WORKSURFACE-BENCH: BENCHMARKING ENTERPRISE AGENTS ON MULTI-SURFACE KNOWLEDGE ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Enterprise agents often need several kinds of knowledge at once: documents for narrative facts, tables for calculations, and dependency graphs for file relationships. Existing benchmarks usually test retrieval or tool use, but do not separate *choosing the right knowledge form* from using it correctly. We introduce **WorkSurface-Bench**, which evaluates this choice as *surface routing*. The benchmark projects five persona-scoped Workspace-Bench-Lite workspaces onto document, table, and graph surfaces and contains 1,151 atomic tasks. Its reference answers are auditable: table answers come from executed DuckDB queries, document answers are checked against verbatim spans, and graph answers derive from source annotations. We evaluate four backbones under six agent settings, yielding 27,624 retained trajectories with no protocol errors. Gold-constrained agents reach 98.7–99.8 Route F1, yet Answer remains 56.1–75.3%. A matched intervention shows that surface hints improve Answer for three of four models, whereas removing irrelevant tools mainly improves routing and efficiency. In an independent three-annotator audit, all 200 sampled tasks pass all six criteria by majority vote. We release the construction pipeline, scoring code, and agent harness at `https://anonymous.4open.science/r/WorkSurface-Bench-Anonymous-D097`.

1 INTRODUCTION

Consider an operations analyst asking: *"Which source files feed the negative-variance inventory report, and what is the total logistics cost by shipping mode?"* The agent must read a report for context, query a table for the total, and follow a dependency graph to identify the source files. These are not interchangeable operations. Document retrieval cannot perform a SQL aggregate, and a table query does not reveal file lineage. Before answering, the agent must first decide which kinds of knowledge the question requires.

We call each kind of knowledge a *surface*. In this paper, **surface routing** means selecting the document, table, or graph surfaces needed for a question. This capability is usually hidden inside an end-to-end score. RAG benchmarks (Friel et al., 2024; Chen et al., 2020; 2021) mainly ask whether the final answer is correct. If a model chooses the wrong source or retrieves the wrong evidence, both failures simply produce an incorrect answer. Tool-use benchmarks (Huang et al., 2023; Chen et al., 2023; Li et al., 2023b) evaluate API selection, but they do not test whether an agent recognizes that a question requires a different representation of knowledge, such as SQL rather than prose search. For deployment, this distinction matters: choosing the right surface and reasoning correctly after that choice are separate abilities.

We introduce **WorkSurface-Bench** (Figure 1) to measure both abilities. We project five persona-scoped workspaces onto three routable surfaces: a document knowledge base, a DuckDB table registry, and a file dependency graph. Procedural SOPs remain task metadata rather than a fourth routable surface. Each task receives four scores—Route, Evidence, Answer, and Efficiency—so the evaluation shows whether an agent chose the right surface, found the right evidence, answered correctly, and used a reasonable amount of computation. Reference answers never come directly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

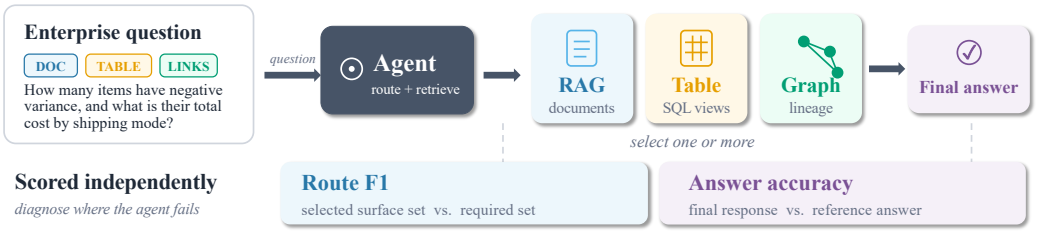

Figure 1: WorkSurface-Bench projects a shared enterprise benchmark corpus onto three canonical surfaces (RAG, Table, Graph) and scores *Route* (surface selection) and *Answer* (final correctness) as separate metrics.

from model free text: we execute table queries, verify document spans, and derive graph answers from source annotations.

This separation makes failures easier to interpret. A low Route score means the agent chose the wrong surfaces; a low Evidence score means it chose the right surfaces but did not access all required artifacts. High Route and Evidence with low Answer is consistent with a downstream computation or synthesis error, although the artifact-level Evidence metric does not prove correct semantic use. End-to-end accuracy alone cannot expose these stages.

We evaluate four backbones (GPT-4o-mini, DeepSeek-V4-Pro, Gemini-3.1-Pro, and GPT-5.5) under six agent settings on 1,151 tasks. The main result is simple: **better routing does not guarantee a better answer**. Gold-constrained agents reach 98.7–99.8 Route F1, while Answer remains 56.1–75.3%. A matched gold-hint condition further separates knowing the required surfaces from removing irrelevant tools. Even after an agent knows where to look, it can still retrieve incomplete evidence, combine sources incorrectly, or make a computation error.

Our contributions are:

1. **A multi-surface routing benchmark** (§3): 1,151 tasks from five persona-scoped workspaces, projected onto document, table, and graph surfaces. To our knowledge, it is the first workspace benchmark to score surface routing separately from the Answer metric.
2. **A verifiable construction pipeline** (§3.3): deterministic rules and verified LLM-assisted proposals produce gold answers from executed queries, checked spans, or source graph annotations rather than model-generated values.
3. **A diagnostic scoring protocol** (§4): Route, Evidence, Answer, and Efficiency are scored separately, making it possible to locate failures that one end-to-end score would hide.
4. **Evidence that routing and answering differ** (§5): gold-surface guidance improves Route F1 much more than Answer across four complete backbone sweeps with no retained protocol errors.

2 Related Work

WorkSurface-Bench sits at the intersection of four active benchmark threads: multi-source retrieval-augmented QA, enterprise-scale agent evaluation, tool and API routing, and semi-structured knowledge retrieval. Table 1 places representative benchmarks along the two axes we care about — how many categorically distinct *surfaces* are present, and whether surface selection itself is a scored metric — and shows that the combination is not represented among the surveyed benchmarks.

2.1 Multi-source Retrieval-Augmented QA

The earliest attempts to move beyond single-surface RAG fused *two* sources into one integrated question. HybridQA (Chen et al., 2020) pairs Wikipedia tables with linked passages; OTT-QA (Chen et al., 2021) extends this to 400K open-domain tables and 5M passages; T^2-RAGBench (Isgorur et al., 2026) adds financial text-and-table retrieval; and MMQA (Talmor et al., 2021) combines text,

Table 1: Positioning against prior benchmarks. WorkSurface-Bench projects three routable knowledge surfaces over a shared enterprise corpus, attaches SOP metadata, and scores surface routing separately from answer correctness. Entries are based on the cited benchmark descriptions.

Benchmark	#Surfaces	Route metric	Enterprise src	Skill/SOP	Graph type
Workspace-Bench 1.0 (Tang et al., 2026)	files (1 form)	✗	✓	rubric	file-dep (1-hop)
STaRK (Wu et al., 2024)	2 (text+KG)	✗	✗	✗	concept KG
HybridQA (Chen et al., 2020)	2 (text+table)	✗	✗	✗	—
OTT-QA (Chen et al., 2021)	2 (text+table)	✗	✗	✗	—
T^2-RAGBench (Isgorur et al., 2026)	2 (text+table)	✗	✗	✗	—
MMQA (Talmor et al., 2021)	3 (text+table+image)	✗	✗	✗	—
MetaTool (Huang et al., 2023)	tools (100s)	✓ (tool)	✗	✗	—
T-Eval (Chen et al., 2023)	tools	✓ (step)	✗	✗	—
API-Bank (Li et al., 2023b)	APIs (∼50)	✓ (API)	✗	✗	—
RAGBench (Friel et al., 2024)	1 (docs)	✗	partial	✗	—
BIRD / Spider (Li et al., 2023a; Yu et al., 2018)	1 (SQL)	✗	✗	✗	—
TheAgentCompany (Xu et al., 2024)	env (tools+web)	✗	✓	✗	—
mmRAG (Xu et al., 2025)	3 (text+table+KG)	✓ (dataset)	partial	✗	Freebase (verbalized)
WorkSurface-Bench (ours)	**3 routable + SOP metadata**	✓ (surface)	✓	✓ (SOP)	file lineage+enrich

tables, and images. In these benchmarks, the model is not scored for deciding which representation it needs. mmRAG (Xu et al., 2025) is closest in covering text, tables, and knowledge graphs and evaluating query routing, but converts them into retrievable documents; we retain surface-native tools and score their actual use. Enterprise Deep Search (Choubey et al., 2025) evaluates multi-hop retrieval across heterogeneous enterprise artifacts, whereas we separate surface selection, artifact access, and answering over native document, SQL, and lineage-graph interfaces.

2.2 Semi-structured Knowledge Retrieval

STaRK (Wu et al., 2024) evaluates retrieval over semi-structured knowledge bases where a concept-level knowledge graph (entities, typed relations) is blended with textual descriptions. Our Graph surface instead encodes workspace-native file lineage (file-depends-on-file, file-supports-output, task-requires-file), supporting questions such as which spreadsheets feed a dashboard. It therefore tests a different structure from concept-level KG retrieval.

2.3 Tool and API Routing

MetaTool (Huang et al., 2023), T-Eval (Chen et al., 2023), API-Bank (Li et al., 2023b), and ToolLLM (Qin et al., 2023) evaluate tool or API selection; AgentBoard (Ma et al., 2024) tracks multi-turn progress. They route among operational endpoints or action sequences. We instead route among information types: tools execute a route, while surfaces determine the representation the agent must consult. This decision is complementary to retriever routing: an agent may first choose a representation and then select a retriever, query program, or API within it. This layering clarifies whether a failure occurs before or after a surface-native interface has been selected, which matters when comparing modular agent systems and multi-stage pipelines.

2.4 Enterprise Workspace Agents

Workspace-Bench (Tang et al., 2026), TheAgentCompany (Xu et al., 2024), WorkArena (Drouin et al., 2024), AppWorld (Trivedi et al., 2024), and τ-bench (Yao et al., 2024) target end-to-end workspace, web, or API success. We inherit Workspace-Bench's source and provenance pipeline, but isolate knowledge routing and expose file dependencies as a lineage-query surface.

3 Benchmark Construction

WorkSurface-Bench is built by a deterministic pipeline that adds no synthetic facts, followed by a verifiable LLM-assisted expansion. Figure 2 gives the overview; we detail each stage below and report the resulting statistics in Table 2.

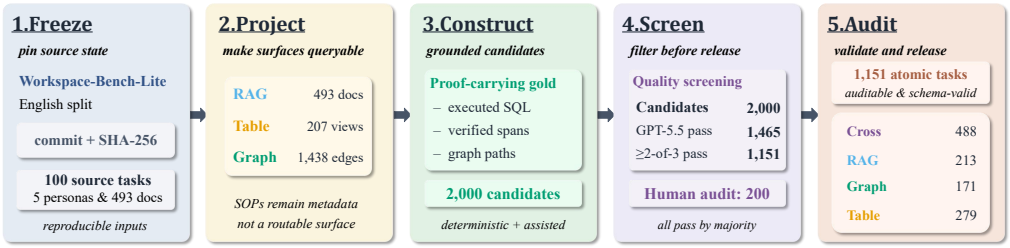

Figure 2: Construction pipeline. Gold answers trace to executed DuckDB results, verified document spans, or source dependency annotations; model-generated proposals are grounded through these proof artifacts. Details in §3.3.

3.1 Source Data and Provenance

We derive the benchmark from the Workspace-Bench-Lite English split. Its data pipeline is hybrid: task scenarios and dependency graphs are human-authored and expert-validated against Lark/ByteDance workflows, while workspace files combine public web resources with grounded LLM-generated artifacts. The result is an enterprise-workflow benchmark with human-authored task, rubric, and graph supervision. For reproducibility, the first pipeline stage records the source commit and every input file's SHA-256 in a lock file. We assess closed-book answerability separately (§5.6).

3.2 Surface Projection

Each persona workspace is projected onto three routable surfaces. **RAG**: documents, reports, PDFs, and slide text are converted to canonical UTF-8 markdown so routing is evaluated separately from OCR/parsing. **Table**: normalized CSV/XLSX sheets are registered as per-workbook DuckDB views with provenance columns; a coverage gate retains non-trivial sheets (at least 30 rows, four columns with at least three rows, or five distinct non-empty values in one column) as tables and maps the remainder to RAG. **Graph**: the Workspace-Bench file dependency graph enriched with programmatic edges (`mentions`, `schema_overlap`, `version_of`, `shared_artifact`). Repeated rubric patterns yield `applicable_skills` metadata, which serves as auxiliary task context rather than a routable surface. Each surface passes executable eligibility and leakage checks reported in Appendix B.

3.3 Task Derivation

Atomic tasks are derived in two passes. The *deterministic* pass derives golds directly from source artifacts: numeric rubric values ($1,710,971.47, 16.89%, "1000 records") become gold answers; one `graph_only` task per source task asks which files are required inputs; and `cross_surface` tasks are emitted when an executed DuckDB aggregate over a named column reproduces the rubric value. In the *verifiable LLM-assisted* pass, the model proposes questions together with proof artifacts. Table and cross-surface proposals include a DuckDB query whose executed result becomes the gold; RAG proposals include an answer verified verbatim in the source document. Proposals with query errors or missing spans are discarded. Graph-only golds come directly from source dependency annotations. A final QC pass deduplicates the tasks and re-validates the schema.

3.4 Automatic Validation and Model-assisted Screening

We generate a 2,000-item candidate pool and first apply deterministic schema, uniqueness, language, leakage, query-execution, span, and graph-path checks. Every table-bearing item must reproduce its gold through a read-only query, every RAG-bearing item must point to a verified span, and every graph-bearing item must match a source dependency path. IDs and normalized questions must also be unique.

Table 2: WorkSurface-Bench composition (left) and verification coverage (right). Table-bearing golds are executed DuckDB results; RAG evidence is verified against document spans; and graph golds derive from source dependency annotations (§3.3).

Dataset statistic	Value	Artifact / check	Value
Personas	5	Routable surfaces	3
Source tasks (WSB-Lite)	100	KB canonical docs	493
Atomic tasks	1,151	Table views (DuckDB)	207
Task types	4	Graph edges (total)	1,438
cross_surface	488	Executed table-bearing gold	453/453
rag_only	213	RAG-bearing span verification	601/601
graph_only	171	Graph-bearing path verification	600/600
table_only	279	2-of-3 screening pass	1,151/2,000
Persona count range	39–595	Efficiency budgets	1,151/1,151

We then apply a semantic screen for answerability, gold correctness, naturalness, atomicity and unambiguity, necessity of the declared surfaces, and absence of leakage cues. GPT-5.5 retains 1,465 candidates under this strict rubric. Independent DeepSeek and Gemini audits produce the final 1,151-item set by requiring a strict pass from at least two of the three auditors; 429 items pass all three. A strict pass requires an affirmative judgment on every dimension, with failed or "unsure" judgments treated as non-passes.

3.5 Independent Human Audit

Three annotators independently audit the same stratified 200-item sample using the released evidence package and rubric. The sample covers task type, persona, and answer type, including all 15 tri-surface items. All 200 items receive majority-pass judgments on all six criteria. Surface necessity and atomicity are unanimously positive for every item, and 192 items receive unanimous judgments across all six criteria. We retain the item-level votes and compute these rates on the originally sampled items.

3.6 Dataset Distribution and Coverage

The final distribution reflects the candidates that satisfy verification and surface-necessity criteria. It contains 488 cross-surface tasks: 314 RAG+Graph, 100 Graph+Table, 59 RAG+Table, and 15 requiring all three surfaces. Tri-surface tasks are less frequent because each of the three evidence forms must be independently necessary and the composed gold must remain executable or traceable.

Persona frequencies follow the source workspaces and remain concentrated in operational roles (Table 2).

3.7 Dataset Statistics

The benchmark comprises 1,151 atomic tasks over five personas (Table 2): 488 `cross_surface`, 213 `rag_only`, 171 `graph_only`, and 279 `table_only`. Each task carries a natural-language question, gold answer, required surfaces, expected tool types, evidence annotations, and Workspace-Bench provenance. Answer-bearing golds are auditable: all 453 table-bearing queries reproduce their reference answers, all 601 RAG-bearing items match verified document spans, and all 600 graph-bearing items pass executable node/path checks (Table 2).

4 Scoring Protocol

For each task i, the evaluator consumes the final answer and tool trace and returns four scores in $[0, 1]$. Table 3 reports macro-averages over all 1,151 tasks. The decomposition separates surface selection, evidence retrieval, answer correctness, and inference cost.

Table 3: Main results (%) across four models, six settings, and 1,151 tasks. Route-F1, Evidence, Answer, and Efficiency form Agg. as defined in §4. No-tool Route is unavailable and scored zero. Shading marks the two gold-surface conditions; bold marks the within-model best. All 27,624 retained trajectories are protocol-error-free.

Configuration		Routing			Agent performance		Resource use	Overall
Model	**Setting**	**P**	**R**	**F1**	**Evidence**	**Answer**	**Eff.**	**Agg.**
GPT-4o-mini	No-tool	–	–	–	0.0	0.0	**98.6**	9.9
	Always-RAG	52.2	35.1	40.8	14.1	18.8	90.8	30.1
	Naive-router	43.5	29.2	33.9	27.6	12.1	74.6	28.4
	ReAct-all	66.8	79.4	70.4	59.7	48.7	53.5	57.9
	Gold-constrained	**99.8**	**98.1**	**98.7**	**66.7**	56.1	75.0	**71.8**
	Gold-hint/all	81.6	90.0	84.0	64.4	**58.6**	52.7	66.1
DeepSeek-V4-Pro	No-tool	–	–	–	0.0	0.0	**96.1**	9.6
	Always-RAG	52.2	35.1	40.8	14.1	20.3	85.7	30.1
	Naive-router	55.4	46.7	49.5	36.3	40.6	51.8	42.6
	ReAct-all	69.0	99.3	79.1	77.2	67.9	27.2	69.4
	Gold-constrained	**100.0**	**99.5**	**99.7**	77.0	**70.7**	43.4	**77.1**
	Gold-hint/all	71.6	99.2	81.0	**77.5**	65.7	26.5	69.1
Gemini-3.1-Pro	No-tool	–	–	–	0.0	0.0	**93.6**	9.4
	Always-RAG	52.2	35.1	40.8	14.1	18.8	83.3	29.3
	Naive-router	79.6	63.9	69.1	48.9	60.5	39.6	57.1
	ReAct-all	90.3	99.6	93.5	77.6	76.0	32.1	76.5
	Gold-constrained	**100.0**	99.7	**99.8**	**78.5**	75.3	40.1	**78.9**
	Gold-hint/all	96.9	**100.0**	98.0	78.3	**78.8**	31.3	78.7
GPT-5.5	No-tool	–	–	–	0.0	2.4	**94.8**	10.3
	Always-RAG	52.2	35.1	40.8	14.1	22.7	85.9	30.9
	Naive-router	79.5	63.4	68.7	46.4	55.3	44.2	54.9
	ReAct-all	74.3	99.1	82.4	76.1	63.8	35.6	69.3
	Gold-constrained	**100.0**	99.4	**99.6**	75.4	61.9	42.3	**73.4**
	Gold-hint/all	79.1	**99.9**	86.1	**76.6**	**68.1**	32.1	71.5

Route. Let $G_i \subseteq \{\text{RAG}, \text{TABLE}, \text{GRAPH}\}$ be the gold surfaces and C_i the surfaces actually used in the trace. We compute $P_i = |C_i \cap G_i|/|C_i|$, $R_i = |C_i \cap G_i|/|G_i|$, and their harmonic mean F_i^{route}. Empty predictions receive zero when a surface is required. Table 3 reports Route-P, Route-R, and Route-F1, but only Route-F1 enters the aggregate. For No-tool, the Route columns are displayed as "–" because no routing action is available, while Route-F1 is scored as zero whenever a surface is required.

Evidence. Each gold evidence item is matched against the trace: a RAG item is hit when its source file is read, a table item when its registered view is queried, and a graph path item only when its terminal evidence node is returned. For graph items annotated as complete neighbor sets, every verified member must be returned. The Evidence score E_i is the fraction of matched artifact-level items, so tasks with multiple annotations receive proportional rather than surface-level credit. It measures access to the required artifacts, not whether the final synthesis uses their contents correctly.

Answer. The Answer score A_i depends on answer type. Integers with absolute value at most 100 are matched exactly; other numbers use a 5% relative-error tolerance. Lists receive order-invariant set F1, strings and booleans use normalized exact match, and abstention requires `INSUFFICIENT_EVIDENCE`. Ordered composite strings are compared element-wise, making the canonical `item; value` form equivalent to a JSON array containing the same elements. The dataset has no free-form items, so no model-based judge is used.

Efficiency. Let t_i be the observed token count and b_i the task-specific common budget, fixed at twice the token count of the canonical GPT-4o-mini S5 trace. The continuous linear score is

$$Q_i = \max\left\{0,\, 1 - \frac{t_i}{2b_i}\right\}.$$

Table 4: Mean Answer (%) by task type. Shading marks the two gold-surface conditions; bold marks the within-model best.

Configuration		Answer score by task type			
Model	**Setting**	**RAG**	**Table**	**Graph**	**Cross**
GPT-4o-mini	No-tool	0.0	0.0	0.0	0.0
	Always-RAG	**70.0**	0.0	26.4	4.5
	Naive-router	4.7	8.6	59.0	0.8
	ReAct-all	19.7	**88.5**	67.1	32.2
	Gold-constrained	58.7	86.0	**72.9**	32.0
	Gold-hint/all	46.0	88.4	70.5	**42.8**
DeepSeek-V4-Pro	No-tool	0.0	0.0	0.0	0.0
	Always-RAG	**80.8**	0.4	24.1	4.1
	Naive-router	74.2	74.5	20.5	13.5
	ReAct-all	**80.8**	85.2	72.3	50.8
	Gold-constrained	80.3	**91.0**	70.7	**54.9**
	Gold-hint/all	78.9	86.7	**72.8**	45.5
Gemini-3.1-Pro	No-tool	0.0	0.0	0.0	0.0
	Always-RAG	81.2	0.0	25.6	0.0
	Naive-router	82.2	87.4	47.1	40.4
	ReAct-all	83.1	86.5	64.9	70.9
	Gold-constrained	82.6	87.0	**68.1**	68.0
	Gold-hint/all	**83.6**	**88.8**	65.3	**75.8**
GPT-5.5	No-tool	12.7	0.0	0.0	0.0
	Always-RAG	82.2	0.0	29.0	7.6
	Naive-router	82.6	84.4	39.9	32.2
	ReAct-all	83.1	86.2	**73.9**	38.9
	Gold-constrained	83.6	85.5	**73.9**	34.6
	Gold-hint/all	**84.0**	**87.6**	73.4	**48.2**

Thus, Q_i decreases from 1 at zero tokens to 0.5 at the common budget and reaches 0 at twice the budget. If no positive budget is available, we assign a default value of $Q_i = 1$; this defensive fallback is not triggered by the released set.

Aggregate. The released set has no activated safety-threat annotations, so Safety is omitted and does not enter this aggregate. We define the aggregate using four evaluation hyperparameters:

$$\begin{aligned} \mathrm{Agg}_i = \alpha A_i + \beta E_i \\ + \gamma F_i^{\mathrm{route}} + \delta Q_i, \\ \alpha + \beta + \gamma + \delta = 1. \end{aligned}$$

Here α, β, γ, and δ weight Answer, Evidence, Route, and Efficiency, respectively. We set $\alpha = 0.35$, $\beta = 0.30$, $\gamma = 0.25$, and $\delta = 0.10$: end-task correctness receives the largest weight, evidence and routing remain substantial, and efficiency is secondary. Equal weighting would assign 25% to Efficiency and allow a cheap but incorrect no-tool system to receive substantial credit. The 10% weight keeps efficiency consequential without allowing it to dominate task success. We average Agg_i across tasks rather than reconstructing it from rounded column means; component scores remain the primary diagnostic outcomes.

5 Experiments

5.1 Experimental Setup

Backbones. Using identical prompts and tool schemas, we query four proxy-exposed API identifiers: `gpt-4o-mini` (OpenAI, 2024), `deepseek-v4-pro` (DeepSeek-AI, 2026), `gemini-3.1-pro-preview` (Google, 2026), and `gpt-5.5` (OpenAI, 2026). Released traces record the requested identifiers (costs in Appendix D).

Agent settings. Six settings span the routing autonomy spectrum: (**S1**) *No-tool* answers from parametric memory only — the closed-book baseline; (**S2**) *Always-RAG* forces a single `kb_search` on every task; (**S3**) *Naive-router* picks exactly one surface up front and uses only its tools; (**S4**) *ReAct-all* exposes all three surfaces and lets the agent freely interleave them; and (**S5**) *Gold-constrained* is given the required surfaces and can only use tools within them; and (**S6**) *Gold-hint/all* receives the same surface names while all three tool families remain available. Both conditions leave tool choice, evidence retrieval, and answering to the model. S4→S6 isolates the effect of surface information, while S6→S5 isolates the effect of removing irrelevant tools after providing the same hint.

Tools and inference. RAG, Table, and Graph expose search, read-only DuckDB, and graph-traversal tools, respectively. We use a portable text-based ReAct loop capped at eight tool turns (median four), with temperature 0 and greedy decoding. S4 and S6 expose all three surfaces, creating distractor pressure; the other settings restrict tool access as defined above. Each task's common Efficiency budget is fixed at $2\times$ the token count of the canonical GPT-4o-mini S5 trace. Exact setting and tool schemas appear in Appendix C; implementation and cost details appear in Appendix D.

5.2 Main Results

Table 3 reports component and aggregate scores for every model–setting pair. Tool access generally improves Answer over No-tool, but the gains are non-monotonic. Gemini attains the highest aggregate (78.9, S5); among unconstrained S4 agents it also leads at 76.5, followed by DeepSeek (69.4), GPT-5.5 (69.3), and GPT-4o-mini (57.9). Across all backbones, S4 improves aggregate over S3 by 14.4–29.5 points, yet remains 2.4–13.9 points below each model's best gold-informed condition. Component scores reveal distinct optima: Gemini's highest Answer occurs in S6 (78.8), although S5 gives a marginally higher aggregate; GPT-4o-mini and GPT-5.5 show the same answer–aggregate trade-off. **No single setting dominates every component.**

Two off-the-shelf SDK agents provide an external-validity check (Appendix C.2). Codex SDK with GPT-5.5 raises Answer over the matched ReAct-all run from 63.8 to 78.5, but Efficiency falls from 35.6 to 0.3: richer orchestration helps, at substantial context cost. Only this Codex comparison is backbone-matched; Claude SDK is an unmatched external baseline.

5.3 Separating Surface Hints from Tool Restriction

The matched S4→S6 contrast isolates the effect of naming the required surfaces while keeping all tools available. Route F1 changes by $+13.5$, $+1.8$, $+4.5$, and $+3.6$ points for GPT-4o-mini, DeepSeek, Gemini, and GPT-5.5; Answer changes by $+9.9$, -2.2, $+2.8$, and $+4.4$. Source-task clustered 95% intervals exclude zero for the three positive Answer changes, but include zero for DeepSeek. **Surface hints help several models, but their value is model-dependent rather than automatic.**

The S6→S5 contrast then isolates removal of irrelevant tools under the same hint. Route F1 increases by 1.8–18.7 points and Efficiency by 8.8–22.3, yet Answer changes by -2.5, $+5.0$, -3.5, and -6.3 points. Only the DeepSeek gain and GPT-5.5 decline have clustered intervals excluding zero. Thus, restricting tools makes traces cleaner and cheaper, but does not uniformly improve the final answer.

5.4 Performance by Surface Composition

Table 4 and Figure 3c decompose Answer by task type. RAG, Table, Graph, and Cross peak at 84.0%, 91.0%, 73.9%, and 75.8%, respectively. Cross is therefore not uniformly the lowest category: Gemini reaches 75.8 under S6. Nevertheless, each model's best cross-surface score is 13.0–45.7 points below its own best single-surface score. **Evidence composition remains a substantial, but model-dependent, challenge.** The profiles differ sharply. GPT-4o-mini is strongest on Table, DeepSeek pairs strong Table performance with moderate Cross performance, and Gemini is much more robust on Cross. GPT-5.5 retrieves competitive single-surface answers but drops to 48.2 on Cross. These differences argue against describing the dataset as uniformly difficult or balanced by outcome. Per-surface reporting is essential.

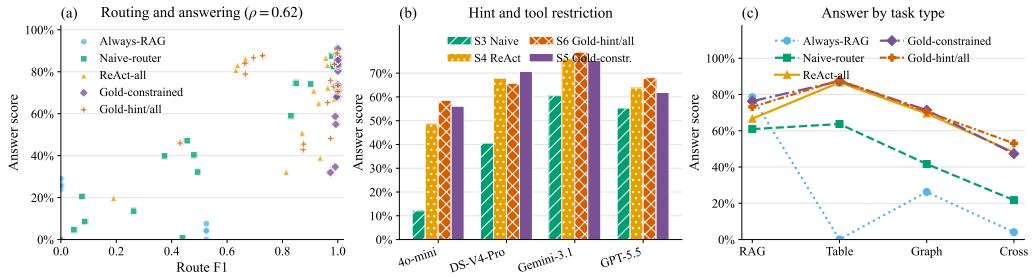

Figure 3: Routing and answering exhibit related but distinct behavior. (a) Route F1 versus Answer across 80 model–setting–task-type buckets ($\rho = 0.62$). (b) Mean Answer for settings that vary routing information and tool access. (c) Mean Answer by task type and setting.

5.5 Post-Routing Bottlenecks

The gap between routing and execution is clearest under S5: all four models obtain 98.7–99.8 Route F1, but Evidence remains 66.7–78.5 and Answer 56.1–75.3. Low S3 Route and Evidence are consistent with selection failures; high Route with incomplete Evidence under S5 indicates acquisition failures; and residual Answer errors after strong Evidence indicate computation or synthesis failures. Because Evidence records artifact access rather than semantic use, these stages are diagnostic categories, not causal proofs. **Near-perfect routing still leaves 24.7–43.9% Answer error.**

Trace behavior supports this separation. Exact-set routing under S5 is 95.9–99.4%, with no unavailable tool family exposed. Under S6, however, 7.8–63.5% of tasks contain at least one unnecessary tool call, depending on the model. The same hint can therefore yield very different execution policies.

5.6 Closed-Book Answerability and Efficiency

No-tool Answer peaks at 2.4%, so S1 is a closed-book diagnostic rather than a contamination test. Its 95.8% mean Efficiency reflects short, evidence-free traces rather than task success and must be interpreted jointly with Answer. Under the common budget defined in §5.1, mean Efficiency is 37.1% for ReAct-all, 35.7% for Gold-hint/all, and 50.2% for Gold-constrained. Surface hints alone do not reduce exploration: S4→S6 changes Efficiency by -0.8 to -3.5 points. Tool restriction does: S6→S5 gains 8.8–22.3 points. **Efficiency gains come from reducing the action space, not merely naming the destination.** Efficiency remains a normalized, tokenizer-dependent diagnostic weighted at 10%; provider-dependent costs are discussed in Appendix D.

6 Conclusion

WorkSurface-Bench separates routing from answering on 1,151 auditable tasks over documents, tables, and dependency graphs. Across four backbones and six settings, routing and Answer are only moderately associated ($\rho = 0.62$). Gold-constrained agents reach 98.7–99.8 Route F1, yet Answer remains 56.1–75.3. The matched S6 condition shows why the distinction matters: surface hints can improve answers, while removing irrelevant tools more reliably improves routing and efficiency. The matched SDK comparison further shows that stronger orchestration can raise Answer while sharply increasing context cost. These interventions are controls, not evidence of autonomous routing ability. The decomposed metrics localize selection, acquisition, and downstream computation failures without treating artifact access as proof of correct reasoning. Agent evaluations should therefore report surface selection separately and test whether a system can convert retrieved evidence into a correct answer. Future work should broaden workspaces and domains, add repeated runs, and study adaptive tool policies. WorkSurface-Bench provides a reproducible basis for tracking these trade-offs as workspace agents and interfaces evolve.

AI USE STATEMENT

We used generative AI tools to generate candidate benchmark questions and to assist with dataset cleaning and reformatting, model-based quality screening, refining experimental hypotheses and methodology, experiment design, implementation and debugging of construction, scoring, and agent code, qualitative error analysis, and interpretation of results. We additionally used them to create and edit scientific figures, survey and summarize related literature, and edit the manuscript's language and structure. Generative AI systems also served as the models under evaluation; these experimental roles are documented separately in Sections 3 and 5. We did not use generative AI to formulate mathematical claims, supply ingredients for proofs, write proofs, translate the manuscript, conduct the human annotation, or supply benchmark gold answers; theoretical-model development and translation were not applicable to this work. Table golds come from executed queries, document golds from verified source spans, and graph golds from source annotations. The authors reviewed all AI-assisted code, analyses, figures, and text and take responsibility for the final content, claims, and artifacts.

ETHICS STATEMENT

The benchmark contains no personal or human-subject data. Its files are public resources or grounded synthetic artifacts inherited from Workspace-Bench-Lite, and human annotators assessed benchmark items rather than contributing personal data. The released source lock supports provenance auditing. Because the benchmark measures performance under a fixed tool contract, its scores should not be treated as deployment guarantees for real enterprise systems.

REPRODUCIBILITY STATEMENT

The benchmark construction and validation procedures are described in §3, the scoring protocol in §4, and the experimental settings in §5.1. Appendix D documents the released code, source lock, trajectories, SDK baselines, and access conditions. The anonymous repository linked in the abstract includes all 1,151 tasks, retained traces, scoring scripts, and tests.

REFERENCES

Wenhu Chen et al. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of EMNLP*, 2020.

Wenhu Chen et al. Open question answering over tables and text. In *ICLR*, 2021.

Zehui Chen et al. T-eval: Evaluating the tool utilization capability of large language models step by step. *arXiv preprint arXiv:2312.14033*, 2023.

Prafulla Kumar Choubey, Xiangyu Peng, Shilpa Bhagavath, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Benchmarking deep search over heterogeneous enterprise data. *arXiv preprint arXiv:2506.23139*, 2025.

DeepSeek-AI. DeepSeek V4 Preview Release. DeepSeek API documentation, 2026. URL `https://api-docs.deepseek.com/news/news260424/`.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, et al. Workarena: How capable are web agents at solving common knowledge work tasks? *arXiv preprint arXiv:2403.07718*, 2024.

Robert Friel et al. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*, 2024.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64 (12):86–92, 2021.

Google. Gemini 3.1 Pro Preview. Gemini API model documentation, 2026. URL `https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview`.

Yue Huang et al. Metatool benchmark for large language models: Deciding whether to use tools and which to use. *arXiv preprint arXiv:2310.03128*, 2023.

Enes Kutay Isgorur, Jan Strich, Maximilian Trescher, Martin Semmann, and Chris Biemann. T2-ragbench: Text-and-table benchmark for evaluating retrieval-augmented generation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*, 2026.

Jinyang Li et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. In *NeurIPS*, 2023a.

Minghao Li et al. Api-bank: A comprehensive benchmark for tool-augmented llms. *arXiv preprint arXiv:2304.08244*, 2023b.

Chang Ma, Junlei Zhang, Zhihao Zhu, et al. Agentboard: An analytical evaluation board of multi-turn llm agents. *arXiv preprint arXiv:2401.13178*, 2024.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence. OpenAI model release, 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

OpenAI. GPT-5.5 System Card. OpenAI system card, 2026. URL https://openai.com/index/gpt-5-5-system-card/.

Yujia Qin, Shihao Liang, Yining Ye, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. Multimodalqa: Complex question answering over text, tables and images. In *International Conference on Learning Representations*, 2021.

Zirui Tang, Xuanhe Zhou, et al. Workspace-bench 1.0: Benchmarking ai agents on workspace tasks with large-scale file dependencies, 2026.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, et al. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *arXiv preprint arXiv:2407.18901*, 2024.

Shirley Wu et al. Stark: Benchmarking llm retrieval on semi-structured knowledge bases. *arXiv preprint arXiv:2404.13207*, 2024.

Chuan Xu, Qiaosheng Chen, Yutong Feng, and Gong Cheng. mmRAG: A modular benchmark for retrieval-augmented generation over text, tables, and knowledge graphs. *arXiv preprint arXiv:2505.11180*, 2025.

Frank F Xu et al. Theagentcompany: Benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, 2018.

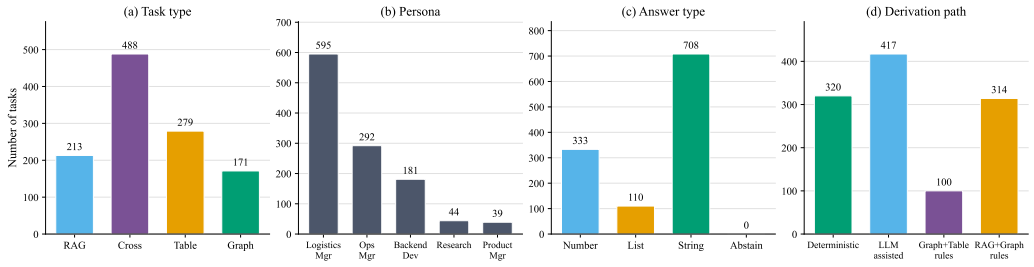

Figure 4: Dataset distributions across (a) task type, (b) persona, (c) answer type, and (d) derivation path. Frequencies are intentionally non-uniform: filtering prioritizes verifiability and strict surface necessity over quotas.

A Limitations

The benchmark covers five personas and three knowledge surfaces derived from Workspace-Bench-Lite, so results may not transfer directly to other enterprise domains. Automated generation can introduce templated language, so we built a 2,000-item candidate pool and retained 1,151 items passing strict checks by at least two of three model auditors. Three annotators independently audited 200 stratified items; all 200 pass all six dimensions by majority vote, but a sample cannot prove that every item is flawless. Persona and source-task frequencies remain uneven. Efficiency uses a fixed GPT-4o-mini reference trace and should be interpreted as a secondary, tokenizer-dependent diagnostic. Broader domains and repeated runs are left to future work.

B Quality Audits and Construction Details

B.1 Rubric-to-Task Conversion

The 100 source tasks carry 1,850 rubrics across four Workspace-Bench rubric types (Basic Evaluation: 450, Outcome Evaluation: 1,030, Process Evaluation: 339, Result Evaluation: 31). Deterministic and proof-carrying generators first produce a 2,000-item candidate pool. GPT-5.5 retains 1,465 items passing answerability, gold correctness, naturalness, atomicity, and leakage checks; independent DeepSeek and Gemini audits then yield 1,151 items passing the strict criteria under at least two of three models.

B.2 Surface Coverage Validation

Each routable surface must support executable gold construction and a non-trivial task set. The final release contains 453 table-bearing items with reproducible DuckDB results, 601 RAG-bearing items with verified source spans, and 600 graph-bearing items whose dependency paths match source annotations. These counts overlap because a cross-surface task contributes to every surface it requires. All five personas and all three surfaces remain represented after screening.

B.3 Gold-Answer Audit

Because we never take a gold answer from LLM free text, we can *execute* or *re-verify* every gold value directly. We audited the whole set: **(i)** all 453 table-bearing items have queries that execute against the DuckDB registry and reproduce their gold results; **(ii)** all 601 RAG-bearing items have an exact evidence span; and **(iii)** all 600 graph-bearing items have valid nodes, paths, or complete neighbor sets. This audit is fully automated and re-runnable from the released code.

B.4 Independent Human Audit

Three annotators independently reviewed the same stratified 200-item sample: 30 Graph, 40 Table, 30 RAG, 25 Graph+Table, 35 RAG+Graph, 25 RAG+Table, and all 15 three-surface tasks. By majority vote, all 200 items are natural, answerable, dependent on every annotated surface, gold-correct,

Table 5: Operational definitions of the six agent settings. All settings use the same question and answer format; only tool exposure and the forced initial behavior differ.

Setting	Exposed surfaces	Initial behavior	Subsequent interaction	Purpose
S1 No-tool	None	Answer directly	No tools	Closed-book answerability baseline
S2 Always-RAG	RAG only	Forced `kb_search`	One-shot answer from returned snippets	Fixed single-surface baseline
S3 Naive-router	One selected surface	Choose exactly one of RAG/ Table/Graph	ReAct restricted to that surface	One-shot surface routing
S4 ReAct-all	RAG, Table, Graph	No forced call	Dynamically interleave exposed tools, up to 8 turns	Unconstrained agent under distractor pressure
S5 Gold-constrained	Required surfaces only	No forced call	Dynamically interleave tools within the required set, up to 8 turns	Hint plus irrelevant-tool removal
S6 Gold-hint/all	RAG, Table, Graph	Required surfaces named in prompt	Dynamically interleave any exposed tools, up to 8 turns	Surface-information intervention

atomic and unambiguous, and free of a leakage cue. Required-surface necessity and atomicity are unanimous on 200/200 items; 192/200 items are unanimous across all six dimensions. Pairwise agreement ranges from 98.7% (naturalness) to 100% (surface necessity and atomicity).

B.5 Distribution

Figure 4 shows the benchmark's cross-cutting distributions. Cross-surface is the largest class (42%), followed by Table (24%), RAG (19%), and Graph (15%); personas span five roles with the operational two dominating, reflecting the Workspace-Bench-Lite source; answer types cover number, list, and string. Derivation paths are intentionally uneven: filtering retains every strictly verified item rather than enforcing equal quotas, while no single path accounts for a majority of the release.

B.6 Closed-Book Answerability Diagnostic

The S1 no-tool setting reaches at most 2.4% Answer over all 1,151 tasks. This result indicates limited closed-book answerability, but it is not a formal contamination test and does not rule out item-level memorization.

B.7 Task Examples

Two illustrative items (abbreviated for space): **cross_surface (LLM-assisted, gold via query execution).** *Question*: "For workspace task 107: what is the total sales amount across the Asia Pacific region?" *Gold*: `800511.75` (number). *Gold evidence*: [surface=rag, file=`t107__report.md`, span="Asia Pacific"; surface=table, table=`t107__asia_pacific_orders`, query=`SELECT ROUND(SUM(sales),2) FROM ...`]. *Required surfaces*: {rag, table}.

C Agent Configuration Details

Table 5 gives the executable distinction among the six settings. S5 and S6 receive only the names of the required surfaces: neither is given the gold query, evidence, tool sequence, or answer. S5 restricts tool exposure to those surfaces, whereas S6 retains all tools. Route scores are computed

Table 6: Off-the-shelf SDK baselines on all 1,151 tasks. The upper panel reports overall metrics (%); the lower panel reports Answer by task type (%). Both runs contain zero protocol errors.

Agent	Backbone	Route P	Route R	Route F1	Evidence	Answer	Eff.	Agg.
Claude SDK	Sonnet 4.6	75.6	99.8	83.8	77.3	71.0	11.1	70.1
Codex SDK	GPT-5.5	79.1	97.5	84.7	74.5	78.5	0.3	71.0

Agent	RAG	Table	Graph	Cross
Claude SDK	83.1	84.0	68.9	59.0
Codex SDK	84.0	88.2	27.9	88.3

from surfaces actually used in the trace, so neither condition is assigned a perfect Route score by construction.

For S3–S6, each ReAct turn emits exactly one JSON object containing either a tool call or `final_answer`. RAG exposes `kb_search`; Table exposes `table_list`, `table_describe`, and read-only `table_query`; Graph exposes entity search, neighbor lookup, and traversal. Invalid JSON receives a format-repair prompt but still consumes a turn. Runs use greedy decoding at temperature 0 and terminate on a final answer or after eight tool turns.

C.1 Statistical and Scoring Robustness

We report paired differences with a 2,000-replicate bootstrap clustered by the originating Workspace-Bench task, so derived items from the same source are resampled together. The intervals cited in §5.3 use this procedure. We also recompute the aggregate under equal, Answer-heavy, and Route-heavy weights. Component metrics and the S4–S6 intervention conclusions are unchanged; some close aggregate rankings do change under equal weights, which is why we treat Route, Evidence, Answer, and Efficiency as the primary outcomes rather than the aggregate alone.

C.2 Off-the-Shelf Agent SDK Baselines

We additionally evaluate Claude Agent SDK with Claude Sonnet 4.6 and Codex SDK with GPT-5.5. Both agents receive the same seven task-scoped tools and at most eight surface-tool calls; shell, unrestricted filesystem, web, sub-agent, and other built-in tools are disabled. Each completes all 1,151 tasks without a protocol error, adding 2,302 trajectories. We report these agents separately because an SDK changes both the orchestration loop and the backbone-level prompt, whereas S1–S6 are controlled interventions within a shared harness. The SDKs are also provider-specific. Codex SDK with GPT-5.5 is directly comparable to the controlled GPT-5.5 ReAct-all run, whereas Claude Agent SDK uses Claude Sonnet 4.6, for which we do not have an S1–S6 counterpart. We therefore treat Claude as an unmatched external baseline and make no cross-framework ranking claim. We do not route every backbone through every SDK: doing so would introduce provider-compatibility layers that can change the wire protocol, tool discovery, and system prompting, confounding the agent implementation with the adapter.

Codex SDK provides the clearest post-routing failure case: on Graph-only tasks, Route F1 is 98.1 and Evidence is 97.7, but Answer is 27.9. Inspection shows both strict output-format failures (for example, a filename set encoded as a JSON string inside a singleton list) and incorrect file counts. The released scorer counts these as Answer failures. SDK-reported end-to-end token usage includes framework-managed context, so its low Efficiency reflects the cost of the complete agent implementation rather than only visible tool calls.

D Reproducibility, Cost, and Access

Software. The release includes source-freezing and surface-projection scripts, deterministic and LLM-assisted task derivation, the scoring package with ten unit tests, and a checkpointed text-based ReAct agent harness.

Runs. Main experimental numbers come from one canonical trajectory per (task, model, setting) through an OpenAI-compatible proxy. The initial sweep used moderate concurrency, then retried failures at lower concurrency. The release contains 27,624 protocol-valid traces for `gpt-4o-mini`, `deepseek-v4-pro`, `gemini-3.1-pro-preview`, and `gpt-5.5`; the two SDK baselines add 2,302 protocol-valid traces.

Cost. Exact monetary cost depends on the proxy and model pricing in effect at run time. Released token counts, retry status, and aggregation scripts permit recalculation under other pricing schedules.

License and access. The derived benchmark inherits Workspace-Bench-Lite's license (see the source repository). Our derived artifacts (task JSONL, canonical KB text, DuckDB parquet registry, surface graphs, scoring code, and agent harness) are released under CC BY 4.0 for data and Apache 2.0 for code. The `wsb_lock.json` freezes the source commit and per-file SHA-256 so any future re-derivation is bit-reproducible.

E DATASHEET

We include an abbreviated Datasheet for Datasets Gebru et al. (2021).

Motivation. WorkSurface-Bench was created to evaluate whether enterprise agents can *route* across heterogeneous knowledge surfaces (documents, tables, dependency graphs), separately from answer correctness. Among the surveyed benchmarks, none reports routing over these three workspace representations as a separate metric.

Composition. Each instance is an atomic question over one persona-scoped workspace, with a gold answer, required surfaces, expected tool types, and evidence annotations. 1,151 instances total; see Table 2 and Figure 4 for distributions.

Collection process. Instances are derived automatically from Workspace-Bench-Lite (English split, commit `60b08b1c`) via the pipeline in §3. Three human annotators independently audited a stratified 200-item sample; the underlying WSB source inherits its authors' expert validation.

Preprocessing. Documents are rendered to canonical UTF-8 markdown; tables are normalized and loaded into a per-workbook DuckDB view registry with provenance columns; the dependency graph is enriched with programmatic edges. The raw source is retained (frozen by hash), so preprocessing is fully reversible.

Uses. Intended: benchmarking enterprise routing agents, ablations over surface selection, and reproducibility research. Not intended: training data (the set is small and constructed from mixed-provenance content), safety-critical decisions, or production deployment gates.

Distribution and maintenance. The versioned dataset and code are hosted at the repository linked in the abstract; issues accept fixes and error reports. Future releases will improve persona coverage and held-out evaluation.

Ethics, privacy, and broader impact. The dataset contains no personally identifying or human-subject data. Files are public resources or grounded synthetic artifacts inherited from Workspace-Bench-Lite. Because some source content is LLM-generated, memorization is possible; the closed-book diagnostic characterizes answerability without claiming to rule out contamination. The benchmark helps test retrieval paths, but scores on scenario-grounded hybrid data should not serve as deployment gates without target-system evaluation.