results and reviewed every AI-assisted change to the manuscript. We take responsibility for the final content of this work, including all text, claims, code, data, and artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. The anthropic economic index. https://huggingface.co/datasets/Anthropic/EconomicIndex, 2026. Accessed: September 19, 2026.

Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault Le Sellier De Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. WorkArena++: Towards compositional planning and reasoning-based common knowledge work tasks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 5996–6051. Curran Associates, Inc., 2024. doi: 10.52202/079017-0195. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0b82662b6c32e887bb252a74d8cb2d5e-Paper-Datasets_and_Benchmarks_Track.pdf.

Wendong Bu, Yang Wu, Qifan Yu, Minghe Gao, Bingchen Miao, Zhenkui Zhang, Kaihang Pan, Yunfei Li, Mengze Li, Wei Ji, Juncheng Li, Siliang Tang, and Yueting Zhuang. What limits virtual agent application? OmniBench: A scalable multi-dimensional benchmark for essential virtual agent capabilities. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5725–5748. PMLR, 2025. URL https://proceedings.mlr.press/v267/bu25b.html.

Apurva Gandhi, Vishwas Suryanarayanan, Raja Hasnain Anwar, Firoz Shaik, Shubhang Desai, Thong Q. Nguyen, Muhammad Taqi Raza, Vishal Chowdhary, and Graham Neubig. PPT-Eval: A benchmark for computer-use agents on PowerPoint tasks. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2606.31154.

Tao Ge, Baolin Peng, Hao Cheng, and Jianfeng Gao. Synthetic computers at scale for long-horizon productivity simulation. *arXiv preprint arXiv:2604.28181*, 2026. URL https://arxiv.org/abs/2604.28181.

Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906, 2017. doi: 10.1007/s00778-017-0486-1. URL https://doi.org/10.1007/s00778-017-0486-1.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, pp. 23247–23275, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6507b115562bb0a305f1958ccc87355a-Paper-Conference.pdf.

Kung-Hsiang Huang, Akshara Prabhakar, Sidharth Dhawan, Yixin Mao, Huan Wang, Silvio Savarese, Caiming Xiong, Philippe Laban, and Chien-Sheng Wu. CRMArena: Understanding the capacity of LLM agents to perform professional CRM tasks in realistic environments. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2411.02305.

Kung-Hsiang Huang, Akshara Prabhakar, Onkar Thorat, Divyansh Agarwal, Prafulla Kumar Choubey, Yixin Mao, Silvio Savarese, Caiming Xiong, and Chien-Sheng Wu. CRMArena-Pro: Holistic assessment of LLM agents across diverse business scenarios and interactions. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=EPlpe3Fx1x.

Lawrence Keunho Jang, Andrew Keunwoo Jang, Jing Yu Koh, and Ruslan Salakhutdinov. MyPCBench: A benchmark for personally intelligent computer-use agents. *arXiv preprint arXiv:2606.16748*, 2026. URL https://arxiv.org/abs/2606.16748.

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. *arXiv preprint arXiv:2607.19865*, 2026. URL https://arxiv.org/abs/2607.19865.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2303.17760.

Jinchao Li, Yunxin Li, Chenrui Zhao, Zhenran Xu, Baotian Hu, and Min Zhang. WindowsWorld: A process-centric benchmark of autonomous GUI agents in professional cross-application environments. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15262–15280. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.findings-acl.750. URL https://aclanthology.org/2026.findings-acl.750/.

Wanli Li, Bowen Zhou, Yunyao Yu, Zhou Xu, Yifan Yang, Dongsheng Li, and Caihua Shan. WeaveBench: A long-horizon, real-world benchmark for computer-use agents with hybrid interfaces. *arXiv preprint arXiv:2606.09426*, 2026b. URL https://arxiv.org/abs/2606.09426.

Shuai Liu, Shulin Tian, Kairui Hu, Yuhao Dong, Zhe Yang, Bo Li, Jingkang Yang, Chen Change Loy, and Ziwei Liu. FileGram: Grounding agent personalization in file-system behavioral traces. *arXiv preprint arXiv:2604.04901*, 2026. URL https://arxiv.org/abs/2604.04901.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.14991.

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

O*NET OnLine. O*NET OnLine. Web page, 2026. URL https://www.onetonline.org/. Accessed: September 19, 2026.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 2023. URL https://arxiv.org/abs/2304.03442.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simon Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie Kim, Samuel Miserendino, Gildas Chabot, David Li, Patrick Chao, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating AI model performance on real-world economically valuable tasks. In *International Conference on Learning Representations*, volume 2026, pp. 24005–24040, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/290c2430f91912204f30bbcc990fff1d-Paper-Conference.pdf.

Kean Shi, Zihang Li, Tianyi Ma, Zengji Tu, Jialong Wu, Xinbo Xu, Qingyao Yang, Ruoyu Wu, Weichu Xie, Ming Wu, Jason Zeng, Michael Heinrich, Elvis Zhang, Liang Chen, Kuan Li, and Baobao Chang. SaaS-Bench: Can computer-use agents leverage real-world SaaS to solve professional workflows? *arXiv preprint arXiv:2605.15777*, 2026. URL https://arxiv.org/abs/2605.15777.

Olly Styles, Sam Miller, Patricio Cerda-Mardini, Tanaya Guha, Victor Sanchez, and Bertie Vidgen. Workbench: a benchmark dataset for agents in a realistic workplace setting. *arXiv preprint arXiv:2405.00823*, 2024.

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. *arXiv preprint arXiv:2605.03596*, 2026. URL https://arxiv.org/abs/2605.03596.

Kiran Tomlinson, Sonia Jaffe, Will Wang, Scott Counts, and Siddharth Suri. Working with ai: measuring the applicability of generative ai to occupations. *arXiv preprint arXiv:2507.07935*, 2025.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850.

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. *arXiv preprint arXiv:2601.14242*, 2026. URL https://arxiv.org/abs/2601.14242.

Weixuan Wang, Dongge Han, Daniel Madrigal Diaz, Jin Xu, Victor Rühle, and Saravan Rajmohan. OdysseyBench: Evaluating LLM agents on long-horizon complex office application workflows. *arXiv preprint arXiv:2508.09124*, 2025. URL https://arxiv.org/abs/2508.09124.

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. *arXiv preprint arXiv:2407.19056*, 2024. URL https://arxiv.org/abs/2407.19056.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, pp. 52040–52094, 2024. doi: 10.52202/079017-1650. URL https://doi.org/10.52202/079017-1650.

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, pp. 10291–10321, 2025. doi: 10.52202/085713-0315. URL https://doi.org/10.52202/085713-0315.

Mengqi Yuan, Zilong Zhou, Xinzhuang Xiong, Weiming Wu, Jiayang Sun, Jiamin Song, Kaiqian Cui, Bowen Wang, Haoyuan Wu, Yitong Li, Dunjie Lu, Haikong Lu, Qi Zhen, Xinyuan Wang, Jiaqi Deng, Yuhao Yang, Cheng Chen, Boyuan Zheng, Alex Su, Xiao Yu, Hao Zou, Saaket Agashe, Xing Han Lu, Manpreet Kaur, Zhengyang Qi, Vincent Sunn Chen, Frederic Sala, Dayiheng Liu, Junyang Lin, Zhou Yu, Yu Su, Siva Reddy, Xin Eric Wang, Peng Qi, Tianbao Xie, and Tao Yu. OSWorld 2.0: Benchmarking computer use agents on long-horizon real-world tasks. *arXiv preprint arXiv:2606.29537*, 2026. URL https://arxiv.org/abs/2606.29537.

Jian Zhu, Yuzheng Zhang, Zeyao Ma, Bohan Zhang, Armin Schoepf, Daniel Woloch, Peter Yiliu Wang, Guangyu Robert Yang, Samuel Jacob, Siddharth Nagisetty, Abhiram Chundru, Jean Lin, Spencer Mateega, and Jing Zhang. SpreadsheetBench 2: Evaluating agents on end-to-end business spreadsheet workflows. *arXiv preprint arXiv:2606.29955*, 2026. URL https://arxiv.org/abs/2606.29955.

APPENDIX

A ADDITIONAL RELATED WORK

Data lineage and controlled task construction. Data-provenance systems have long represented how outputs derive from upstream data and transformations (Herschel et al., 2017). More recently, Workspace-Bench constructs heterogeneous workspaces and associates each task with a file-dependency graph for evaluating retrieval and reasoning over explicit and implicit dependencies (Tang et al., 2026). Graph composition also provides scalable control over task structure in OmniBench (Bu et al., 2025) and paired explicit/implicit variants in WorkArena++ (Boisvert et al., 2024). However, none of these works use the data lineage as a way to construct the tasks themselves, which is the focus of the current work. We construct the lineage graph in a backward manner from realistic goals rather than as an emergent state and then, use the same graph to construct tasks with controlled complexity and to construct rubrics for evaluation.

Artifact quality evaluation. Open-ended professional artifacts admit multiple valid realizations, making exact final-state checks insufficient on their own. GDPval (Patwardhan et al., 2026) uses blind comparisons by occupation experts; APEX-Agents (Vidgen et al., 2026) pairs expert rubrics with gold outputs and a validated model judge; PPT-Eval (Gandhi et al., 2026) combines programmatic checks with language- and vision-model rubric judgments; and WeaveBench (Li et al., 2026b) grades both deliverables and execution traces. LineageBench uses rubric-based model evaluation, but derives its criteria blindly from the organizational roles already encoded in the lineage graph: upstream producers specify factual-grounding requirements, the owner specifies goal satisfaction and alignment, and downstream consumers specify the information needed for subsequent work.

B TELEMETRY-GROUNDED GOAL SAMPLING

We use O*NET's hierarchical decomposition of occupation-specific tasks and work activities to seed each organization's goals O*NET OnLine (2026); Tomlinson et al. (2025). As shown in Table 4, we generate 2,700 goals across 540 seed organizations that span 12 major O*NET occupation groups/industries. We do not consider 9 of the occupations to create organization graphs as these do not form valid graphs for the sampled goals. At this stage, our generated goals are dominated by activities relating to healthcare and education-related occupations while legal and community-service occupations are scarce. Its work activity hierarchy classifies broad categories of work into specific activities across 4 levels of granularity: *Generalized* → *Intermediate* → *Detailed* work-activity hierarchy (GWA → IWA → DWA → tasks). To assign goals to each persona, we map tasks to intermediate work activity (IWA) definitions following Tomlinson et al. (2025), and maximize diversity and coverage of generated goals. Table 4 summarizes the coverage for our seed tasks at different levels of the work activity taxonomy.

Quantity	Count
Generated goals	2,700
Seed organizations	531
Distinct O*NET occupations (SOC)	540
Distinct Generalized Work Activities (GWA)	36
Distinct Intermediate Work Activities (IWA)	246
Distinct Detailed Work Activity (DWA)	835
Occupation-major-group "industries"	12

Table 4: O*NET taxonomy coverage of the seed corpus. Each goal and persona belonging to a seed organization/occupation is classified into an `GWA→IWA→DWA`.

B.1 REAL-WORLD USAGE SIGNALS

We then leverage the weight that the labor market survey assigns to the IWA for that occupation along with AI-assisted work data Anthropic (2026) to identify IWAs that can be accomplished with the use of AI. In particular, for each IWA, we get:

- **Labor-market weight**: For each O*NET occupation, we assess how economically central each IWA is to the role. Using task frequency ratings, we calculate the share of each IWA across the work each occupation performs.
- **Claude usage**: We use Claude usage data Anthropic (2026) to measure what types of activities are performed with the help of an AI assistant. We calculate the share of each AI-assisted IWA performed compared as a fraction of the work performed by each occupation.

Table 5 characterizes the distribution of occupation-specific IWA shares and Claude conversation IWA assigned to the generated goals. We observe that both distributions are skewed, where most IWAs in the AI-assisted distribution see little usage, and the labor market weight distribution has one goal in five with zero weight for its occupation.

Signal	p10	p25	p50	p75	p90	max
Claude usage share (%)	0.017	0.053	0.163	0.528	1.590	20.46
Labor-market weight	0.000	0.886	5.167	11.905	22.364	100.0

Table 5: Per-goal distribution as percentiles of the two usage signals over 2,700 goals.

We thus **discard** an organization for being unrealistic if *any* of its goals falls below a threshold on any of the usage signals, so that every surviving org is strong on both axes, i.e.,

$$\text{keep org } o \iff \min_{g \in o} \text{claude}(g) \geq \lambda_c \;\wedge\; \min_{g \in o} \text{lm}(g) \geq \lambda_\ell. \tag{1}$$

Table 6 sweeps the two thresholds over the 540 organizations. We thus adopt λ_ℓ=2.0, λ_c=0.05, which retains **93** organizations where every industry still contributes multiple survivors.

$\lambda_\ell \setminus \lambda_c$	0.00	0.01	0.02	**0.05**	0.10
0.01	347	278	235	169	103
0.50	317	257	218	154	97
1.00	277	221	189	132	87
2.00	200	160	135	**93**	67
5.00	88	83	75	54	46

Table 6: Number of organizations over threshold sweeps where the bold cell is adopted.

B.2 Industry-balanced undersampling

The 93 survivors remain skewed as the usage strength correlates with industry. Thus, we sample a compact target of **30** organizations with a round-robin that fills the remaining slots by rotating across the least-represented industries, ranking within each by the usage strength. Table 7 the per-industry counts at each stage. For the depth and generalization experiments we additionally materialize a held-out extension of 5 organizations, giving the 35 enterprises evaluated in Section 4. Table 8 lists all 35: the seed occupation (O*NET title and SOC code), its major-group industry, the dominant seed work activity (IWA), and the number of artifacts (total and derived) in the materialized graph.

C Additional Experimental Details

C.1 Synthetic generation pipeline

Algorithm 1 describes the synthetic generation pipeline in more detail. We first sample a persona and a high-level deliverable that such a person would own, which are then *expanded backward* where a model repeatedly names the upstream inputs each datum requires and who would own them, recursing until every branch reaches either a raw source or a maximum lineage depth. Dependencies that coincide across branches or goals reuse the same node, so deliverables share upstream work

Industry (O*NET major group)	seed 531	filtered 93	final 30
Management	56	6	5
Business and Financial Operations	47	10	3
Life, Physical, and Social Science	60	12	3
Office and Administrative Support	51	21	3
Educational Instruction and Library	61	10	2
Computer and Mathematical	36	7	2
Sales and Related	22	7	2
Architecture and Engineering	56	6	2
Arts, Design, Entertainment, Sports, Media	40	5	2
Community and Social Service	14	4	2
Legal	7	3	2
Healthcare Practitioners and Technical	89	2	2
Total	**540**	**93**	**30**

Table 7: Per-industry counts through the pipeline. The usage filter re-shapes the pool (healthcare collapses (low per-IWA Claude usage), while office/admin rises) while balancing then flattens the final 30 to 2–5 per industry across all 12.

as they do in a real organization without forming a cycle. Repeating this for several goals yields a lineage graph of *specifications*: who owns what, and what depends on what, but not the content for any node. This is done by *materializing forward* in dependency topological order, generating each artifact only from its already-produced parents, so information never leaks from a result back into its own evidence. Crucially, each freshly generated artifact passes through an automated *grounding audit*: an auditor model (we use Claude-Opus-4.8 in our implementation) checks that the artifact is faithful to, and derivable from, its declared inputs. If it is not, the artifact is regenerated with the audit's feedback, up to a fixed number of revision rounds. The output is a materialized enterprise: a set of realistic files plus a complete record of who produced each artifact and from which inputs.

C.2 Rubric generation

Each generated artifact is evaluated against a *blind rubric*: a weighted checklist of concrete, atomic acceptance criteria written *before* any candidate answer exists (Figure 5). Working only from the target's specification and its input artifacts, three critics author criteria from complementary viewpoints—the *owner* (does the artifact accomplish what its author set out to produce?), the *consumer* (does it give downstream users what they need?), and the *producer* (is every claim grounded in the supplied inputs?). For each, we only provide the agentic generation task description along with the direct inputs and do not provide the materialized artifact for the generation. Additionally, we provide the data providers and consumers with the context of the request that connects them with the requested artifact. A synthesizer then merges and de-duplicates these into a single weighted rubric organized under six dimensions: visual structure, goal satisfaction, owner alignment, completeness, consumer usefulness, and producer grounding. Authoring the rubric without using any candidate artifact prevents the criteria from being tailored to a particular answer and lets every model be graded against a fair standard. While *Visual_structure* is largely a surface property, *goal_satisfaction*, *completeness*, and *producer_grounding* are the content-bearing dimensions that require reading the artifact against its inputs, and they account for most of the spread between strong and weak generators. On the other hand, *consumer_usefulness* measures how well the models can anticipate downstream consumer usage.

C.3 Candidate authoring and scoring

Given a task, i.e., the target's instruction, its owner's role context, and the provided k-hop input files, each generator authors an answer in the *sandboxed office agent* as detailed below. The agent reads the inputs, works over multiple turns to analyze and assemble the content, and saves a real Office file. That file is then graded by each judge against the task's blind rubric, item by item; the per-item results combine by their weights into a single blended score in $[0, 1]$ and into the six per-dimension scores reported in the main text. The reference artifact produced during materialization is graded by the identical procedure, providing an in-distribution anchor rather than a human gold standard. Figure 6 illustrates this workflow in more detail.

Algorithm 1 Synthetic Enterprise-State Generation

Require: Number of high-level goals N_G, Maximum lineage depth $H_{\max}$, Thresholds λ_1, λ_2, LLM $\mathcal{M}$ called using GETONETWORKACTIVITY, SAMPLEGOALOWNER, SAMPLEGOALDELIVERABLE, DERIVEUPSTREAMDEPENDENCIES, and GENERATEDELIVERABLE
Ensure: Synthetic enterprise state $\mathcal{S} = (\mathcal{G}, \mathcal{O})$

$\mathcal{G} := (\mathcal{V}, \mathcal{E}) \leftarrow (\varnothing, \varnothing)$
$\mathcal{O} \leftarrow \varnothing$
for $i = 1$ to N_G **do**
  $o \leftarrow$ SAMPLEGOALOWNER$(\mathcal{S})$
  $g \leftarrow$ SAMPLEGOALDELIVERABLE$(o, \mathcal{S})$
  if ISGOALREALISTIC$()g, o, \lambda_1, \lambda_2)$ **then**
    Add node: $\mathcal{O}(g) \leftarrow o$, $\mathcal{V} \leftarrow \mathcal{V} \cup \{g\}$
    $(\mathcal{G}, \mathcal{O}) \leftarrow$ BACKWARDEXPAND$(g, o, 0, H_{\max}, \mathcal{G}, \mathcal{O})$
  else
    return $((\varnothing, \varnothing), \varnothing)$
  end if
end for
$\mathcal{A} \leftarrow$ FORWARDMATERIALIZE$(\mathcal{G}, \mathcal{O})$
return Materialized enterprise state $\mathcal{S}^* = (\mathcal{G}, \mathcal{O}, \mathcal{A})$

procedure ISGOALREALISTIC$(g, o, \lambda_1, \lambda_2)$
  ▷ *Check if the sampled goal is realistic for the persona based on the telemetry.*
  Get O*Net work activity for the goal $g_{wa} \leftarrow$ GETONETWORKACTIVITY$()g)$
  if (Labor statistics expectation of g_{wa} for $o \geq \lambda_1$) and (Claude usage for $g_{wa} \geq \lambda_2$) **then**
    return True
  end if
  return False
end procedure

procedure BACKWARDEXPAND$(v, p, h, H_{\max}, \mathcal{G}, \mathcal{O})$
  ▷ *Recursively constructs the upstream lineage rooted at v. Expansion terminates when v is a source artifact or when the current depth reaches $H_{\max}$.*
  if $h \geq H_{\max}$ or ISSOURCE$(v, p, \mathcal{S})$ **then**
    return $(\mathcal{G}, \mathcal{O})$
  end if
  $\mathcal{R}_v \leftarrow$ DERIVEUPSTREAMDEPENDENCIES(p, v)
  for all dependency-owner pairs $(u, p_u) \in \mathcal{R}_v$ **do**
    if $u \in \mathcal{V}$ **then**
      Add edge: $\mathcal{E} \leftarrow \mathcal{E} \cup \{(u, v)\}$
    else
      Add node: $\mathcal{V} \leftarrow \mathcal{V} \cup \{u\}$, $\mathcal{O}(u) \leftarrow p_u$; and add edge $\mathcal{E} \leftarrow \mathcal{E} \cup \{(u, v)\}$
      $(\mathcal{G}, \mathcal{O}) \leftarrow$ BACKWARDEXPAND$(u, p_r, h + 1, h_{max}, \mathcal{G}, \mathcal{O})$
    end if
  end for
  return $(\mathcal{G}, \mathcal{O})$
end procedure

procedure FORWARDMATERIALIZE$(\mathcal{G}, \mathcal{O})$
  ▷ *Materializes each node into its deliverable in topological order, such that its owner generates it using only its materialized upstream dependencies.*
  $\mathcal{T} \leftarrow$ TOPOLOGICALSORT$(\mathcal{G})$
  $\forall v \in \mathcal{T} : \mathcal{A}(v) \leftarrow$ GENERATEDELIVERABLE$(v, \mathcal{O}(v), \{\mathcal{A}(u) : u \in \mathcal{U}(v)\})$
  return $\mathcal{A}$
end procedure

Table 8: The 35 evaluated enterprises: 30 industry-balanced core organizations (with O*NET major-group industry) followed by the 5 held-out extension organizations. Each is seeded from one occupation; *IWA* is the most frequent Intermediate Work Activity across the organization's five goals; #Art. and #Der. are the total and derived (non-source) artifact counts in the lineage graph.

Occupation (O*NET)	SOC	Industry	Intermediate work activity (IWA)	#Art.	#Der.
Automotive Engineering Technicians	17-3027.00	Arch/Eng	Test characteristics of materials or pro…	107	38
Electronics Engineers, Except Computer	17-2072.00	Arch/Eng	Advise others on business or operational…	201	79
Graphic Designers	27-1024.00	Arts/Media	Create visual designs or displays.	156	54
Special Effects Artists and Animators	27-1014.00	Arts/Media	Create visual designs or displays.	250	71
Cost Estimators	13-1051.00	Bus/Fin	Determine values or prices of goods or s…	58	21
Farm Labor Contractors	13-1074.00	Bus/Fin	Supervise personnel activities.	153	49
Logistics Analysts	13-1081.02	Bus/Fin	Analyze data to improve operations.	54	18
Child, Family, and School Social Workers	21-1021.00	Comm/Soc	Present research or technical information.	152	48
Probation Officers and Correctional Treatment Specialists	21-1092.00	Comm/Soc	Present research or technical information.	115	42
Database Architects	15-1243.00	Comp/Math	Design databases.	55	19
Software Quality Assurance Analysts and Testers	15-1253.00	Comp/Math	Test performance of computer or informat…	58	20
Special Education Teachers, Middle School	25-2057.00	Education	Develop educational programs, plans, or …	54	19
Teaching Assistants, Postsecondary	25-9044.00	Education	Assess student capabilities, needs, or p…	250	76
Medical Dosimetrists	29-2036.00	Healthcare	Analyze health or medical data.	57	20
Occupational Therapists	29-1122.00	Healthcare	Advise patients or clients on medical is…	152	44
Paralegals and Legal Assistants	23-1011.00	Legal	Prepare legal or regulatory documents.	250	73
Title Examiners, Abstractors, and Searchers	23-2093.00	Legal	Prepare legal or regulatory documents.	95	33
Epidemiologists	19-1041.00	Life Sci	Analyze health or medical data.	152	51
Political Scientists	19-3094.00	Life Sci	Analyze data to improve operations.	57	22
Sociologists	19-3041.00	Life Sci	Research historical or social issues.	54	18
Advertising and Promotions Managers	11-2011.00	Management	Develop business or marketing plans.	202	61
Fundraising Managers	11-2033.00	Management	Develop financial or business plans.	201	59
Marketing Managers	11-2021.00	Management	Develop business or marketing plans.	54	19
Postmasters and Mail Superintendents	11-9131.00	Management	Direct organizational operations, activi…	55	19
Quality Control Systems Managers	11-3051.01	Management	Monitor operations to ensure adequate pe…	54	19
Desktop Publishers	43-9031.00	Office/Admin	Process digital or online data.	250	74
Statistical Assistants	43-9111.00	Office/Admin	Analyze data to improve operations.	54	19
Word Processors and Typists	43-9031.00	Office/Admin	Process digital or online data.	250	72
Real Estate Brokers	41-9021.00	Sales	Maintain current knowledge in area of ex…	250	75
Sales Representatives, Wholesale and Manufacturing, Except Technical and Scientific Products	41-4012.00	Sales	Advise others on products or services.	202	64

C.4 AGENT HARNESS

Every artifact, whether a materialized reference or a candidate answer, is produced by the same kind of *sandboxed office agent*. The model runs as a multi-turn, tool-using agent inside an isolated container equipped with a Python runtime and standard spreadsheet, document, and presentation libraries. On each turn the agent may inspect the provided input files, run code to analyze or transform their contents, and write output; it must ultimately save a real Office file (`.xlsx`, `.docx`, or `.pptx`) rather than merely describe one.

The task is delivered on the first turn as a single first-person request from the artifact's owner (prompt A.5, Appendix E): the owner's *name*, *title*, *team*, and O*NET-derived *responsibilities*, followed by the artifact's name and description, the list of upstream input files already staged in the working directory, a grounding rule that constrains every concrete fact to those inputs, and the target output path. This owner context is exactly the role information $\mathcal{O}(t)$ used in the task definition—and it is the *only* persona detail the agent sees: the richer O*NET descriptors (knowledge, skills, and detailed work activities in Prompts F.1–F.3) are used solely to sample goals and compose the respon-

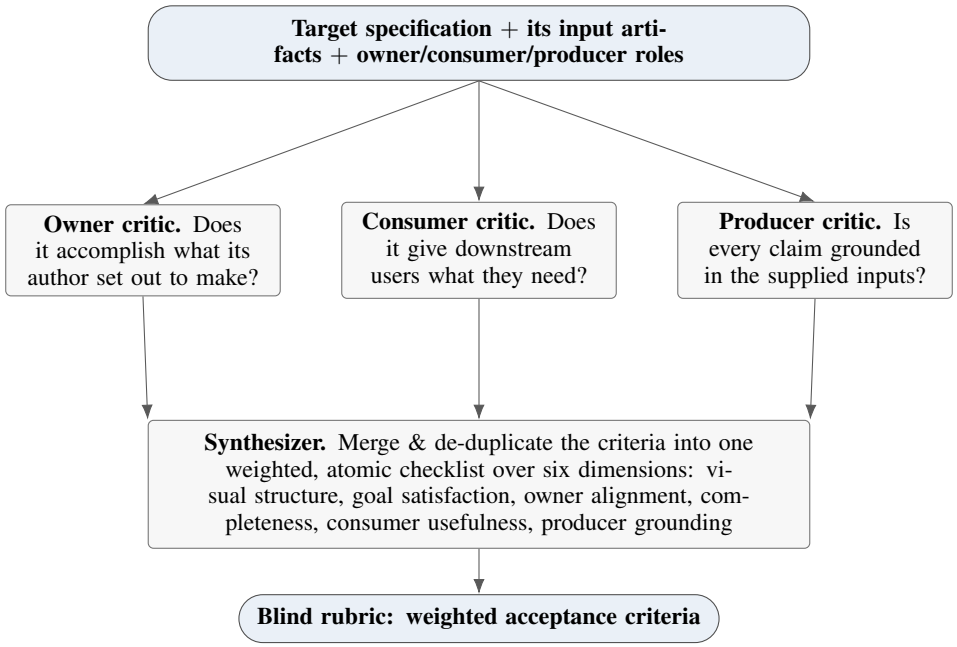


Figure 5: **Rubric construction.** Three stakeholder critics independently propose acceptance criteria from the target and its inputs alone; a synthesizer consolidates them into one weighted rubric. The rubric is fixed before any answer is produced, so all models face the same bar.

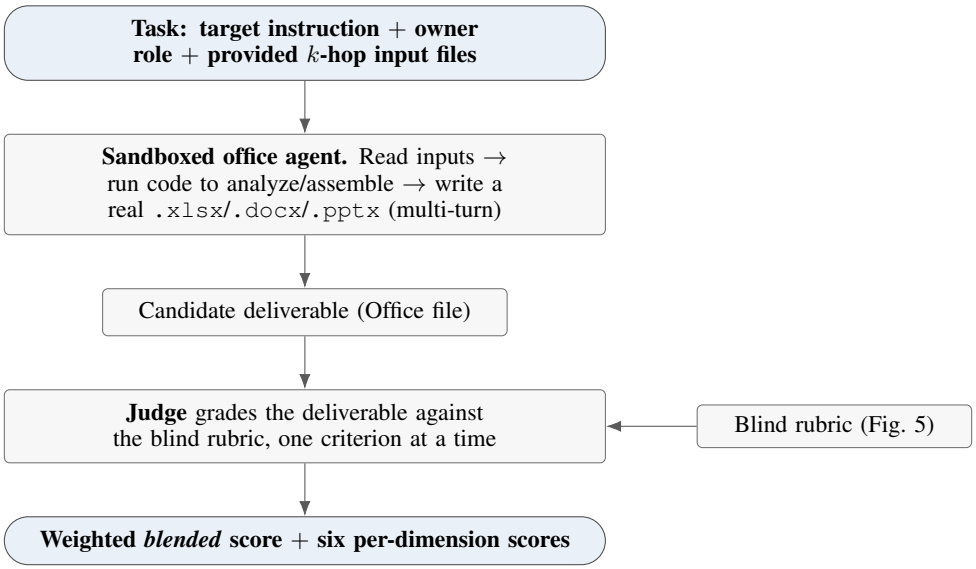


Figure 6: **Authoring and scoring.** A generator produces a real Office file from the provided inputs in the sandboxed agent; each judge then scores that file against the task's blind rubric, yielding a blended score and per-dimension scores.

sibilities during generation, and are *not* handed to the authoring agent. The system prompt carries no persona; it only establishes the tool-using office-agent role and the persistent-kernel note. The same persona and inputs are given identically to the reference materialization and to every candidate, so the only thing that varies across the leaderboard is the model.

The environment is capped at 200 turns and a per-call token budget, with reasoning effort set to its highest setting; agents that exhaust the budget are recorded as failures. Judges operate as single-call

graders over the produced file's rendered contents and the task's rubric. This uniform harness means the reference artifacts and the candidate answers are produced under the same conditions, so quality differences reflect the models rather than the tooling.

Representation of the inputs. The authoring agent and the judges receive artifacts in *different* formats. The *authoring agent* works on the *raw editable files*: its upstream inputs are staged in the working directory as real `.xlsx`/`.docx`/`.pptx` files that it opens and manipulates programmatically (reading exact cell values, formulas, and text), and it likewise saves a real Office file. The *judges* on the other hand do not see the raw file for efficiency since they are all big, powerful, and expensive models. Each deliverable is instead rendered once into a *dual representation*, and both parts are sent to the grader: (i) a *visual* view—the file is converted with headless LibreOffice to PDF and then to one page-image (PNG) per sheet, slide, or page, including layout, formatting, charts, (ii) a *textual* view—a flattened text/Markdown extraction of the file's contents (cell values and formulas, body text, speaker notes) that preserves the exact numbers and strings which may be illegible or clipped in the image. Grading the two views jointly lets the rubric check both presentation (*visual_structure*) and exact, grounded content (*completeness*, *producer_grounding*) that a screenshot alone would miss. Each judge grades in a single call, and the identical dual rendering is used for every judge and every candidate. This uniform harness means the reference artifacts and the candidate answers are produced under the same conditions, so quality differences reflect the models rather than the tooling.

C.5 Final corpus

The 30 selected organizations contain **4,052** artifacts – **2,756** primary sources (raw data the owners originate) and **1,296** derived deliverables (authored, then trace-grounded for fabrication removal). The result is a corpus that is simultaneously (i) *grounded* in real O*NET occupations and their work activities, (ii) *usage-relevant* (every goal clears the Claude-usage and labor-market floors), and (iii) *industry-diverse* (all 12 major groups represented).

D Additional Experimental Results

D.1 Per-judge leaderboards

Table 2 reports the averaged score over three judges per target. Because each grader is itself an LLM, we also give the single-judge leaderboards on the identical 236 common targets to show how each judge behaves—Claude Opus 4.8 (Table 9), Claude Sonnet 5 (Table 10) and GPT-6 (Table 11). We note that the absolute calibration differs slightly as GPT-6 grades every model ~ 0.15–0.20 lower than either Claude judge. Second, and more consequentially, the judges disagree at the *top* of the board. Opus 4.8 ranks Grok-4.5 first and Grok-4.6 eighth, but both Sonnet 5 and GPT-6 rank Grok-4.6 *first*. Specifically, we note difference where its register-sparse workbook style is penalized by Opus 4.8 to near-zero but is scored far more leniently by the other two judges. The middle and lower tiers (GPT-5.6 family, Nemotron, Haiku, and the older GPT-5.4/5.5) keep a stable relative order under all three judges. This top-of-board judge-dependence is the main reason a single-judge leaderboard can be misleading, and is precisely why the main-text blended score averages across the three judges.

D.2 How much do the judges agree?

In addition, we also test the reliability of the judges by noting their direct correlations Table 12 reports two complementary views. At the *item* level—agreement on individual artifacts, pairwise Pearson correlations lie between 0.72 and 0.79 and Spearman between 0.72 and 0.79. When we pool across all three judges, we get an intraclass correlation $\mathrm{ICC}(2,1) = 0.61$ and Krippendorff's $\alpha = 0.57$ (interval), i.e. moderate-to-substantial absolute agreement on a continuous $[0,1]$ scale. The residual disagreement is dominated by *calibration*, not ordering: the mean absolute score difference is only 0.11 between the two Claude judges but 0.22 against the stricter GPT-6, consistent with the offset visible in Tables 10 and 11. Once scores are aggregated to per-agent means, this item-level noise averages out and the judges' *rankings* converge sharply as model-level Spearman 0.75–0.92, which means that the leaderboard ordering is robust to the choice of judge even though absolute calibration may not be.

Table 9: Blended and per-dimension scores at $\tau_1(t)$ under the **Claude Opus 4.8** judge, on the same 236 common targets as Table 2. Each cell is the mean with SD as a gray subscript; each model is averaged over its own coverage of S under this judge (225–236 targets per model). Best per column in **bold**.

Generator	VS	GS	OA	CP	CU	PG	Bl.
Grok-4.5	**0.926**$_{0.18}$	**0.858**$_{0.23}$	**0.882**$_{0.20}$	**0.872**$_{0.22}$	**0.878**$_{0.21}$	**0.896**$_{0.16}$	**0.881**$_{0.16}$
Claude Opus 4.6	0.905$_{0.18}$	0.830$_{0.23}$	0.810$_{0.23}$	0.854$_{0.19}$	0.851$_{0.20}$	0.839$_{0.16}$	0.836$_{0.15}$
Claude Opus 4.8	0.910$_{0.16}$	0.815$_{0.22}$	0.803$_{0.22}$	0.827$_{0.20}$	0.852$_{0.22}$	0.819$_{0.16}$	0.819$_{0.15}$
GPT-5.6-nano	0.904$_{0.16}$	0.773$_{0.24}$	0.808$_{0.22}$	0.791$_{0.22}$	0.803$_{0.24}$	0.828$_{0.14}$	0.807$_{0.14}$
GPT-5.6	0.907$_{0.17}$	0.774$_{0.23}$	0.786$_{0.24}$	0.795$_{0.21}$	0.824$_{0.23}$	0.795$_{0.16}$	0.791$_{0.15}$
Reference	0.908$_{0.16}$	0.748$_{0.26}$	0.784$_{0.23}$	0.778$_{0.22}$	0.805$_{0.24}$	0.799$_{0.15}$	0.784$_{0.14}$
Claude Sonnet 4.6	0.844$_{0.29}$	0.731$_{0.34}$	0.774$_{0.30}$	0.759$_{0.32}$	0.770$_{0.33}$	0.801$_{0.25}$	0.776$_{0.26}$
Grok-4.6	0.782$_{0.33}$	0.713$_{0.35}$	0.771$_{0.31}$	0.727$_{0.35}$	0.695$_{0.37}$	0.789$_{0.26}$	0.753$_{0.28}$
GPT-5.6-mini	0.876$_{0.19}$	0.718$_{0.26}$	0.734$_{0.24}$	0.759$_{0.23}$	0.759$_{0.25}$	0.761$_{0.15}$	0.749$_{0.14}$
Nemotron-3-Ultra	0.794$_{0.28}$	0.675$_{0.33}$	0.677$_{0.28}$	0.706$_{0.29}$	0.750$_{0.30}$	0.697$_{0.22}$	0.698$_{0.22}$
Claude Haiku 4.5	0.726$_{0.27}$	0.625$_{0.29}$	0.631$_{0.30}$	0.659$_{0.25}$	0.676$_{0.27}$	0.616$_{0.21}$	0.629$_{0.19}$
GPT-5.4	0.565$_{0.36}$	0.318$_{0.30}$	0.452$_{0.31}$	0.348$_{0.30}$	0.382$_{0.33}$	0.478$_{0.23}$	0.421$_{0.20}$
GPT-5.5	0.555$_{0.35}$	0.313$_{0.30}$	0.447$_{0.31}$	0.353$_{0.30}$	0.372$_{0.33}$	0.476$_{0.22}$	0.417$_{0.20}$

Table 10: Blended and per-dimension scores at $\tau_1(t)$ under the **Claude Sonnet 5** judge, on the same 236 common targets as Table 2. Each cell is the mean with SD as a gray subscript. Best per column in **bold**.

Generator	VS	GS	OA	CP	CU	PG	Bl.
Grok-4.6	0.819$_{0.29}$	**0.733**$_{0.32}$	**0.808**$_{0.26}$	**0.737**$_{0.33}$	0.696$_{0.34}$	**0.820**$_{0.21}$	**0.776**$_{0.22}$
Grok-4.5	**0.841**$_{0.26}$	0.731$_{0.29}$	0.777$_{0.26}$	0.725$_{0.30}$	**0.723**$_{0.30}$	0.813$_{0.20}$	0.773$_{0.20}$
Claude Opus 4.6	0.815$_{0.26}$	0.690$_{0.30}$	0.687$_{0.29}$	0.704$_{0.29}$	0.709$_{0.30}$	0.757$_{0.20}$	0.723$_{0.21}$
Claude Opus 4.8	0.834$_{0.23}$	0.676$_{0.28}$	0.694$_{0.27}$	0.680$_{0.27}$	0.702$_{0.30}$	0.735$_{0.19}$	0.709$_{0.18}$
Reference	0.829$_{0.24}$	0.665$_{0.28}$	0.697$_{0.27}$	0.674$_{0.26}$	0.668$_{0.31}$	0.735$_{0.18}$	0.703$_{0.17}$
GPT-5.6-nano	0.797$_{0.26}$	0.632$_{0.30}$	0.672$_{0.29}$	0.653$_{0.29}$	0.636$_{0.31}$	0.734$_{0.20}$	0.688$_{0.19}$
GPT-5.6	0.829$_{0.24}$	0.646$_{0.28}$	0.671$_{0.29}$	0.679$_{0.28}$	0.662$_{0.31}$	0.711$_{0.19}$	0.686$_{0.18}$
GPT-5.6-mini	0.797$_{0.26}$	0.599$_{0.29}$	0.627$_{0.28}$	0.632$_{0.28}$	0.613$_{0.31}$	0.685$_{0.20}$	0.650$_{0.18}$
Claude Sonnet 4.6	0.734$_{0.33}$	0.563$_{0.34}$	0.617$_{0.32}$	0.596$_{0.34}$	0.616$_{0.35}$	0.691$_{0.26}$	0.641$_{0.26}$
Nemotron-3-Ultra	0.628$_{0.36}$	0.517$_{0.35}$	0.538$_{0.32}$	0.524$_{0.32}$	0.590$_{0.36}$	0.609$_{0.24}$	0.568$_{0.23}$
Claude Haiku 4.5	0.650$_{0.31}$	0.516$_{0.30}$	0.519$_{0.30}$	0.538$_{0.27}$	0.589$_{0.30}$	0.580$_{0.20}$	0.558$_{0.18}$
GPT-5.5	0.510$_{0.36}$	0.296$_{0.29}$	0.432$_{0.32}$	0.327$_{0.29}$	0.352$_{0.32}$	0.475$_{0.22}$	0.405$_{0.19}$
GPT-5.4	0.531$_{0.36}$	0.303$_{0.29}$	0.426$_{0.32}$	0.319$_{0.28}$	0.344$_{0.33}$	0.473$_{0.23}$	0.404$_{0.19}$

D.3 How does the performance vary at different input lineage depths?

Section 4 (Figure 2) summarizes how quality degrades as the input frontier recedes from the immediate parents (τ_1) to 5-hop ancestors (τ_5) at the same set of targets. Because the initial set of targets were randomly sampled and may not be necessarily deep enough to be evaluated at each successive level, the common set of targets were few. We extend this analysis to more agents and targets by instead looking how the performance varies at different input lineage depths even when the targets may be different. Figure 7 shows the trend for every model and every depth k, where the blended score is averaged over exactly those targets that admit a k-hop frontier for that model. So, the number of targets at each depth falls from $\sim$236 at τ_1 to 12–38 at τ_5 (Grok-4.5/4.6 and the GPT-5.6-luna/terra variants are run on only a subset of the deep levels, with $n \leq 16$ at τ_4/τ_5). Thus, these estimates are indicative rather than strictly comparable across models. Yet they reveal consistent and universal patterns that the performance degrades as the lineage depth is increased for a task. GPT-5.6-sol is found to be the most depth-robust ($0.689 \to 0.532$, $\Delta = -0.157$, highest at every $k \geq 2$), while both the Opus and Grok models drop the hardest (Δ between -0.24 and -0.26). Other Claude models (Sonnet 4.6 and Haiku 4.5) fall off fastest (to 0.411 and 0.296), while we see that by τ_3, the GPT-5.6 family already overtakes the Grok/Opus models that are found to lead at τ_1.

Table 11: Blended and per-dimension scores at $\tau_1(t)$ under the **GPT-6** judge, on the same 236 common targets as Table 2. Each cell is the mean with SD as a gray subscript. Best per column in **bold**.

Generator	VS	GS	OA	CP	CU	PG	Bl.
Grok-4.6	**0.736**$_{0.30}$	**0.620**$_{0.30}$	**0.714**$_{0.29}$	**0.651**$_{0.29}$	**0.590**$_{0.31}$	**0.770**$_{0.18}$	**0.704**$_{0.18}$
Grok-4.5	0.656$_{0.32}$	0.560$_{0.29}$	0.644$_{0.29}$	0.570$_{0.30}$	0.536$_{0.32}$	0.695$_{0.19}$	0.636$_{0.18}$
GPT-5.6-nano	0.637$_{0.29}$	0.530$_{0.28}$	0.587$_{0.28}$	0.560$_{0.26}$	0.518$_{0.30}$	0.657$_{0.17}$	0.603$_{0.15}$
GPT-5.6	0.687$_{0.29}$	0.527$_{0.29}$	0.588$_{0.27}$	0.582$_{0.24}$	0.554$_{0.31}$	0.635$_{0.17}$	0.599$_{0.15}$
Reference	0.696$_{0.27}$	0.527$_{0.29}$	0.566$_{0.28}$	0.553$_{0.26}$	0.519$_{0.32}$	0.623$_{0.17}$	0.582$_{0.14}$
GPT-5.6-mini	0.625$_{0.31}$	0.496$_{0.28}$	0.522$_{0.26}$	0.516$_{0.28}$	0.493$_{0.30}$	0.600$_{0.17}$	0.555$_{0.15}$
Claude Opus 4.6	0.583$_{0.30}$	0.488$_{0.29}$	0.554$_{0.29}$	0.546$_{0.28}$	0.487$_{0.29}$	0.571$_{0.19}$	0.545$_{0.17}$
Claude Opus 4.8	0.579$_{0.31}$	0.499$_{0.29}$	0.528$_{0.28}$	0.511$_{0.27}$	0.466$_{0.31}$	0.566$_{0.20}$	0.537$_{0.16}$
Claude Sonnet 4.6	0.531$_{0.34}$	0.453$_{0.32}$	0.515$_{0.31}$	0.505$_{0.30}$	0.439$_{0.33}$	0.568$_{0.22}$	0.525$_{0.21}$
Nemotron-3-Ultra	0.482$_{0.33}$	0.385$_{0.30}$	0.418$_{0.30}$	0.448$_{0.30}$	0.445$_{0.32}$	0.459$_{0.20}$	0.442$_{0.18}$
Claude Haiku 4.5	0.395$_{0.29}$	0.350$_{0.27}$	0.357$_{0.28}$	0.391$_{0.26}$	0.365$_{0.31}$	0.371$_{0.20}$	0.371$_{0.17}$
GPT-5.4	0.326$_{0.33}$	0.223$_{0.24}$	0.325$_{0.30}$	0.248$_{0.25}$	0.221$_{0.28}$	0.412$_{0.20}$	0.327$_{0.15}$
GPT-5.5	0.331$_{0.33}$	0.220$_{0.23}$	0.324$_{0.30}$	0.254$_{0.25}$	0.219$_{0.27}$	0.409$_{0.20}$	0.327$_{0.15}$

	Item-level			**Model-level**
Judge pair	**Pearson** r	**Spearman** ρ	**MAD**	**Spearman** ρ
Opus 4.8 – Sonnet 5	0.782	0.760	0.111	0.917
Opus 4.8 – GPT-6	0.786	0.788	0.220	0.750
Sonnet 5 – GPT-6	0.719	0.720	0.169	0.833

Table 12: Inter-judge agreement at $\tau_1(t)$. *Item-level* columns are computed over all the (target, agent) instances scored by all three judges (Pearson r, Spearman ρ, and mean absolute score difference MAD); *Model-level* ρ is the Spearman correlation of per-agent mean scores. Pooling all three judges yields $\mathrm{ICC}(2,1) = 0.61$ and interval Krippendorff's $\alpha = 0.57$.

D.4 Cost of generation

Table 13 reports the per-task API cost of authoring a τ_1 deliverable, measured directly from the agent trajectories on a subset of 40 tasks. Cost is dominated by *input* tokens: because the sandboxed agent re-sends its growing context on every turn and we apply no prompt-cache discount for consistency. Thus, a single task streams 0.3–1.8M input tokens, so the figures are an *upper bound* on the real bill. Under public list prices, the two Opus configurations cost $\sim$\$20 per deliverable, which is two orders of magnitude more than the GPT-5.6 family (\$0.07–\$1.00), while Sonnet 4.6 sits in between at \$6. This cost axis is orthogonal to the quality as the GPT-5.6 family (Section 4) reach top-tier scores at a small fraction of the Opus spend. We omit Grok-4.5 and Nemotron-3-Ultra from this analysis as their logs could not be logged properly on our resources.

D.5 Human annotation

Since both the rubric and the grades are produced by LLMs, the authors annotated that the blind rubric captures *task-relevant* criteria rather than generic or off-task requirements. Five annotators each reviewed a set of tasks, where for every task they were shown the owner's request and 8 candidate acceptance criteria in shuffled order—4 drawn from that task's own rubric ("belongs") and 4 *distractors* sampled from *other* tasks' rubrics within the *same* rubric dimension. Then, they were asked to rank them by how well each fits the assignment. Since the distractors are real, well-formed criteria from the same dimension (e.g. another task's producer-grounding check), a rater can only separate them by genuine task relevance, not by surface style.

We score each rater by the pairwise probability that a real criterion is ranked above a distractor (an AUC; chance $= 0.5$). Agreement is high and consistent (Table 14): the mean AUC is $\mathbf{0.86}$ over 315 real–distractor comparisons, ranging 0.69–1.00 across the five annotators, with every annotator well above chance. Human raters thus reliably recognize the rubric's criteria as belonging to their task

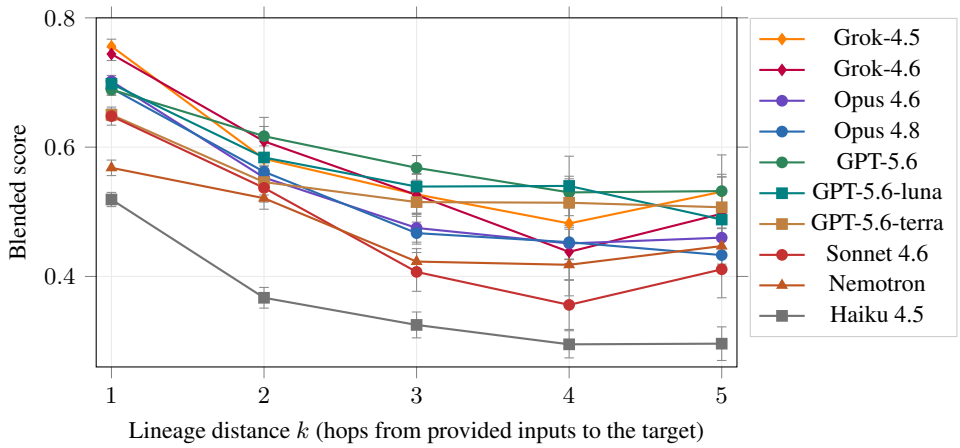


Figure 7: **Quality vs. lineage depth.** Mean blended score (±SEM) as the input frontier recedes from the immediate parents (τ_1) to 5-hop ancestors (τ_5). Each point is averaged over all the targets that reach that depth for that model.

Table 13: Per-task authoring cost on the 40-task subset, under public list prices (per 1M tokens, input/output): Opus \$15/\$75, Sonnet \$3/\$15, Haiku \$0.80/\$4, GPT-5.6 sol \$2/\$10, terra \$0.50/\$2, luna \$0.10/\$0.50. Input tokens are the full context re-sent each turn (no prompt-cache discount), so cost/task is an upper bound.

Generator	Input (M tok)	Output (K tok)	Cost/task ($)
Claude Opus 4.6	1.25	31.9	21.10
Claude Opus 4.8	1.23	28.4	20.55
Claude Sonnet 4.6	1.80	38.9	5.99
GPT-5.6-sol	0.39	21.2	1.00
Claude Haiku 4.5	0.52	22.6	0.51
GPT-5.6-luna	0.28	17.4	0.17
GPT-5.6-terra	0.55	26.0	0.07

and reject plausible-but-off-task distractors, indicating that the blind, stakeholder-authored rubric produces *relevant*, task-grounded acceptance criteria rather than boilerplate.

E Prompts and Workflow

This appendix lists the prompts that drive each stage of the benchmark pipeline: (i) **backward expansion** (the owner of an artifact names the direct upstream inputs it is built from, recursively, and a two-tier deduplicator reuses shared nodes so the lineage forms a DAG); (ii) **artifact authoring** (an agent writes each deliverable as a real Office file from its upstream inputs); (iii) **trace-grounded fabrication removal** (an auditor inspects the agent's tool-call trace and feeds invented facts back for re-authoring); (iv) **blind multi-stakeholder rubric generation** (owner/consumer/producer critics author an acceptance rubric *before* seeing any candidate, then a synthesizer consolidates it); and (v) **candidate scoring** against that rubric. Backward expansion and authoring use the `gpt-5.6` generation backbone; all judge/critic calls use the `claude-opus-4.8` model.

E.1 Backward expansion (dependency generation)

Generation starts from a sampled goal and works *backward*: the owning persona is asked, one artifact at a time, for the DIRECT upstream inputs required to produce it and who owns each; every answer appends nodes and edges to a mutable lineage graph, and each newly discovered dependency is expanded in turn until the chain reaches raw `dataset` sources or a depth bound (Appendix C.1). To keep the graph a *DAG* rather than a tree of duplicated inputs, the *same* underlying deliverable

Table 14: Human validation of rubric relevance. Each annotator ranked 4 genuine ("belongs") criteria against 4 same-dimension distractors per task; the score is the pairwise AUC that a genuine criterion is ranked above a distractor (chance = 0.5).

Annotator	Ranking AUC	# real–distractor pairs
A1	0.688	80
A2	0.949	59
A3	0.912	80
A4	1.000	16
A5	0.762	80
Mean	**0.862**	315

named across different branches or goals (e.g. a shared "FY25 Revenue Actuals" workbook feeding several decks) is deduplicated onto one canonical node in two tiers.

(1) Deterministic exact match. Before any model call, each proposed dependency is matched against every existing artifact by *normalized name* (lower-cased, whitespace-collapsed) + *type* (optionally ignoring owner, so the same substance owned by a slightly different role still converges). On a hit the existing node is reused: a converging edge is added and the node is *not* re-expanded (it was already expanded when first discovered). A cycle guard drops the new edge if reusing that node would close a cycle. This resolves all verbatim and whitespace/case duplicates without spending a token.

(2) LLM semantic match. When *growing* an existing organization (adding new goals onto an already-materialized graph), an exact match is tried first; if none is found, an LLM matcher decides whether the needed input is *essentially the same artifact* as one already in the catalog—the same underlying data/report/model, even if owned by a different person or team—returning the canonical `id` to reuse or `null` to create a fresh node.

E.1.1 Dependency expansion — system

```
BACKWARD_EXPANSION. You role-play the owner of a work artifact and
  identify the DIRECT upstream inputs required to produce it. Only
  list first-order dependencies (not transitive ones). Prefer
  terminating at raw 'dataset' sources when the input is simply raw
  data. Assign each dependency to the team/role that would own it.
  Choose a realistic app FORMAT for each input and favor a DIVERSE
  mix across the organization: Excel workbooks/models, Word
  documents/memos, PowerPoint decks, BI dashboards, raw datasets,
and approvals -- pick whatever a real owner would use.
```

E.1.2 Dependency ask — user (one call per artifact being expanded)

```
depth=<DEPTH>
You are <OWNER NAME>, <OWNER TITLE> (<OWNER ROLE>).
You must produce this artifact:
  name: <ARTIFACT NAME>
  type: <excel|word|powerpoint|dashboard|dataset|approval>
  description: <ARTIFACT DESCRIPTION>

List the direct upstream inputs you need. Return a JSON object:
{ "dependencies": [ { "name": ...,
    "format": "excel|word|powerpoint|dashboard|dataset|approval",
    "app": "outlook|teams|excel|word|powerpoint|sharepoint|planner
   |power_automate|web|other",
    "description": ..., "owner_role": ..., "rationale": ... } ] }
```

```
Use 'dataset' only for raw/atomic source data. Prefer a realistic MIX
  of formats (not everything the same). Return an empty list if this
  artifact is itself a raw source
needing no inputs.
```

E.1.3 Dependency matcher (deduplication) — system

```
DEPENDENCY_MATCH. Decide whether a needed input is ALREADY provided by
  one of the org's existing artifacts -- the SAME underlying
  data/report/model/deliverable -- EVEN IF owned by a different
  person/team. Reuse liberally when substance matches; return null
  only when nothing provides it.
```

E.1.4 Dependency matcher (deduplication) — user (one call per proposed input)

```
Needed input:
  name: <DEPENDENCY NAME>
  type: <excel|word|powerpoint|dashboard|dataset|...>
  description: <DEPENDENCY DESCRIPTION>

Catalog:
[ { "id": ..., "name": ..., "type": ..., "owner": <owner title>,
    "desc": <first 120 chars of description> }, ... ]   # most-recent
  140 artifacts

Return {"match_id": "<id or null>", "reason": "..."}.
% <STRICT GUIDE (E.1.5), calibrated grower only>
```

E.2 Artifact authoring

The authoring agent runs a multi-turn `run_python`/`run_bash` loop. Its system prompt frames it as the owner's assistant; a grounding rule constrains every concrete fact to the upstream inputs; the user turn is the owner's first-person request.

E.2.1 System prompt (owner-voiced authoring agent)

```
You are an expert Office-document creation agent on a host machine
  with a Python toolchain. A colleague will ask you -- in their own
  words -- to build a work deliverable from their data. Your output
  is a REAL Office file, not a chat message.

TOOLS (call them -- do not merely describe code):
  run_python(code)  execute Python 3; openpyxl, python-docx,
  python-pptx, pandas installed.
  run_bash(command) run a shell command in the working directory.
<STATE NOTE>
Their source files are already in your CURRENT WORKING DIRECTORY. Read
  them first, then BUILD the <SURFACE> programmatically and SAVE it
  to the exact target path in the cwd; re-open it with run_python to
  verify structure before finishing.

Build what they'd actually ship: a complete, professional <SURFACE>
  with real structure, clear sections, and TABLES for tabular/numeric
  content -- not a stub.
```

E.2.2 State note (inserted for the persistent-kernel harness)

```
STATE PERSISTS: run_python runs in a PERSISTENT interpreter (like a
  notebook) -- variables, imports, and DataFrames you define in one
  call stay available in later calls, so you can build up the file
  incrementally across calls. Just make sure the final file saved to
  disk is complete before finishing.
```

E.2.3 Grounding rule (derived artifact)

```
One hard rule: every concrete fact -- numbers, $ amounts, %, dates,
  IDs, named
entities -- has to come straight from the input files in the working
  directory (copy them, aggregate them, or compute them from those
  inputs). Please don't introduce any figure, date, or name that
  isn't in them or derivable from them. You're very welcome to add
  the structure, narrative, and recommendations I'd expect on top of
  that data.
```

E.2.4 Grounding rule (primary source, no upstream)

```
There's no upstream for this one -- it's the raw data I own, so
  populate it with realistic, concrete, internally-consistent records
  with generous volume (lots of rows/ detailed records), the way my
  real system would actually generate it.
```

E.2.5 Authoring request (owner's first-person user turn)

```
Hi -- I'm <OWNER NAME>, <OWNER TITLE> on the <TEAM> team. A big part
  of my role is
<RESPONSIBILITIES>.

I need you to put together a <SURFACE> for me: **<ARTIFACT NAME>**.

Here's what I'm after:
<ARTIFACT DESCRIPTION>

I've dropped the source files I actually work from into the working
  directory --
please read them first:
  - <UPSTREAM INPUT 1>
  - <UPSTREAM INPUT 2>
  - ...

<GROUNDING RULE (E.2.3 or E.2.4)>

Please build it as a real <SURFACE> file and save it to <TARGET PATH>.
  It must be a <SURFACE> file even if the title says 'workbook',
  'deck', 'memo' or similar -- the <SURFACE> format is what I need.
  Build it by calling run_python with
  openpyxl/python-docx/python-pptx code that writes it to <TARGET
  PATH> (relative to the cwd); re-open it with run_python to
  double-check. Once it's there and reads like something I'd be happy
  to put my name on, we're done.
```

E.3 TRACE-GROUNDED FABRICATION REMOVAL

After authoring, the full tool-call trace (files read + code run) is saved. An auditor reads *only* the trace and lists concrete facts written into the deliverable that were never read or computed. Those are fed back into the authoring prompt, which re-authors; the loop repeats until the trace shows no invented facts.

E.3.1 Trace-only fabrication audit

```
TRACE-GROUNDED FABRICATION AUDIT. Below is the COMPLETE tool-call
  trace of an agent that authored the deliverable '<ARTIFACT NAME>':
  every input file it READ (tool outputs show the data actually
  available) and the code it RAN to build the file. Using ONLY this
  trace, list every CONCRETE fact (numbers, $ amounts, dates, IDs,
  named entities) the agent WROTE INTO the deliverable that does NOT
  appear in any input it read and is NOT computed from that read data
  -- i.e. it was invented/hardcoded. Ignore narrative and
  recommendations; flag only unsupported concrete facts.

AGENT TRACE:
<TOOL-CALL TRACE DIGEST>

Return JSON only: {"fabricated":["<invented value + why it is
  unsupported by the trace>", ...]}
```

E.3.2 Re-authoring feedback (appended to the authoring request E.2.5)

```
GROUNDING FEEDBACK -- in your PREVIOUS attempt (per your own tool-call
  trace) you stated these concrete facts that you never READ from the
  input files; you invented them. Re-author using ONLY values you
  actually read from the inputs; where the inputs contain no such
  value, write 'not available' rather than inventing. Do NOT
  reproduce any of these:
  - <FABRICATED FACT 1>
  - <FABRICATED FACT 2>
  - ...
```

E.4 BLIND MULTI-STAKEHOLDER RUBRIC GENERATION

Three critic roles author acceptance criteria *from the goal + persona + upstream inputs only* (never seeing any candidate). All three share a form-agnostic clause and JSON schema; a synthesizer then consolidates them into one atomic, weighted rubric.

E.4.1 Owner critic

```
You ARE <OWNER NAME>, <OWNER TITLE> (<TEAM>). Responsibilities:
  <RESPONSIBILITIES>.
You will receive a deliverable "<TARGET>", meant to: <OBJECTIVE>
Success criteria on record: <CRITERIA>

You have NOT seen any submission. It will be built from these UPSTREAM
  INPUTS:
### INPUT: <NAME>
<DIGEST>
...

As the OWNER, BEFORE any submission exists, define the acceptance
  rubric -- what a top-tier "<TARGET>" MUST accomplish to satisfy
  your goal and role, grounded in what these inputs make possible.
```

```
  Use "dimension" in owner_alignment, goal_satisfaction,
  completeness, visual_structure. Reference concrete figures/fields
  the inputs contain.
<FORM-AGNOSTIC CLAUSE (E.4.4)> 8-15 requirements.
Return JSON only: <RUBRIC SCHEMA (E.4.5)>
```

E.4.2 Consumer critic (one per downstream reader)

```
You are <CONSUMER NAME, TITLE (TEAM)>. You will build your OWN
  deliverable "<CONSUMER ARTIFACT>" (<CONSUMER DESC>) using several
  inputs, including "<TARGET>" (<TARGET DESC>) PLUS <OTHER SIBLING
  INPUTS>.

You have NOT seen "<TARGET>" yet. BEFORE receiving it, define what it
  MUST provide YOU -- the specific slice of your workflow you expect
  FROM THIS input (NOT what the other inputs provide). Use
  "dimension": consumer_usefulness. <FORM-AGNOSTIC CLAUSE> 4-8
  requirements.
Return JSON only: <RUBRIC SCHEMA>
```

E.4.3 Producer critic (one per upstream input)

```
You produced the upstream input "<UPSTREAM NAME>". YOUR DATA (digest):
<UPSTREAM TEXT>

Your data will feed a derived artifact "<TARGET>" (<TARGET DESC>). You
  have NOT seen the derived artifact. Define the grounding
  requirements ANY faithful derivation MUST meet: which specific
  figures/grains/definitions/coverage-windows it MUST preserve from
  YOUR data, and what it MUST NOT fabricate or conflate. Use
  "dimension": producer_grounding. <FORM-AGNOSTIC CLAUSE> 4-8
  requirements.
Return JSON only: <RUBRIC SCHEMA>
```

E.4.4 Form-agnostic clause (shared by all three critics)

```
Phrase each requirement as an underlying NEED / OUTCOME -- WHAT
  information or
capability MUST be available to satisfy the goal -- NOT a prescribed
  layout, sheet, section, chart, or visual form. Each requirement
  must be satisfiable by MULTIPLE valid presentations (e.g., a table
  OR a chart OR an aggregate with drill-down); do NOT dictate a
  specific structure unless that structure is itself the genuine
  need. A different-but-valid form that delivers the same substance
  must be able to pass.
```

E.4.5 Rubric schema (critic output)

```
{"criteria":[{"text":"The artifact MUST ... (specific to this
  deliverable + inputs)",
              "dimension":"<dimension>","importance":"high|med|low"}]}
```

E.4.6 Rubric synthesizer – system

```
You consolidate multi-stakeholder feedback into ONE authoritative
  acceptance rubric for an enterprise artifact. You are given
  criteria raised by the OWNER (persona + goal), each CONSUMER
  (downstream reader), and each PRODUCER (upstream data source).
Produce a single de-duplicated checklist of ATOMIC, independently
  verifiable pass/fail checks that a grader will apply to ANY
  candidate version of this artifact. JSON only.
```

E.4.7 Rubric synthesizer – consolidation rules (user)

```
TARGET ARTIFACT: <TARGET>

RAW STAKEHOLDER CRITERIA (role, importance, dimension, text):
<COLLECTED OWNER/CONSUMER/PRODUCER CRITERIA>

Consolidate into the final rubric. RULES:
1. ATOMIZE + NORMALIZE. Break every raised criterion into ATOMIC
   checks -- each item asserts EXACTLY ONE independently verifiable
   thing (a single field/column, one metric, one section, one
   behavior, or one grounding fact). If a raised criterion bundles
   several facts (e.g. 'supplies assay, pathogen, collection date, and
   result date'), SPLIT it into one check per fact. Phrase each as a
   neutral, testable requirement 'The artifact MUST ...' --
   candidate-agnostic. NEVER phrase it as a finding about a specific
   version (do not say 'is not populated'; say 'MUST populate...').
   Strip any claim about whether a prior version met it.
2. Each item MUST be BINARY-CHECKABLE: a grader can mark it met/unmet
   by inspecting the artifact with no ambiguity about what 'met'
   means. Avoid vague adjectives ('professional', 'clear',
   'well-organized') UNLESS tied to a concrete observable (e.g. 'MUST
   place a titled banner on each sheet', 'MUST include a summary
   metric table on the overview').
3. MERGE only EXACT duplicates -- the SAME atomic check raised by
   different critics into ONE item; list every role in "raised_by". Do
   NOT merge two checks that differ in the specific
   field/metric/section they test.
4. Assign each item a "dimension" from EXACTLY this set:
   visual_structure, goal_satisfaction, owner_alignment, completeness,
   consumer_usefulness, producer_grounding. (Map consumer
   workflow_fit/trustworthiness -> consumer_usefulness.)
5. Assign each item an integer "weight" 1-5 (5 = goal-critical or
   factual-grounding the whole deliverable hinges on; 1 = minor
   polish).
6. Keep requirements SPECIFIC to this deliverable (reference concrete
   fields/metrics/sections), not generic boilerplate.
7. Do NOT target a fixed count. Atomization EXPANDS the list -- expect
   MORE items than the raw criteria, since each compound criterion
   splits into several atomic checks.

   Preserve EVERY distinct checkable fact; merge ONLY identical checks.

Return JSON only: {"criteria":[{"requirement":"The artifact MUST ...",
  "dimension":"<one of the set>","weight":1-5,
  "raised_by":["owner","consumer","producer"]}]}
```

E.5 CANDIDATE SCORING

Each candidate artifact is rendered to images (for layout) plus extracted text (for facts) and graded against the consolidated rubric in one call; per-item scores aggregate by normalized weight into a blended score and per-dimension scores.

E.5.1 Rubric grader

```
You are a fair, SUBSTANCE-focused grader. The <N> attached image(s) +
  extracted text are the candidate artifact '<NAME>'. Study the
  images for layout/structure and the text for facts.
EXTRACTED CONTENT:
<EXTRACTED TEXT DIGEST>

Each rubric requirement is an ATOMIC, single-fact check. Grade each
  one CRISPLY:
  1.0 = met (the single fact/capability is present and correct),
  0.0 = not met (absent, lost, or incorrect).
Use 0.5 ONLY for a coverage check of the form 'for every X' that is
  verifiably populated for a substantial majority of X but not all.
  Do NOT use 0.5 as a hedge for uncertainty -- if you are unsure,
  decide met vs not-met on the evidence.

JUDGE SUBSTANCE, NOT FORM. A check is MET if the candidate delivers
  the underlying information or capability in ANY reasonable
  presentation -- a different layout, aggregation level, chart vs
  table, section ordering, or format that still conveys the required
  substance counts as met. Do NOT penalize stylistic or structural
  choices merely for differing from one expected form. Mark UNMET
  only when the underlying information/capability is genuinely
  MISSING, LOST (e.g., aggregated so a required detail can no longer
  be recovered), or INCORRECT -- not simply presented differently
than you might expect.
For goal_satisfaction and completeness checks, an objective that is
  PROMISED but not actually DELIVERED (an empty/unpopulated result, a
  queue or plan instead of a resolved output, or a deferred action)
  scores 0.0.

RUBRIC:
[C01] (w<WEIGHT>/<DIMENSION>) <REQUIREMENT>
...

Return JSON only: {"scores":[{"id":"C01","score":0.0,"evidence":"<=1
  line"}]}
```

F PERSONA AND O*NET GROUNDING (WORKED EXAMPLES)

Every organization is seeded from real O*NET 30.3 occupations. Each persona (a *stakeholder*) is grounded in its matched occupation's *description*, sampled *core tasks*, and generalized *work activities*; these become the persona's responsibilities that seed goal sampling. After a goal is drafted it is classified back onto the O*NET work-activity taxonomy (GWA $\rightarrow$ IWA $\rightarrow$ DWA). The same persona identity then flows into authoring (prompt E.2.5, first-person owner voice) and evaluation (prompt E.4.1, owner critic). Below are three personas from distinct occupation families with the exact O*NET context each contributes.

F.1 Persona — Marketing Manager (SOC 11-2021.00, Management, Job Zone 4)

```
NAME:  Avery Sorensen        TITLE: Marketing Manager
  TEAM/INDUSTRY: Management
O*NET OCCUPATION: Marketing Managers (11-2021.00), match confidence 96
WORK-ACTIVITY CLASSIFICATION (of the sampled goal):
  GWA: Thinking Creatively
   -> IWA: Develop business or marketing plans.
   -> DWA: Develop marketing plans or strategies.
REPRESENTATIVE TASKS (O*NET, sampled into responsibilities):
```

```
  - Identify, develop, or evaluate marketing strategy, based on
    knowledge of establishment objectives, market characteristics, and
    cost and markup factors.
  - Formulate, direct, or coordinate marketing activities or policies
    to promote products or services, working with advertising or
    promotion managers.
KNOWLEDGE: Sales and Marketing; Administration and Management;
  Customer and Personal Service; English
SKILLS: Reading Comprehension; Active Listening; Speaking; Critical
  Thinking
WORK ACTIVITIES: Communicating with Supervisors, Peers, or
  Subordinates; Establishing and Maintaining Interpersonal
  Relationships; Working with Computers
SAMPLED GOAL: "Create a market entry, pricing, and launch strategy
  deck."
```

F.2 Persona — Epidemiologist (SOC 19-1041.00, Life/Physical/Social Science, Job Zone 5)

```
NAME:  Jordan Nguyen          TITLE: Epidemiologist
  TEAM/INDUSTRY: Life, Physical, and Social Science
O*NET OCCUPATION: Epidemiologists (19-1041.00)
WORK-ACTIVITY CLASSIFICATION (of the sampled goal):
  GWA: Documenting/Recording Information
   -> IWA: Present research or technical information.
   -> DWA: Prepare scientific or technical reports or presentations.
REPRESENTATIVE TASKS (O*NET, sampled into responsibilities):
  - Communicate research findings on various types of diseases to
    health practitioners, policy makers, and the public.
  - Oversee public health programs, including statistical analysis,
    health care planning, surveillance systems, and public health
    improvement.
KNOWLEDGE: Mathematics; Biology; Medicine and Dentistry; English
  Language
SKILLS: Reading Comprehension; Critical Thinking; Active Listening;
  Writing
WORK ACTIVITIES: Analyzing Data or Information; Making Decisions and
  Solving Problems; Processing Information
SAMPLED GOAL: "Create a quarterly disease surveillance and hotspot
  response deck."
```

F.3 Persona — Paralegal / Legal Assistant (SOC 23-2011.00, Legal, Job Zone 3)

```
NAME:  Jordan Kim             TITLE: Paralegal / Legal Assistant
  TEAM/INDUSTRY: Legal
O*NET OCCUPATION: Paralegals and Legal Assistants (23-2011.00)
WORK-ACTIVITY CLASSIFICATION (of the sampled goal):
  GWA: Documenting/Recording Information
   -> IWA: Prepare legal or regulatory documents.
   -> DWA: Prepare legal documents.
REPRESENTATIVE TASKS (O*NET, sampled into responsibilities):
  - Prepare affidavits or other documents, such as legal
    correspondence, and organize and maintain documents in a paper or
    electronic filing system.
  - Prepare, edit, or review legal documents, including legislation,
    briefs, pleadings, appeals, wills, contracts, and real estate
    closing statements.
KNOWLEDGE: Law and Government; English Language; Administrative;
  Computers and Electronics
SKILLS: Writing; Reading Comprehension; Active Listening; Speaking
```

```
WORK ACTIVITIES: Documenting/Recording Information; Working with
  Computers; Communicating with People Outside the Organization
SAMPLED GOAL: "Prepare a litigation readiness memo with supporting
  evidence index."
```

F.4 How the O*NET context threads through the three stages

```
GOAL SAMPLING : the persona's responsibilities = [occupation
  description] + [sampled core tasks] + [generalized work activities]
  seed an emergent goal; a one-line human phrasing tool turns the
  deliverable into the natural task ("Create a ... deck").
GENERATION    : the SAME persona (name, title, team=industry,
  responsibilities) is voiced in the authoring request (E.2.5) --
  "Hi, I'm Avery Sorensen, Marketing Manager ...".
EVALUATION    : the SAME persona authors the blind acceptance rubric
  as the OWNER critic (E.4.1) -- "You ARE Avery Sorensen, Marketing
  Manager (Management). Responsibilities: ...".
```

G WALK-THROUGH EXAMPLES: TASK, GENERATIONS, AND RUBRIC

Finally, we walk through a couple of examples to show the request the agent receives, the deliverables the models produce for it, and how the blind rubric grades them.

G.1 A POWERPOINT PRESENTATION EXAMPLE

First, we consider a target *Correction Evidence and Effective-Date Packet*—a `presentation` owned by an *Authorized Community Partner Representative* (O*NET Office and Administrative Support). At τ_1, the agent is handed the owner's first-person request together with four upstream source files, and must build the deck.

```
Hi -- I'm Morgan Okafor, Authorized Community Partner Representative
  on the Office and Administrative Support team. A big part of my
  role is determining eligibility for assistance programs and keeping
  accurate case records and reports.

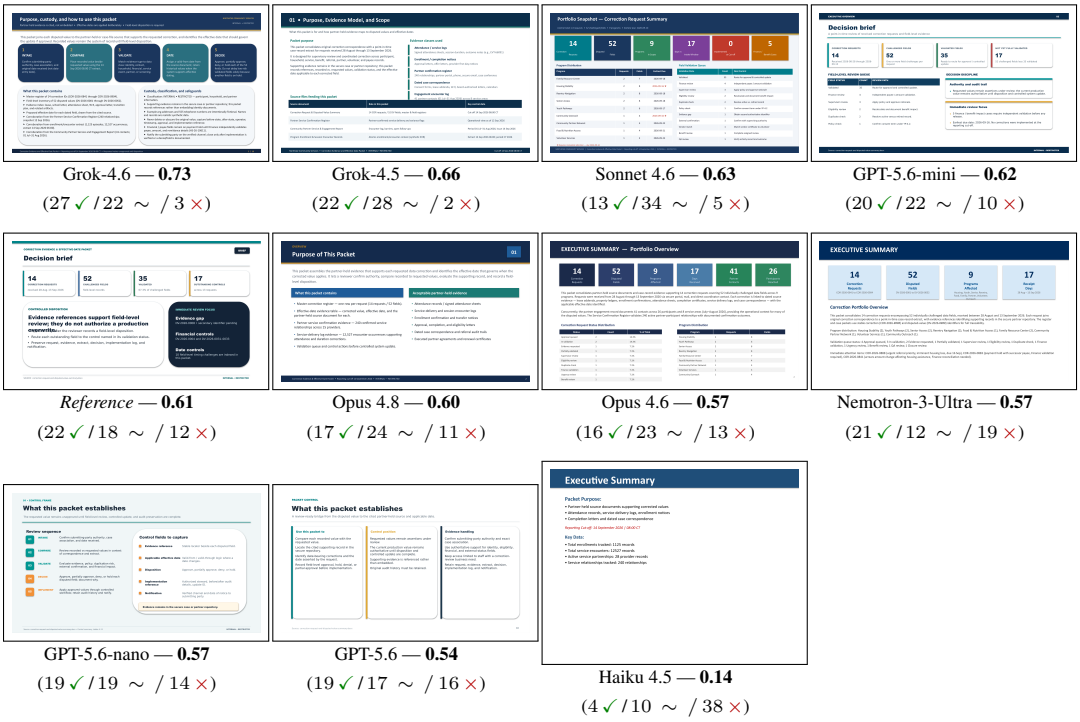
I need you to put together a presentation for me: Correction Evidence
  and Effective-Date Packet.

Here's what I'm after: partner-held source documents supporting the
  corrected values (attendance records, service-delivery logs,
  enrollment notices, completion letters, dated case correspondence),
  with the applicable effective date identified for each correction.

I've dropped the source files I actually work from into the working
  directory -- please read them first:
  - Correction Request and Disputed-Value Summary
  - Partner Service Confirmation Register
  - Community Partner Service and Engagement Report
  - Program Enrollment and Session Encounter Records

One hard rule: every concrete fact -- numbers, dates, IDs, named
  entities -- must come straight from those inputs (copy, aggregate,
  or compute from them); don't introduce any figure, date, or name
  that isn't in them or derivable from them.

Please build it as a real presentation and save it to the target path.
  Once it reads like something I'd be happy to put my name on, we're
  done.
```



Grok-4.6 — **0.73** (27 ✓ / 22 ∼ / 3 ×)
Grok-4.5 — **0.66** (22 ✓ / 28 ∼ / 2 ×)
Sonnet 4.6 — **0.63** (13 ✓ / 34 ∼ / 5 ×)
GPT-5.6-mini — **0.62** (20 ✓ / 22 ∼ / 10 ×)
Reference — **0.61** (22 ✓ / 18 ∼ / 12 ×)
Opus 4.8 — **0.60** (17 ✓ / 24 ∼ / 11 ×)
Opus 4.6 — **0.57** (16 ✓ / 23 ∼ / 13 ×)
Nemotron-3-Ultra — **0.57** (21 ✓ / 12 ∼ / 19 ×)
GPT-5.6-nano — **0.57** (19 ✓ / 19 ∼ / 14 ×)
GPT-5.6 — **0.54** (19 ✓ / 17 ∼ / 16 ×)
Haiku 4.5 — **0.14** (4 ✓ / 10 ∼ / 38 ×)

Figure 8: The same content slide from *every* model that generated the *Correction Evidence and Effective-Date Packet*, ordered by **multi-judge blended** score, with the rubric tally over the 52 criteria (✓ met / ∼ partial / × unmet); each per-criterion verdict is the average of the three judges (Opus 4.8, Sonnet 5, GPT-6). All eleven decks are visually polished; quality separates almost entirely on grounded, record-level content, not formatting.

Generations. Figure 8 shows the *same* content slide from *every* model that produced this deck, each with its *multi-judge* blended score and rubric tally. The decks are uniformly polished and the *visual_structure* formatting is consistely high, and the spread in blended score (0.73 down to 0.14) is driven almost entirely by *grounded, record-level content*. The two Grok releases build most of the evidence bridge (per-correction `COR-2026`/`DV-2026` identifiers, recorded-versus-requested values, cited source documents, and effective dates)—Grok-4.6 leads at 0.73 and Grok-4.5 follows at 0.66—yet once the three judges are averaged even they leave many record-level checks only *partially* met. On the other hand, Sonnet 4.6, the GPT-5.6 family, the Opus pair, Nemotron, and the pipeline's own *Reference* cluster in the 0.54–0.63 band, as these are polished but miss the exhaustive per-record tracing. The weak Haiku 4.5 (0.14) emits a sparse executive summary with a few aggregate counts and no traceable records.

Rubric checklist. Tables 15–16 list the *complete* 52-criterion blind rubric and three models' per-criterion verdict under the *multi-judge* average (each cell is the mean of the three judges, mapped to ✓ ≥ 0.75, × ≤ 0.25, else ∼). The separation is systematic, not cosmetic. Grok-4.6 meets 27 criteria outright, is *partial* on 22, and fails only 3: even the strongest deck cannot fully satisfy the exhaustive per-source anchoring the rubric demands—reconciling all three registers to a single effective date (C13, C21) is unmet for *every* model. GPT-5.6 holds on high-level structure and the no-fabrication rules (C01, C03–C07, C31–C32) but slips to partial or unmet on the record-grain producer-grounding criteria that require *every* corrected value to be traced, dispositioned, and dated (e.g. C24–C28, C36–C39, C43–C47), ending at 19 met / 17 partial / 16 unmet. Haiku fails 38 outright, meeting only a handful of generic "do-not-fabricate"/synthetic-data clauses (C06, C31, C32, C52). The blended scores (0.73 / 0.54 / 0.14) are the weighted aggregate of these verdicts.

Table 15: Blind rubric checklist for the example target, criteria C01–C26 of 52 (**multi-judge**: each verdict is the mean of the three judges, ✓ ≥ 0.75, × ≤ 0.25, else ∼). **G**: Grok-4.6, **P**: GPT-5.6, **H**: Haiku 4.5. Dimensions: VS visual_structure, GS goal_satisfaction, OA owner_alignment, CP completeness, CU consumer_usefulness, PG producer_grounding.

ID	Dim	Requirement	G	P	H
C01	PG	The artifact MUST link each corrected value to its specific correction identifier in the COR-2026-#### format.	✓	✓	×
C02	PG	The artifact MUST link each corrected value to its specific disputed-value identifier in the DV-2026-#### format.	✓	∼	×
C03	PG	The artifact MUST preserve the exact COR-2026-#### and DV-2026-#### identifiers from the source without inventing new identifiers or reassigning existing ones.	✓	✓	×
C04	GS	The artifact MUST present, for each disputed value, the recorded value as captured in the cited extract as of the 14 September 2026 reporting cut-off.	✓	✓	×
C05	GS	The artifact MUST present, for each disputed value, the requested/corrected value alongside the recorded value.	✓	✓	×
C06	PG	The artifact MUST NOT present a requested/corrected value as if it were already the system-of-record value.	✓	✓	✓
C07	VS	The artifact MUST clearly separate confirmed/recorded facts from unapproved requested values so a pending correction is not mistaken for an already-effective change.	✓	✓	×
C08	CU	The artifact MUST distinguish each corrected value from any values that remain unchanged so unchanged fields are left intact.	∼	✓	×
C09	GS	The artifact MUST identify an applicable effective date for each correction.	∼	×	×
C10	GS	The artifact MUST distinguish the effective date of each correction from the publication or extract dates.	✓	∼	×
C11	PG	The artifact MUST derive each effective date from or consistent with the underlying partner-held source document and MUST NOT fabricate effective dates absent from supporting evidence.	✓	∼	×
C12	PG	The artifact MUST set each effective date to reflect the service plan/attendance/enrollment/completion event date as recorded at the contact.	∼	∼	×
C13	PG	The artifact MUST anchor every corrected value from the Partner Service Confirmation Register to an effective date consistent with the register's as-of 12 Sep 2026 operational reference point.	×	×	×
C14	PG	The artifact MUST anchor every corrected value from the Program Enrollment and Session Encounter Records to an effective date within the source reporting period 2026-01-01 through 2026-12-31 and the extract-as-of moment 2026-09-14 06:00 local.	∼	×	×
C15	CP	The artifact MUST cite concrete partner-held source documents (attendance/encounter records, service delivery logs, enrollment notices, completion letters, or dated case correspondence) substantiating each corrected value.	✓	✓	×
C16	PG	The artifact MUST attribute each supporting evidence item to its actual document type as present in the underlying correction record and MUST NOT invent document types, contents, or provenance.	✓	✓	×
C17	CU	The artifact MUST carry document dates, coverage periods, or receipt indicators on each evidence item confirming it is current and authoritative for the stated effective date.	∼	∼	×
C18	OA	The artifact MUST tie each cited source document to the underlying system record it corrects, distinguishing the program case identifier from the business-object key on which the challenged field resides.	✓	∼	×
C19	CU	The artifact MUST identify which specific service, participant, enrollment, or confirmation instance each piece of evidence pertains to.	✓	✓	×
C20	PG	The artifact MUST identify which specific figure or field it corrects (participation level, status, due date, count) with enough specificity to reconcile against the original value in the Community Partner Service and Engagement Report.	∼	∼	×
C21	CP	The artifact MUST reconcile evidence drawn from the Community Partner Service and Engagement Report encounter log, the Partner Service Confirmation Register, and the Program Enrollment and Session Encounter Records, noting where they agree or conflict on the corrected value.	×	×	×
C22	GS	The artifact MUST surface disputed values for which no adequate partner-held source document exists or documentation is insufficient to support the requested value.	✓	✓	×
C23	PG	The artifact MUST NOT imply evidence exists where the source record shows none.	✓	✓	∼
C24	OA	The artifact MUST record a disposition (approve, deny, or hold pending further evidence) for each disputed value.	∼	×	×
C25	OA	The artifact MUST state a rationale for each disposition grounded in the cited evidence and effective date.	∼	×	×
C26	CU	The artifact MUST convey the reason or nature of the discrepancy the evidence reveals (what was wrong with the original value) for each correction.	✓	∼	×

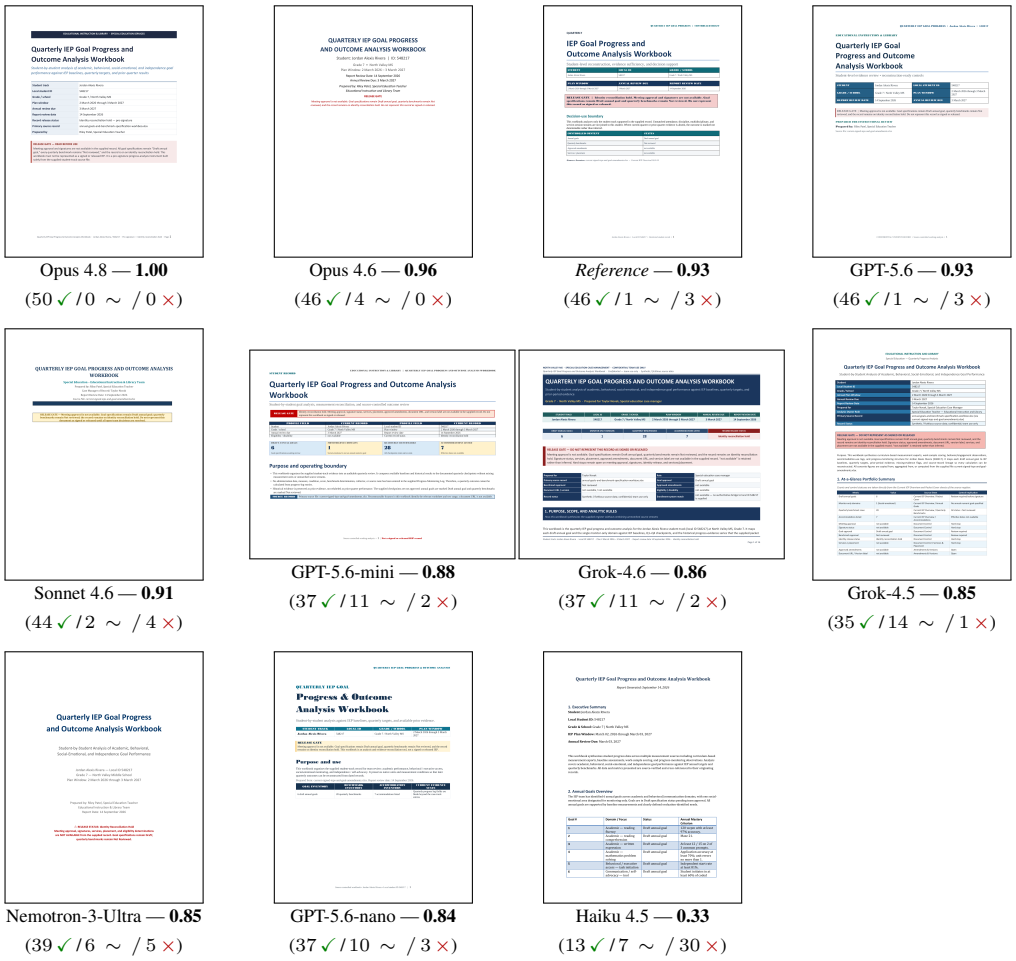
G.2 A WORD REPORT EXAMPLE

Our second example asks to create a Word `document`—a *Quarterly IEP Goal Progress and Outcome Analysis* owned by a middle-school special-education teacher (O*NET `25-2057.00`)—built from a single supplied source record (the student's annual goals-and-benchmark specification). It is deliberately adversarial on *grounding*: in that record all 6 annual goals are still "Draft," all 28 quarterly benchmarks are "Not reviewed," meeting approval and signatures are absent, and the whole record sits on an *identity-reconciliation hold* (pre-signature). A faithful report must therefore (i)

Table 16: Blind rubric checklist for the example target, criteria C27–C52 of 52 (**multi-judge**; legend as in Table 15). Grok-4.6's edge is concentrated in producer-grounding (PG): it retains record-level provenance where GPT-5.6 abstracts to summaries and Haiku omits it.

ID	Dim	Requirement	G	P	H
C27	CP	The artifact MUST account for the past-due service-plan data-quality caution, noting later activity may exist outside the reporting extract, when determining reliability of a corrected value and its effective date.	✓	×	×
C28	CP	The artifact MUST account for the caution that current-state reflects only the latest contact when determining reliability of a corrected value.	✓	×	×
C29	PG	The artifact MUST distinguish a validated corrected value from an unverified extract figure.	✓	✓	×
C30	PG	The artifact MUST respect that outcome/completion rates and attendance dispositions are incomplete for scheduled occurrences after the extract date and MUST NOT present future or not-yet-occurred sessions as having realized outcomes.	✓	✓	∼
C31	CP	The artifact MUST scope its evidence to correction requests received 28 August through 13 September 2026.	✓	✓	✓
C32	PG	The artifact MUST preserve the reporting cut-off of 14 September 2026 08:00 CT and mark any evidence supporting requests outside the coverage window as out-of-scope.	✓	✓	✓
C33	OA	The artifact MUST identify the specific provider/partner (e.g., Access Mobility Services, Bright Futures Youth Services, Community Legal Access, Harbor Community Health, Language Bridge Collective) associated with each evidence item.	∼	∼	×
C34	VS	The artifact MUST be organized so a supervisory reviewer can move from a summary of all corrections to the detailed evidence and disposition for any single disputed value via a navigable structure.	✓	✓	×
C35	VS	The artifact MUST present dates, identifiers, and recorded-versus-requested values in a consistent, legible format.	✓	✓	∼
C36	PG	The artifact MUST preserve the encounter-level grain (one row per partner contact) consistent with the source-of-record encounter log and MUST NOT trace evidence to invented aggregate rows.	∼	×	×
C37	PG	The artifact MUST preserve the definition that 'Participation' measures attendance/depth of involvement and NOT clinical or program outcome, and MUST NOT conflate participation with outcome.	∼	∼	×
C38	PG	The artifact MUST preserve the definition that 'Status' (open/escalated/closed) describes the action generated by a contact and MUST NOT restate status meanings in correction claims.	✓	∼	×
C39	PG	The artifact MUST remain within the Community Partner Service and Engagement Report's coverage of 41 contacts, 26 participants, and eight service areas and MUST NOT introduce participants, contacts, or service areas outside that extract.	∼	∼	×
C40	PG	The artifact MUST represent corrective source documents at the same de-identified grain as the partner reports and coordinator-owned contact notes and MUST NOT re-identify records.	∼	∼	∼
C41	PG	The artifact MUST preserve the identity of each specific partner relationship being corrected (provider name and individual enrollment relationship grain) as tracked in the Service Register.	∼	×	×
C42	PG	The artifact MUST map any corrected enrollment status to a defined value from the source vocabulary (Active, Pending Intake, Waitlisted, Paused, Completed, Discontinued) without inventing new values.	∼	∼	∼
C43	PG	The artifact MUST map any corrected review flag to a defined value from the source vocabulary (OVERDUE ISSUE, RECONFIRM, STALE CONTACT, CURRENT) without inventing new values.	×	×	×
C44	PG	The artifact MUST identify the original (as-recorded) value from the Partner Service Confirmation Register alongside each corrected value so the register's data is not silently overwritten.	∼	×	×
C45	PG	The artifact MUST NOT fabricate counts, shares, or aggregate figures (relationship totals, active enrollments, open/overdue issue counts, per-provider tallies) shown as blank/unpopulated in the source dashboard.	∼	∼	∼
C46	PG	The artifact MUST confine coverage to relationships and providers actually present in the source register and MUST NOT introduce providers or enrollment relationships not represented in the underlying data.	∼	∼	∼
C47	PG	The artifact MUST trace each corrected value from the Program Enrollment and Session Encounter Records to a specific enrollment episode and/or individual session occurrence rather than an ambiguous aggregate.	∼	×	×
C48	PG	The artifact MUST retain the original pre-correction values as recorded in the Program Enrollment and Session Encounter Records alongside any corrected values.	∼	∼	×
C49	PG	The artifact MUST map corrected values to the source's controlled vocabularies for encounter status, attendance disposition (including cancellations and no-shows), service type, program, location, modality, enrollment status, payer class, and treating clinician without inventing new categories or conflating distinct dispositions.	∼	×	∼
C50	PG	The artifact MUST NOT introduce figures or evidence that have no basis in the underlying enrollment/encounter atomic records.	∼	✓	∼
C51	PG	The artifact MUST treat all example.org addresses, 555 telephone numbers, and synthetic names/records as fictional placeholder data and MUST NOT represent them as real contactable or verified external evidence.	✓	×	∼
C52	PG	The artifact MUST identify the corrected data as synthetic / non-production demonstration data and MUST NOT represent it as authoritative production clinical records.	✓	×	✓

treat the student identity as provisional and not publish the identifiers as reconciled, (ii) carry the hold and draft/unreviewed status as a prominent gate, and (iii) state that no quarterly progress determination against baselines or targets can yet be finalized.



Opus 4.8 — **1.00** (50 ✓ / 0 ∼ / 0 ×)
Opus 4.6 — **0.96** (46 ✓ / 4 ∼ / 0 ×)
Reference — **0.93** (46 ✓ / 1 ∼ / 3 ×)
GPT-5.6 — **0.93** (46 ✓ / 1 ∼ / 3 ×)
Sonnet 4.6 — **0.91** (44 ✓ / 2 ∼ / 4 ×)
GPT-5.6-mini — **0.88** (37 ✓ / 11 ∼ / 2 ×)
Grok-4.6 — **0.86** (37 ✓ / 11 ∼ / 2 ×)
Grok-4.5 — **0.85** (35 ✓ / 14 ∼ / 1 ×)
Nemotron-3-Ultra — **0.85** (39 ✓ / 6 ∼ / 5 ×)
GPT-5.6-nano — **0.84** (37 ✓ / 10 ∼ / 3 ×)
Haiku 4.5 — **0.33** (13 ✓ / 7 ∼ / 30 ×)

Figure 9: Second example: first page of the *Quarterly IEP Goal Progress and Outcome Analysis* `document` from *every* model, whose single source record is on an identity-reconciliation hold with all goals Draft and benchmarks Not reviewed. Ordered by **multi-judge blended** score, with the tally over the 50 criteria (✓ met / ∼ partial / × unmet). The strong models lead with a release-gate/provenance panel and carry the hold; Haiku 4.5 writes a fluent progress report that publishes the provisional identity as reconciled, failing the grounding block.

Figure 9 shows the first page of the deliverable from *every* model, each with its multi-judge blended score over the complete rubric set. Tables 17–18 give the complete per-criterion verdicts for three of them. Unlike the presentation example, the top models handle this target well: Opus 4.8 satisfies all 50 criteria (blended 1.00), and Opus 4.6, the Reference, GPT-5.6, and Sonnet 4.6 all clear 0.90. They open with a *source-records / release-gate* panel that surfaces the identity hold and Draft/Not-reviewed status up front, and they refrain from issuing progress determinations the held data cannot support. Haiku 4.5 collapses to 0.33 (13 met / 7 partial / 30 unmet): it writes a fluent, well-formatted progress narrative but publishes the provisional identity as reconciled and claims its metrics are "source-verified," failing the owner-alignment and producer-grounding block the hold imposes (e.g. C06–C08, C11–C12, C35–C37). Fluent formatting again masks a grounding failure that the lineage-aware rubric catches.

Table 17: Blind rubric checklist for the second example (Quarterly IEP `document`), criteria C01–C25 of 50 (**multi-judge**: each verdict is the mean of the three judges, ✓ ≥ 0.75, × ≤ 0.25, else ∼). **O**: Opus 4.8, **P**: GPT-5.6, **H**: Haiku 4.5. Dimensions as in Table 15.

ID	Dim	Requirement	O	P	H
C01	PG	The artifact MUST NOT represent Jordan Alexis Rivera's IEP goals or quarterly benchmarks as signed, approved, finalized, or released.	✓	✓	✓
C02	PG	The artifact MUST show Meeting approval as 'not available'.	✓	✓	×
C03	PG	The artifact MUST show Signature status as 'not available'.	✓	✓	×
C04	PG	The artifact MUST show Goal approval as 'Draft annual goal' for all annual goals.	✓	✓	✓
C05	PG	The artifact MUST show Benchmark approval as 'Not reviewed' for the quarterly benchmarks.	✓	✓	✓
C06	OA	The artifact MUST carry an unresolved-status caveat prominently on any progress analysis built on this record.	✓	✓	×
C07	OA	The artifact MUST treat the student identity as provisional and on identity reconciliation hold, not reconciled or verified.	✓	✓	×
C08	OA	The artifact MUST NOT publish or act on the student identifiers (name, Local ID 548217) as reconciled or verified.	✓	✓	×
C09	PG	The artifact MUST NOT merge or conflate this record with any other student record.	✓	✓	✓
C10	PG	The artifact MUST bind all progress data to the stated student identity (Jordan Alexis Rivera, Local ID 548217, Grade 7, North Valley MS).	✓	✓	✓
C11	GS	The artifact MUST state explicitly that no quarterly progress determination against baselines or targets can be finalized while goals remain Draft and benchmarks remain Not reviewed.	✓	✓	×
C12	PG	The artifact MUST flag any progress computed against these goals or benchmarks as measured against unapproved, non-authoritative targets.	✓	✓	×
C13	CP	The artifact MUST provide a per-goal readiness/blocker view covering all 6 draft annual goals and the 1 monitor-only domain.	✓	✓	×
C14	CP	The artifact MUST distinguish goals that could be measured versus those blocked by draft status.	✓	✓	×
C15	CP	The artifact MUST surface missing-evidence flags for every goal.	✓	✓	×
C16	CP	The artifact MUST provide source-record linkage for every goal to its underlying curriculum-based measurement export, gradebook record, behavior event, attendance, work-sample, or observation source.	✓	✓	∼
C17	CP	The artifact MUST group or attribute goals across the four IEP goal domains (academic, behavioral, social-emotional, independence).	✓	✓	✓
C18	CP	The artifact MUST expose measurement dates for each data point.	✓	✓	×
C19	CP	The artifact MUST expose goal mappings for each data point.	✓	✓	✓
C20	CP	The artifact MUST expose quarter attribution for each data point.	✓	✓	∼
C21	GS	The artifact MUST reflect the plan window 2 March 2026 through 3 March 2027.	✓	✓	✓
C22	GS	The artifact MUST identify the specific quarter under analysis.	✓	✓	×
C23	PG	The artifact MUST NOT attribute progress evidence to periods outside the plan coverage window without distinguishing it.	✓	✓	×
C24	CP	The artifact MUST capture the 7 listed accommodations context where relevant to interpreting performance.	✓	✓	×
C25	PG	The artifact MUST show the Document URL field as 'not available' rather than a fabricated value.	✓	✓	×

Table 18: Blind rubric checklist for the second example (Quarterly IEP `document`), criteria C26–C50 of 50 (**multi-judge**; legend as in Table 17). Opus 4.8 meets every criterion; GPT-5.6 slips only on the change-over-time consumer-usefulness checks (C35–C38) that the draft record cannot fully support; Haiku fails the identity/approval-gate block outright.

ID	Dim	Requirement	O	P	H
C26	PG	The artifact MUST show the Version label field as 'not available' rather than a fabricated value.	✓	✓	×
C27	PG	The artifact MUST show the Approved amendments field as 'not available' rather than a fabricated value.	✓	✓	×
C28	VS	The artifact MUST make the hard-stop/gate conditions immediately visible at the top level rather than buried in downstream detail.	✓	✓	×
C29	VS	The artifact MUST support drill-down from summary-level progress claims to individual contributing evidence records.	✓	✓	∼
C30	VS	The artifact MUST use consistent, distinguishable status labels for goal approval, benchmark approval, and evidence sufficiency such that 'Draft', 'Not reviewed', 'missing evidence', and 'measurable' states are never conflated.	✓	✓	∼
C31	GS	The artifact MUST identify the specific next actions needed to unblock analysis (obtain meeting approval and signature, review goals/benchmarks, clear identity hold, resolve missing-evidence flags).	✓	✓	∼
C32	GS	The artifact MUST tie each blocker to what would make the affected goals scorable.	✓	✓	×
C33	CU	The artifact MUST express each goal's current-quarter performance against that goal's baseline.	✓	✓	×
C34	CU	The artifact MUST express each goal's current-quarter performance against that goal's quarterly target.	✓	✓	×
C35	CU	The artifact MUST enable classification of each goal as on-track, at-risk, or not-progressing.	✓	×	×
C36	CU	The artifact MUST show the direction of change relative to the prior quarter for each goal.	✓	×	×
C37	CU	The artifact MUST show the magnitude of change relative to the prior quarter for each goal.	✓	×	×
C38	CU	The artifact MUST distinguish improvement, plateau, or regression rather than reporting only a single snapshot.	✓	∼	×
C39	CU	The artifact MUST flag goals with stale evidence, including where measurement dates are old or missing.	✓	✓	×
C40	CU	The artifact MUST map each measured outcome to its specific IEP goal.	✓	✓	✓
C41	CU	The artifact MUST map each measured outcome, where available, to the associated accommodation, behavior support, assistive technology, or instructional strategy.	✓	✓	∼
C42	CU	The artifact MUST identify the evidence type for each reported value.	✓	✓	✓
C43	CU	The artifact MUST present performance across all four domains for the student together to support a whole-student view.	✓	✓	∼
C44	CU	The artifact MUST surface which goals are furthest from target or most at risk to support prioritization.	✓	✓	×
C45	PG	The artifact MUST preserve exactly 6 draft annual goals and MUST NOT invent additional goals.	✓	✓	✓
C46	PG	The artifact MUST preserve exactly 28 quarterly benchmarks and MUST NOT invent additional benchmarks.	✓	✓	×
C47	PG	The artifact MUST preserve exactly 1 monitor-only domain and MUST NOT invent additional domains.	✓	✓	✓
C48	PG	The artifact MUST preserve exactly 7 accommodations and MUST NOT invent additional accommodations.	✓	✓	×
C49	PG	The artifact MUST distinguish the single monitor-only domain from goal-bearing domains so monitor-only content is not scored or reported as an active measurable goal with targets.	✓	✓	✓
C50	PG	The artifact MUST assert signature, approval, amendment, URL, and version fields only when supported by an actual supplied record, keeping derived figures traceable to source and free of unsupported authority.	✓	✓	×