Generalist Agents for Event Extraction in Aviation Accident Investigations

Anonymous ACL submission

Abstract

Aviation accident investigations produce heterogeneous evidence that must be mapped to standardized safety codes, but fixed extraction pipelines are brittle in this setting. We evaluate whether generalist agents can recover aircraft event sequences from raw aviation accident investigation records without a fixed data preprocessing pipeline. On 127 cases, agents must output occurrence and phase-of-flight code pairs from analyzing raw evidence files. Results show that current models struggle with fine-grained event extraction: the best system reaches 0.39 tuple micro-F1, with errors arising from both incomplete evidence exploration and taxonomy-level reasoning. We propose aviation accident investigation records as a largely unexplored, real-world, safety-critical setting for evaluating generalist agents.

1 Introduction

Each year, aviation safety organizations process large numbers of civil aviation accidents, producing detailed records that require sustained expert analysis. These investigations are intended not to assign blame or liability, but to convert case-specific evidence into reusable safety knowledge for accident prevention: identifying what happened, when it happened, what causal or contributing factors were involved, and how similar events can be avoided (ICAO, 2024; NTSB, 2002). Because these lessons are most useful when compared across many cases and reporting organizations, international bodies such as ICAO and national investigation agencies have developed standardized taxonomies that encode accident and incident information using fixed categories (ICAO, 2026).

These investigations produce evidence artifacts like factual reports, forms, interview transcripts, maintenance records, recorder data, images, and other case-specific materials that are anlayzed by investigators and mapped onto fixed categories with fine-grained, domain-specific boundaries (NTSB, 1999). Although the output can be naturally framed as a classification problem, the input is not straightforward.

This creates a bottleneck for fixed data extraction pipelines (Wang et al., 2024a; Zhang et al., 2026). Because aviation investigations are sociotechnical, the relevant evidence, file formats, and degree of noise vary across cases, making fixed preprocessing and retrieval systems costly and brittle. However, agents equipped with a general computation substrate such as Python may be better suited to this setting because they can inspect available materials, decide what evidence to extract, and construct extraction strategies on demand (Wang et al., 2024b).

This raises our central question: can current generalist agents turn messy accident evidence into standardized codes without a fixed preprocessing pipeline? We evaluate this question on one aspect of aviation accident coding: reconstructing the sequence of aircraft events. We assemble a dataset of 127 NTSB cases from the last decade, stratified across major CICTT category groups. For each case, an agent is placed in a sandbox containing the case evidence and taxonomy instructions, and must output the sequence of events for the aircraft involved in the accident as occurrence-code and phase-of-flight-code pairs. We evaluate outputs against investigator-annotated gold sequences using multiple F1 metrics.

Our results show that fine-grained event extraction is currently challenging even for state-of-the-art models. Error analysis suggests failures in both evidence extraction and taxonomy-level reasoning: agents often do not explore the available evidence fully, and they struggle with the scope and applicability of fine-grained taxonomy categories. These

[0]Data set information and code can be found at https://github.com/aviation-agents/GA-EEAAI.

patterns reveal a gap between current agent capabilities and the requirements of reliable accident coding.

2 Related Work

Direct analysis of raw aviation accident investigation evidence remains largely absent from NLP. Prior aviation-safety NLP work is adjacent: it focuses on textual narratives and conventional supervised methods (Tanguy et al., 2016; Zhang et al., 2021).

Recent agent benchmarks evaluate tool use, code execution, file exploration, multimodal reasoning, and long-horizon task completion (Liu et al., 2025; Mialon et al., 2023; Jimenez et al., 2024; Xie et al., 2024; Zhou et al., 2024). Yet they rarely test these capabilities in safety settings that require handling complex data formats and domain-specific reasoning. Raw aviation accident dockets therefore provide a largely unexplored evaluation setting for generalist agents.

3 Dataset

We use National Transportation Safety Board (NTSB) accident dockets because they provide a rare public source that connects real investigation materials with standardized aviation safety codes. The NTSB investigates U.S. civil aviation accidents and, for investigated cases, releases case records that include both the evidence files assembled by investigators and a final report summarizing their findings (NTSB, 2026; Fielding et al., 2011).

The target codes are stored in an accessible database and annotated using a consistent aviation safety taxonomy. The annotations include internal NTSB categories as well as corresponding CICTT categories. We use the CICTT taxonomy because it is coarser, well-defined, and suitable for our evaluation. It defines 36 standardized occurrence categories, such as abnormal runway contact (ARC) and controlled flight into terrain (CFIT). In our task,

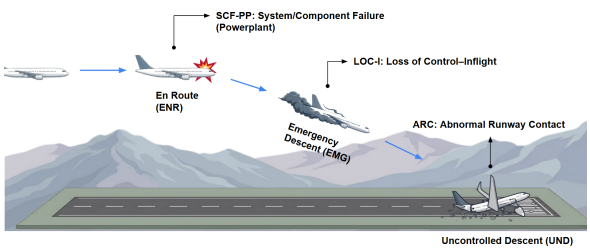

Figure 1: Example coded aircraft sequence of events.

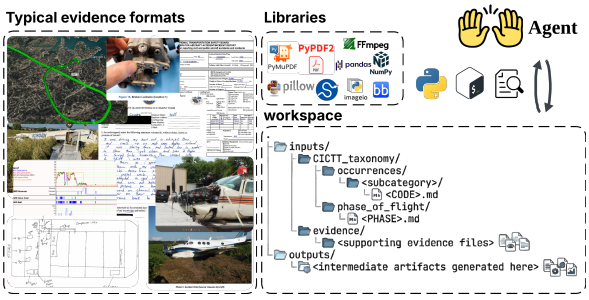

Figure 2: Agent environment and evidence formats.

each event is represented as an occurrence category paired with one of 13 broad phase-of-flight categories, such as takeoff (TOF) and landing (LDG), indicating when that occurrence happened during the aircraft's history of flight (ICAO, 2026).

Our dataset contains 127 cases over the last decade sampled across major CICTT category groups to cover diverse accident types. We limit each case docket to at most 35 files to scope the initial evaluation. Some NTSB cases contain hundreds of files, especially major Class 1 commercial aviation investigations, which are outside the scope of this study. Even within this limit, the evidence may include documents, tables, images, video, audio, geospatial files, and archives, making the dataset a realistic testbed for evaluating whether agents can construct extraction strategies over heterogeneous accident evidence.

4 Agent Setup

Agent Harness. We use the OpenHands SDK as the baseline agent harness because it supports the interactive capabilities required for our evaluation, including file inspection, terminal use, code execution, and iterative reasoning (Wang et al., 2026). For each case, we place the agent in an isolated working directory containing the raw evidence files and a set of markdown files, each corresponding to a taxonomy code and its usage instructions.

Execution Environment. The agent can execute commands and code in a terminal, inspect and edit files, view images, and record observations with a think tool. Because the file editor cannot directly open PDFs or other binary documents, the agent must extract or OCR them into text files or render pages as images before inspection. It can also generate and store intermediate artifacts in an outputs directory. The environment includes Python libraries and system utilities for working with heterogeneous evidence formats.

Model	Tuple	Occurrence		Phase
	Micro-F1	Micro-F1	Macro-F1	Micro-F1
GPT-5.4	$0.3885_{\pm 0.0468}$	$0.4806_{\pm 0.0408}$	$0.4768_{\pm 0.0425}$	$0.5871_{\pm 0.0456}$
↪ gold narrative	$0.6845_{\pm 0.0346}$	$0.6997_{\pm 0.0336}$	$0.7021_{\pm 0.0404}$	$0.9873_{\pm 0.0124}$
Claude Haiku 4.5	$0.3523_{\pm 0.0445}$	$0.4249_{\pm 0.0432}$	$0.4387_{\pm 0.0458}$	$0.5492_{\pm 0.0486}$
GPT-5 Mini	$0.3211_{\pm 0.0400}$	$0.4273_{\pm 0.0368}$	$0.4190_{\pm 0.0405}$	$0.5664_{\pm 0.0471}$
↪ vision disabled	$0.2815_{\pm 0.0452}$	$0.4382_{\pm 0.0465}$	$0.4007_{\pm 0.0493}$	$0.4728_{\pm 0.0540}$
↪ gold narrative	$0.6514_{\pm 0.0375}$	$0.6539_{\pm 0.0374}$	$0.6592_{\pm 0.0423}$	$0.9975_{\pm 0.0040}$
Gemini 2.5 Flash	$0.2169_{\pm 0.0443}$	$0.3952_{\pm 0.0571}$	$0.4019_{\pm 0.0537}$	$0.4145_{\pm 0.0504}$
Gemma 4 31B IT	$0.3473_{\pm 0.0501}$	$0.4251_{\pm 0.0452}$	$0.3911_{\pm 0.0494}$	$0.5479_{\pm 0.0518}$
↪ vision disabled	$0.3368_{\pm 0.0484}$	$0.4106_{\pm 0.0425}$	$0.3805_{\pm 0.0454}$	$0.5318_{\pm 0.0449}$
↪ gold narrative	$0.6074_{\pm 0.0367}$	$0.6125_{\pm 0.0368}$	$0.6066_{\pm 0.0432}$	$0.9886_{\pm 0.0121}$
Gemma 4 26B A4B	$0.2660_{\pm 0.0461}$	$0.3514_{\pm 0.0435}$	$0.3052_{\pm 0.0461}$	$0.4499_{\pm 0.0551}$
Qwen3-VL 32B Thinking	$0.1681_{\pm 0.0415}$	$0.2367_{\pm 0.0477}$	$0.2079_{\pm 0.0444}$	$0.3602_{\pm 0.0530}$
Qwen3-VL 30B A3B Thinking	$0.1258_{\pm 0.0391}$	$0.1821_{\pm 0.0444}$	$0.1721_{\pm 0.0461}$	$0.2583_{\pm 0.0517}$
↪ vision disabled	$0.1192_{\pm 0.0401}$	$0.1523_{\pm 0.0421}$	$0.1274_{\pm 0.0373}$	$0.2715_{\pm 0.0575}$
↪ gold narrative	$0.5980_{\pm 0.0412}$	$0.6107_{\pm 0.0398}$	$0.6108_{\pm 0.0428}$	$0.9898_{\pm 0.0118}$
Qwen3-VL 8B Thinking	$0.0952_{\pm 0.0324}$	$0.1552_{\pm 0.0408}$	$0.1400_{\pm 0.0398}$	$0.2751_{\pm 0.0485}$
Post-knowledge-cutoff runs on 98 cases				
GPT-5 Mini	$0.3411_{\pm 0.0496}$	$0.4147_{\pm 0.0491}$	$0.4090_{\pm 0.0553}$	$0.5349_{\pm 0.0588}$
Qwen3-VL 30B A3B Thinking	$0.0806_{\pm 0.0379}$	$0.1662_{\pm 0.0523}$	$0.1761_{\pm 0.0571}$	$0.1965_{\pm 0.0572}$

Table 1: Baseline and ablation scores from one evaluation run. Rows prefixed with ↪ indicate ablation runs. Subscripted ± values indicate half-widths of 95% confidence intervals over cases.

Prediction Task. The agent is instructed to analyze the evidence and reconstruct the sequence of events for the aircraft involved in the accident and save its prediction as a structured JSON file. Each predicted event must include both the occurrence-category code and the phase-of-flight code.

5 Evaluation

Evaluation Scope. We evaluate predicted event sequences against investigator-anotated codes. Although the annotation order reflects the chronological development of the accident, the coding scheme does not impose global ordering constraints between events. We therefore do not evaluate full ordered-sequence reconstruction or treat the sequence as a causal graph. Instead, our evaluation focuses on whether models recover the expert-selected event codes from raw accident evidence.

Metrics. We report three F1 metrics. Occurrence-code F1 measures recovery of the correct occurrence categories, regardless of phase of flight. Phase-code F1 measures recovery of the correct phase-of-flight categories, regardless of occurrence category. Event-tuple F1 is stricter: a predicted event is correct only when both its occurrence code and phase code match a gold event. Because occurrence categories are not mutually exclusive, models are expected to recover every pertinent coded event rather than only a primary event. These metrics measure agreement with expert annotations rather than absolute ground truth.

Bootstrap Uncertainty and Model Comparisons. Because the evaluation uses a finite sample, point estimates and model gaps may reflect sampling variation rather than stable performance differences. We therefore use case-level bootstrap resampling over the 127 cases, reporting 95% confidence intervals from 10,000 replicates. For model comparisons, we use the same paired bootstrap samples to estimate confidence intervals over differences in F1, treating a gap as statistically supported when the 95% interval excludes zero (Koehn, 2004; Bestgen, 2022; Dror et al., 2018).

6 Results

Baselines. Main results show that state-of-the-art models such as GPT-5.4 achieve the strongest performance, as expected, but the gap over smaller models is relatively modest. Open-source mod-

Model A	Model B	95% CI	Sig.?
GPT-5.4	GPT-5 Mini	$[-0.1093, -0.0257]$	Yes
GPT-5 Mini	Vision disabled	$[0.0013, 0.0788]$	Yes
GPT-5 Mini	Gemma 4 31B IT	$[-0.0725, 0.0199]$	No
Gemma 4 31B IT	Vision disabled	$[-0.0280, 0.0470]$	No
Qwen3-VL 30B A3B Thinking	Vision disabled	$[-0.0393, 0.0543]$	No

Table 2: Bootstrap comparisons on event-tuple micro-F1. Significance is determined by whether the paired 95% confidence interval excludes zero.

Error	Definition
`event_not_grounded`	The agent predicted an event that was not supported by the evidence it inspected. The event appears fabricated, guessed, or inferred from weak or speculative evidence rather than grounded in observed case materials.
`taxonomy_partial_read`	Relevant taxonomy instructions were not consulted.
`evidence_artifact_ignored`	Clearly available evidence was not inspected.
`extraction_error`	An error or unexpected behavior occurred during extraction, such as when running code or executing commands.

Table 3: Error taxonomy used to characterize common agent failures.

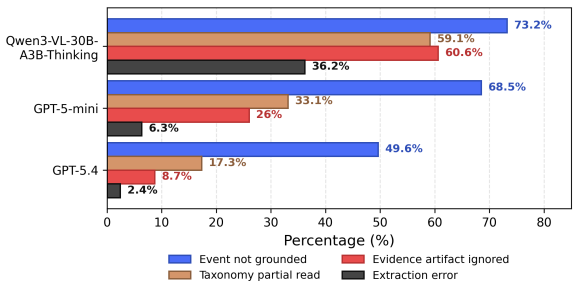

Figure 3: Percentage of errors by model across error categories.

els with vision and tool-use capabilities, such as Gemma4-31B-IT, are also competitive. Overall, point estimates follow a consistent pattern across the three metrics: phase-code F1 and occurrence-code F1 are higher than event-tuple F1. This suggests that models can often identify the relevant categories, but assigning the exact phase of flight to each occurrence remains more difficult.

Our **ablations** suggest that contamination is not a primary concern for this dataset or task. For GPT-5-mini and Qwen3-VL-30B-A3B-Thinking, we evaluate 98 cases published after their knowledge cutoffs and observe performance close to the main runs, suggesting that the task is likely out of distribution. As an additional test, we disable vision and rerun the models on the main 127 cases. Although image views should help agents interpret complex data, plan extraction, and identify cues for classification, disabling vision produces only limited changes, with paired bootstrap comparisons in Table 2 showing a drop for GPT-5-mini but non-significant differences for Gemma 4 31B IT and Qwen3-VL 30B A3B Thinking. This suggests that agents may not be using vision as much as expected in this setup. Finally, to decouple raw evidence extraction from taxonomy classification reasoning, we provide models with gold investigator narratives and factual summaries from the public-facing NTSB final report for each case and ask them to output the event sequence in the same format. Performance improves but remains below ideal, indicating that taxonomy classification reasoning remains a bottleneck even when extraction is simplified. Together, the baseline and ablation results identify three limitations: evidence extraction, limited use of vision capabilities, and taxonomy reasoning.

7 Error Analysis

We conduct a qualitative error analysis by sampling approximately 30 failed execution traces across models and manually reviewing them to identify process errors that led to incorrect outputs. From this analysis, we consolidate recurring failure patterns into the error taxonomy defined in Table 3. We then use LLM-assisted labeling to annotate additional failed traces according to this taxonomy.

Figure 3 shows the approximate distribution of these error types across models. The main trend is that process-level failures decrease as model capability increases, but still persist. This pattern helps explain the evaluation scores and performance differences between models. Improving agent performance on this task therefore requires addressing the full evidence-to-code process, along with the final-label selection.

8 Conclusion

Our evaluation shows that current generalist agents are not reliable enough for converting messy accident evidence into standardized taxonomy codes. Their performance on this multi-step task is limited by process-level failures and domain-specific reasoning errors.

More broadly, this work identifies a practically relevant safety task for evaluating generalist agents beyond software engineering and web-based settings. Similar evidence-to-code problems arise across safety-critical domains, including energy, rail, health care, and space manufacturing. If agents can eventually perform this process reliably, they could make large-scale safety monitoring more practical. The public and largely uncontaminated nature of NTSB aviation accident investigation data makes it a valuable testbed for developing methods and benchmarks that push generalist agents toward complex real-world tasks.

Limitations

Our evaluation focuses on only one small part of aviation accident coding: reconstructing the sequence of aircraft events as occurrence–phase pairs. Full accident coding involves many additional dimensions, including aircraft, environmental, operational, injury, damage, causal, contributing-factor, and metadata fields. These dimensions are outside the scope of this evaluation, so our results should not be interpreted as a complete assessment of an agent's ability to perform full-scale accident coding.

This is also not a deployment-level evaluation. We study whether current generalist agents can extract standardized event sequences from heterogeneous accident evidence in a controlled sandbox. We also do not test multi-run consistency, so the results do not measure how stable each agent's coding decisions are across repeated runs on the same case. Operational use would require additional safeguards, including expert review, calibration against agency-specific procedures, robustness testing across a broader range of accident types and docket sizes, and explicit controls for privacy, security, traceability, and accountability.

Finally, our evaluation depends on investigator-annotated reference labels. Although these labels are produced by domain experts and are the best available target for this task, human annotation can still contain errors, omissions, or reasonable disagreements, especially when evidence is incomplete or taxonomy boundaries are subtle. Reported scores should therefore be understood as agreement with the available annotations rather than an absolute measure of factual correctness.

We do not propose a new modeling method or claim that our task and evaluation scope by themselves solves accident investigation or safety coding. Instead, our goal is to characterize current agent capabilities and failure modes while identifying a new, realistic, safety-relevant evaluation setting for the broader agent and NLP communities.

References

Yves Bestgen. 2022. Please, don't forget the difference and the confidence interval when seeking for the state-of-the-art status. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5956–5962, Marseille, France. European Language Resources Association.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Eric Fielding, Andrew W. Lo, and Jian Helen Yang. 2011. The national transportation safety board: A model for systemic risk management. *Journal of Investment Management*, 9(1).

ICAO. 2024. *Annex 13 to the Convention on International Civil Aviation: Aircraft Accident and Incident Investigation*, thirteenth edition. International Civil Aviation Organization, Montréal, Canada.

ICAO. 2026. Accident/incident data reporting taxonomy. Accessed: 2026-05-24.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? *Preprint*, arXiv:2310.06770.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2025. Agentbench: Evaluating llms as agents. *Preprint*, arXiv:2308.03688.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

NTSB. 1999. *Aviation Coding Manual*. National Transportation Safety Board, Washington, DC.

NTSB. 2002. *Major Investigations Manual*. National Transportation Safety Board, Washington, DC.

NTSB. 2026. The investigative process. Accessed: 2026-05-24.

Ludovic Tanguy, Nikola Tulechki, Assaf Urieli, Eric Hermann, and Céline Raynal. 2016. Natural language processing for aviation safety reports: From classification to interactive analysis. *Computers in Industry*, 78:80–95. Natural Language Processing and Text Analytics in Industry.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui,

Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024a. Mineru: An open-source solution for precise document content extraction. *Preprint*, arXiv:2409.18839.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024b. Executable code actions elicit better llm agents. *Preprint*, arXiv:2402.01030.

Xingyao Wang, Simon Rosenberg, Juan Michelini, Calvin Smith, Hoang Tran, Engel Nyst, Rohit Malhotra, Xuhui Zhou, Valerie Chen, Robert Brennan, and Graham Neubig. 2026. The openhands software agent sdk: A composable and extensible foundation for production agents. *Preprint*, arXiv:2511.03690.

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. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Preprint*, arXiv:2404.07972.

Qintong Zhang, Bin Wang, Victor Shea-Jay Huang, Junyuan Zhang, Zhengren Wang, Hao Liang, Conghui He, and Wentao Zhang. 2026. Document parsing unveiled: Techniques, challenges, and prospects for structured information extraction. *Preprint*, arXiv:2410.21169.

Xiaoge Zhang, Prabhakar Srinivasan, and Sankaran Mahadevan. 2021. Sequential deep learning from ntsb reports for aviation safety prognosis. *Safety Science*, 142:105390.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. *Preprint*, arXiv:2307.13854.

A Appendix Overview

The appendix provides additional context for readers, methodological details needed to recreate the experiments, reference material for interpreting the label space, and qualitative examples that support the main paper. It is organized as follows:

- Appendix B summarizes the paper's contributions, scope, and future directions.
- Appendix C reports compute and implementation details for the agent runs.
- Appendix D describes licenses and access terms for source and redistributed artifacts.
- Appendix E describes the data sources and dataset construction process.
- Appendix F details the occurrence and phase-of-flight taxonomy codes used in the task.
- Appendix G summarizes the agent prompt and required output schema.
- Appendix H presents agent case studies that illustrate common failure modes.

B Contributions, Impact, and Future Directions

This work introduces aviation accident investigation records as a benchmark setting for evaluating generalist agents on realistic evidence-to-code workflows. The key contribution is not a new model, but a task formulation that requires agents to connect heterogeneous investigation evidence with standardized aviation safety taxonomies. By placing agents directly in raw NTSB docket environments, the evaluation exposes failure modes that are difficult to observe in cleaner classification, question-answering, or document-understanding settings.

More broadly, this work pushes agent evaluation toward complex, multi-step reasoning in safety-critical domains. Aviation accident investigation requires agents to inspect heterogeneous evidence, determine which artifacts matter, reconcile partial signals across files, and apply domain-specific coding rules with precision. The results show that current generalist agents still struggle with this kind of evidence-grounded taxonomy following: their failures often arise not only from incorrect labels, but also from incomplete evidence exploration, weak grounding, skipped taxonomy definitions, and limited use of multimodal evidence. This makes the benchmark useful for probing whether agents can support rigorous analytical workflows rather than merely produce plausible summaries.

The broader impact of this work is therefore twofold. For AI research, it provides a public, realistic testbed for studying tool-using agents under conditions closer to high-stakes professional analysis: messy inputs, domain rules, heterogeneous files, and expert-coded targets. For aviation safety and related domains, it points toward a long-term possibility of expert-assistive systems that could help organize evidence, surface relevant events, and support large-scale safety monitoring. We do not view this as replacing investigators or safety analysts; rather, the benchmark helps measure the gap between current agents and the reliability required for such assistance.

These findings also suggest future directions. Agent methods need stronger file triage, adaptive evidence exploration, multimodal inspection, and domain-grounded reasoning over fine-grained taxonomy definitions. Evaluation should move beyond aggregate F1 scores toward more diagnostic measures that separate evidence discovery, evidence interpretation, taxonomy selection, and phase assignment. Larger versions of the benchmark could include more complex investigations, especially cases with much larger dockets, and the task could be expanded to additional coding dimensions such as causal factors, environmental conditions, operational context, damage, and contributing factors.

Overall, we aim to establish aviation accident investigation records as a realistic, safety-critical testbed for generalist agents. This setting highlights concrete capability gaps in evidence exploration, multimodal evidence use, and taxonomy-grounded reasoning, while encouraging broader benchmark development for complex, evidence-driven tasks across safety-critical scientific and technical domains.

C Compute and Implementation Details

For proprietary API models, per-case cost varied by model size and provider pricing. The larger models, GPT-5.4 and Haiku 4.5, cost approximately \$0.40–\$1.00 per case after accounting for cache reads. Smaller API models, including GPT-5-mini and Gemini 2.5 Flash, cost approximately \$0.10 per case. These figures should be interpreted as approximate run costs under our implementation rather than fixed benchmark costs, since API pricing, cache behavior, and retry frequency can vary over time.

Open-source models were served on NVIDIA A100 GPUs. A single full-dataset run for dense models took approximately 20 hours. Smaller mixture-of-experts models were faster, typically requiring approximately 5–10 hours for a full run. The choice of open-source models was largely constrained by the requirements of the task: agents needed both reliable tool-calling behavior and multimodal capabilities for inspecting the evidence files. At the time of evaluation, the pool of open-source models that supported both requirements in an agent harness was limited, so model selection should be interpreted in light of these capability constraints rather than as an exhaustive comparison of all available open-source language models.

Runtime and execution errors were handled according to whether they reflected the evaluated agent behavior. Basic recoverable errors, such as malformed or incorrect tool calls, were handled by OpenHands within the agent loop. If a model did not save the required output file, the case was treated as a no-submission agent process error and the corresponding gold labels were counted as false negatives. In contrast, API failures and model-serving failures outside the agent's control were discarded and retried until a successful run was obtained, since infrastructure reliability is outside the scope of our evaluation.

D Artifact Licenses and Terms

The dataset is constructed from publicly accessible NTSB accident investigation records and CAROL exports, and from ICAO/CICTT taxonomy resources. We use these sources only for research evaluation and do not claim ownership over the underlying source records or taxonomy materials. Third-party software and model artifacts, including OpenHands and the evaluated proprietary and open-source models, are used under their respective licenses or provider terms. We do not redistribute proprietary model weights or third-party source documents unless permitted by their applicable terms.

E Data Source and Dataset Construction

E.1 Data Source

NTSB case records are the product of a staged investigative workflow rather than a single narrative source. During an investigation, specialists gather and analyze heterogeneous evidence such as wreckage observations, maintenance records, flight logs, interviews, weather information, recorder data, photographs, and operational records (NTSB, 2026, 2002). In larger investigations, this work is divided across technical groups, whose factual reports document domain-specific evidence before the Board-level final report synthesizes facts, analysis, findings, probable cause, and safety recommendations (NTSB, 2002; Fielding et al., 2011). The public accident docket therefore represents an intermediate evidentiary layer: it contains the factual reports and supporting materials investigators considered in developing probable cause, while the final report provides the later analytical synthesis. The source is also live rather than static:

NTSB continues to investigate and annotate new aviation cases, adding nearly a thousand records per year. This distinction is important for our task because standardized event codes are downstream abstractions of the investigation. Assigning them requires analysis of the raw evidence and expert judgment about how observed aircraft events and phases of flight should be represented in the taxonomy (NTSB, 1999; ICAO, 2026).

E.2 Evidence Docket Data

The docket files combine several kinds of evidence rather than a single document type. A case may contain narrative reports, investigator forms, scanned exhibits, maintenance and weather records, photographs, maps, tabular attachments, geospatial traces, audio or video, and compressed collections of supplemental material. These artifacts differ not only in file format but also in their evidentiary role: some summarize the accident chronology, while others provide indirect clues about aircraft configuration, environmental conditions, damage, or operational context. As a result, the extraction problem is not merely optical character recognition or PDF parsing, but deciding which artifacts matter for a particular event-code decision and how to reconcile partial signals across them.

This heterogeneity makes fixed preprocessing brittle. Modern document-parsing systems are improving rapidly, but even pipeline-oriented work identifies forms, tables, figures, and other complex page layouts as persistent challenges for reliable extraction (Wang et al., 2024a; Zhang et al., 2026). Aviation dockets compound this difficulty because relevant evidence can appear in different modalities and in different places across cases. A parser tuned for clean reports may miss handwritten forms, embedded tables, image-only exhibits, or attachments whose importance is only clear after reading other files. We therefore treat the docket itself as the input environment, rather than assuming a single canonical text representation produced by a fixed preprocessing stage.

Event grouping	Cases	%
Airborne	28	22.05
Takeoff and Landing	26	20.47
Ground Operations	22	17.32
Miscellaneous	21	16.54
Aircraft	20	15.75
Weather	6	4.72
Non-aircraft-related	4	3.15
Total	127	100.00

Table 4: Assigned-stratum distribution of the 127-case dataset.

E.3 Dataset Construction

We build the dataset from NTSB CAROL exports covering the last decade. The full pool contains 13,583 investigations with structured metadata and sequence-of-events rows, each of which includes NTSB tier names and their CICTT counterparts. We use this information to map the more fine-grained internal NTSB codes to their CICTT equivalents. We then aggregate occurrence codes into seven broad event groupings used only for stratification: Airborne, Aircraft, Ground Operations, Miscellaneous, Non-aircraft-related, Takeoff and Landing, and Weather. Detailed API mappings and dataset construction files are provided in the repository.

Before sampling, we exclude cases with more than 35 docket files, leaving 13,502 eligible investigations and 33,601 mapped events. Sampling targets are based on event-level grouping prevalence rather than case counts, with bias multipliers that reduce the dominance of Miscellaneous events and increase representation of rare Non-aircraft-related events. The resulting SAMPLE127 dataset contains 127 single-aircraft cases drawn without replacement according to these quotas. Cases contain 1–8 coded events, with an average of approximately 3.1 events and 5.8 docket files per case. Table 4 summarizes the distribution.

F Taxonomies

The CICTT taxonomy was developed by the CAST/ICAO Common Taxonomy Team to provide shared definitions for aviation accident and incident reporting. It is intended to help organizations compare safety issues across reporting systems, jurisdictions, and data sources. The taxonomy resources cover several parts of aviation safety coding, including occurrence categories and phase of flight (ICAO, 2026).

The source NTSB annotations are more fine-grained than the label space used in our evaluation. The NTSB-specific fields contain 96 occurrence names and 48 phase names; we were able to map all 96 occurrence names and 47 of the phase names to 36 occurrence and 13 phase CICTT codes.

F.1 Occurrence Codes

The occurrence taxonomy distinguishes broad accident patterns using short codes and category-

specific usage rules. The examples below illustrate the level of definition and boundary guidance:

Loss of Control–Ground (LOC–G). Loss of aircraft control while the aircraft is on the ground.

Usage notes. Use this code only for non-airborne phases of flight, including ground and surface operations. Loss of control may result from a contaminated runway or taxiway, such as rain, snow, ice, or slush. LOC–G may also co-occur with other coded events: for example, a loss of control caused by a powerplant failure, non-powerplant failure, runway incursion, or wildlife encounter should be coded with both LOC–G and the relevant companion category, such as SCF–PP, SCF–NP, RI, or WILD. Do not use LOC–G when a mechanical failure rendered the aircraft uncontrollable. Rotorcraft events during sloping-ground or moving-helideck operations, including dynamic rollover and ground resonance, are included in this category.

Abnormal Runway Contact (ARC). Any landing or takeoff involving abnormal runway or landing-surface contact.

Usage notes. Use this code for events such as hard or heavy landings, long or fast landings, off-center landings, crabbed landings, nose-wheel-first touchdown, tail strikes, and wingtip or nacelle strikes. Gear-up landings are also recorded as ARC; however, if a system or component failure led to the gear-up landing, the event should also be coded under the appropriate system/component failure category. Do not use ARC for runway contact after a loss of control, such as runway contact after takeoff. Gear collapse during the takeoff run or landing roll is not included unless one of the ARC conditions above is met.

F.2 Phase-of-Flight Codes

Phase-of-flight codes identify when an occurrence happened during the aircraft's history of flight. The examples below illustrate how broad phase codes are defined using operational boundaries:

Takeoff (TOF). From the application of takeoff power, through rotation, and to an altitude of 35 feet above runway elevation.

Subphases. Takeoff covers the application of takeoff power through rotation and to 35 feet above runway elevation, or until gear-up selection, whichever comes first. Rejected takeoff covers the period after the decision to abort has been made until the aircraft begins to taxi from the runway.

Usage notes. Landback during rotorcraft operations is considered a rejected takeoff. For rotorcraft, takeoff may include rearward helicopter flight and is also subject to achieving a positive climb gradient and an appropriate takeoff safety speed specified in the Helicopter Flight Manual. For unmanned aircraft systems, TOF includes launch from any system or method, including catapult launch.

Landing (LDG). From the beginning of the landing flare until the aircraft exits the landing runway, comes to a stop on the runway, or when power is applied for takeoff in the case of a touch-and-go landing.

Subphases. Flare covers the transition from nose-low to nose-up attitude just before landing until touchdown. Landing roll covers the period after touchdown until the aircraft exits the landing runway or comes to a stop, whichever occurs first. Aborted landing after touchdown covers an attempt to get airborne after touchdown, whether successful or not, but excludes the takeoff portion of a touch-and-go.

Usage notes. For rotorcraft, LDG includes both vertical and running landings. For unmanned aircraft, LDG includes non-conventional landing or recovery methods, such as use of a cable or parachute, inducing a deep stall, or other recovery methods.

Source. The CICTT taxonomy page provides detailed lists of the developed taxonomies. The two PDF files used for our occurrence and phase codes include the corresponding code lists and usage instructions (ICAO, 2026).

G Agent Prompt and Output Schema

The full agent prompt used in the evaluation is provided in the repository. We summarize its structure here to make the task interface explicit without reproducing the full prompt text.

G.1 Prompt outline.

The prompt defines the agent as an aviation accident investigation assistant whose goal is prevention-oriented event reconstruction, not blame assignment. It then specifies the workspace layout, including accident evidence under `inputs/`, CICTT taxonomy references under `inputs/CICTT_taxonomy/`, and writable intermediate artifacts under `outputs/`. The agent is instructed to inventory all evidence files, inspect each relevant modality, reconstruct a chronological sequence for each aircraft, propose candidate occur-

rence and phase codes, verify every selected code against the taxonomy definition files, and submit only evidence-grounded events.

The prompt also emphasizes taxonomy fidelity and sequencing constraints: agents must not assign codes from memory, must compare definitions when multiple codes are plausible, must preserve chronological order, must produce one sequence per aircraft, and must avoid both speculative overcoding and omission of clearly documented secondary events.

G.2 Final output structure.

Agents submit a single JSON object to `outputs/final_answer.json`. The required structure is:

```
{
  "vehicles": [
    {
      "vehicle_num": 1,
      "events": [
        {
          "sequence_number": 1,
          "occurrence_code": "LOC-I",
          "phase_code": "ENR"
        },
        {
          "sequence_number": 2,
          "occurrence_code": "CFIT",
          "phase_code": "ENR"
        }
      ]
    }
  ]
}
```

G.3 Ablations.

For the vision-disabled ablation, we disable multimodal vision in the underlying OpenHands logic. We also pass agents a separate prompt variant that removes references implying visual-inspection capabilities, so the model is not instructed to view rendered pages, images, or other visual artifacts.

For the gold-narrative evaluation, the model receives a shortened version of the main prompt that removes references to agent capabilities while retaining the domain-relevant and task-specific information. The input also includes the analysis narrative, factual narrative, and history of flight from the official NTSB final report, which investigators create and publish after completing an investigation. These narratives include investigator interpretations, identified causal and contributing factors, and the key factual information deemed relevant for the public-facing report. This setting therefore provides a clean textual input that summarizes the raw evidence files. The taxonomy usage instructions for both occurrence codes and phase-of-flight codes are also included in the model input. This is a zero-shot check: the model outputs the sequence of events in the same format as the main task, and we evaluate it using the same metrics.

For the post-knowledge-cutoff ablation, we use the same sampling script as the main 127-case evaluation, with the same biasing strategy for category-group coverage. We filter the sampling pool to include only cases published or marked complete after May 31, 2025. This date is well after the GPT-5-mini knowledge cutoff and close to the release date of the Qwen3-VL-30B-A3B models evaluated in this study.

H Agent Case Studies

H.1 Case A: Fuel Exhaustion and ATC Runway Conflict

Aircraft and setting. Case ID `WPR24LA212`; Piper PA-31; Lemoore Naval Air Station (NXP), California; 18 June 2024. The docket contains 2 files; the accident caused substantial damage and minor injuries to the two crewmembers. The gold sequence contains 5 coded events.

NTSB narrative. The left-seated pilot stated that she was recently hired with the operator and was new to flying survey operations. The right-seated pilot was training her for survey operations. They filed a flight plan for a 5-hour survey flight and filled the fuel tanks full. After takeoff, the fuel selector was set to the outboard tanks and after 1 hour and 50 minutes, the inboard tanks were selected. After another 2 hours and 20 minutes, the pilot noticed lower-than-expected fuel on the gauges and opted to end the flight early. Approximately 10 minutes after turning back, the engines had intermittent power loss and were sputtering due to low fuel. The pilot made a request to land at a nearby Naval Base and declared an emergency. The controller communicated runway 14L was closed. The pilot approached runway 14R and noticed people and objects on the runway. They were redirected to land on another runway, but the airplane was too low and touched down on grass. The airplane continued the landing roll into a security fence, damaging the fuselage and both wings.

The pilot reported there were no pre-impact mechanical malfunctions or failures that precluded normal operation.

NTSB probable cause. The pilot's improper in-flight decision-making resulted in a total loss of

Sequence	Event tuples
Gold	(FUEL, ENR), (FUEL, EMG), (NAV, APR), (NAV, APR), (USOS, LDG)
GPT-5.4 prediction	(FUEL, ENR), (ADRM, APR), (ATM, APR), (USOS, LDG)
GPT-5-mini prediction	(FUEL, ENR), (FUEL, APR), (RI, APR), (RE, LDG), (CTOL, LDG)

Table 5: Gold and predicted event sequences for Case A.

Dimension	GPT-5.4	GPT-5-mini
Evidence extraction	Image rendering plus vision for all pages	Image rendering plus vision for all pages
Taxonomy grounding	10 definitions verified	0 definitions read
Intermediate reasoning	6 structured reasoning steps	No explicit reasoning steps
Evidence breadth	10 images viewed	9 images viewed
Total tool calls	56	25
Tuple match	2/5	1/5
Occurrence match	2/5	2/5
Phase match	4/5	4/5
Key failure mode	NAV → ADRM/ATM substitution	Unsupported runway and obstacle codes without taxonomy checking

Table 6: Process-level contrast for Case A.

engine power due to fuel exhaustion and a subsequent emergency landing.

H.1.1 GPT-5.4 Trace

GPT-5.4 completed 57 steps. It viewed 10 rendered report images and read 10 taxonomy definitions.

Agentic flow. The model first inventoried all inputs and sampled the taxonomy while doing so. Because the main PDF was scanned, it decided to render the report pages as images and view them with vision. It then synthesized the narrative from page 7, including the fuel timeline, diversion decision, and runway confusion. The model also rendered the photo attachment, observed the wreckage position off the runway at the fence, verified 10 CICTT occurrence definitions and 6 phase definitions, and considered but rejected CTOL, LOC-G, RE, and SCF-PP. It selected FUEL → ADRM → ATM → USOS, wrote `final_answer.json`, and called finish.

Post hoc analysis and key error. In our reading of the trace, the output captured the fuel event and off-runway landing. The submitted sequence assigned the runway-closure and runway-obstruction evidence to ADRM and ATM rather than to the two NAV codes in the gold sequence. The trace also shows taxonomy checks for CTOL and RE, which were not included in the final answer. The main error was the use of aerodrome and ATC categories for events coded as navigation errors in the gold sequence.

H.1.2 GPT-5-mini Trace

GPT-5-mini completed 26 steps. It viewed 9 rendered report images and read no taxonomy definitions.

Agentic flow. The model listed the input files, inspected file metadata, extracted empty text from the scanned NTSB report PDF, and extracted text from the photo PDF. It rendered the nine report pages to PNG and viewed each page through the file editor. It skipped all CICTT taxonomy definition files, created `outputs/final_answer.json` directly after image review, and submitted FUEL → FUEL → RI → RE → CTOL without verifying any code definitions.

Post hoc analysis and key error. The trace shows review of the same nine report-page images but no consultation of the taxonomy definitions. The final answer includes the fuel event, then assigns RI, RE, and CTOL to the runway and fence evidence. These codes do not match the case evidence: the accident aircraft was the landing aircraft, it was not on a runway before leaving it, and the fence contact occurred after touchdown.

H.2 Case B: Night VFR into IMC and Controlled Flight into Terrain

Aircraft and setting. Case ID CEN23FA187; Cessna 172D; cross-country flight from Newton, Kansas, to Glen Ullin, North Dakota; 14 May 2023. The docket contains 12 files: 11 PDF documents and 1 MOV security-camera video. The gold sequence contains 2 coded events.

Sequence	Event tuples
Gold	(UIMC, MNV), (CFIT, MNV)
GPT-5.4 prediction	(CFIT, ENR)
Gemma 4 31B prediction	(UIMC, ENR), (FUEL, ENR), (CFIT, ENR)

Table 7: Gold and predicted event sequences for Case B.

Dimension	GPT-5.4	Gemma 4 31B
Evidence extraction	Extracted all 11 PDFs and processed video metadata plus brightness profile	Extracted 5 of 11 PDFs and did not process video
Taxonomy grounding	11 definitions read	9 definitions read
Intermediate reasoning	2 structured reasoning steps	1 structured reasoning step
Evidence breadth	11 PDFs plus 1 MOV video	5 PDFs, no video
Total tool calls	42	18
Tuple match	0/2	0/2
Occurrence match	1/2	2/2
Phase match	0/2	0/2
Key failure mode	Strict UIMC interpretation despite documented IMC conditions	Extra FUEL event from carburetor-icing evidence

Table 8: Process-level contrast for Case B.

NTSB narrative. The pilot departed on the cross-country flight with instrument meteorological conditions (IMC) reported over the entire region along the route of flight. The NWS had multiple forecasts calling for instrument flight rule conditions as well as an AIRMET for the threat of low ceilings and low visibility over the area. There was no record of the pilot having received a weather briefing and no flight plan was filed. In addition, there were no known communications with the pilot during the flight. Witnesses reported hearing the airplane flying at a low altitude beneath the clouds just before the accident. They reported that the weather at the time was foggy, misty, and raining. The airplane was substantially damaged when it impacted the terrain in a slight nose-down, wings-level attitude.

Postaccident examination of the airframe and engine did not reveal any anomalies that would have prevented normal operation of the airplane. The airplane was operating in conditions conducive to serious carburetor icing; however, the witnesses reported the airplane was loud when they heard it flying overhead just before impact.

The pilot's autopsy revealed coronary artery disease; however, there was no clear evidence that the disease contributed to the accident.

Weather forecasts for the area called for precipitation and overcast ceilings at 300 ft. The forecasts were consistent with witness accounts of the weather at the time of the accident. The nighttime IMC conditions would have made visual navigation next to impossible without a discernable horizon.

Based on the accident site, wreckage examinations, and the witness accounts, the pilot was likely attempting to fly beneath the low clouds when the airplane impacted the ground.

NTSB probable cause. The pilot's continued VFR flight into instrument meteorological conditions at night and failure to maintain adequate altitude resulted in controlled flight into terrain.

H.2.1 GPT-5.4 Trace

GPT-5.4 completed 85 steps. It read all 11 PDFs, processed the video, and read 11 taxonomy definitions.

Agentic flow. The model listed all 12 docket files and sampled the taxonomy directory. Because the PDFs contained extractable text, it read all 11 PDF documents directly without image rendering. It then attempted to process the MOV security-camera video several times using `cv2`, `ffprobe`, and `ffmpeg`. These attempts appeared to recover some video metadata and a rough temporal brightness profile, but the extracted signal was limited and should be treated cautiously. The model read 6 occurrence definitions, including `UIMC`, `CFIT`, `LOC-I`, `SCF-PP`, `LALT`, and `AMAN`, and 5 phase definitions. In its reasoning steps, it accepted `CFIT`, rejected `UIMC` because the evidence did not explicitly document pilot instrument qualification or aircraft IMC equipment status, also rejected `SCF-PP`, `FUEL`, and `LOC-I`, and submitted only `CFIT/ENR`.

Post hoc analysis and key error. The trace covers all documents and includes limited video processing, but the final answer appears to impose

an overly strict evidentiary requirement for UIMC: explicit documentation of pilot instrument qualification or aircraft IMC equipment status. Under that interpretation, UIMC was omitted despite documented IMC conditions and VFR operation. The output matches CFIT but misses UIMC.

H.2.2 Gemma 4 31B Trace

Gemma 4 31B completed 19 steps. It read 5 of the 11 PDFs, did not process the video, and read 9 taxonomy definitions.

Agentic flow. The model extracted text from 5 PDFs: the accident-site examination memorandum, witness-statement summaries, owner-interview summary, security-camera memorandum, and weather factual report. It did not process the MOV video and did not read the remaining 6 PDF documents. It read 9 occurrence and phase definitions, including UIMC, CFIT, FUEL, and LOC-I. In its reasoning step, it treated documented sub-IFR conditions during a VFR flight as support for UIMC, treated the carburetor-icing evidence as support for FUEL, and treated the wings-level, nose-down terrain impact as support for CFIT. It submitted UIMC/ENR $\rightarrow$ FUEL/ENR $\rightarrow$ CFIT/ENR.

Post hoc analysis and key error. The final answer matches both gold occurrence codes, with UIMC supported by the documented weather and VFR operation. However, it also includes an extra FUEL event based on the carburetor-icing evidence, even though the investigation did not establish a fuel or power-loss event.

I Acknowledgment of AI Assistance

We used AI and LLM-based tools for limited language editing, low-novelty drafting support, literature-search assistance, and help writing or debugging code and LaTeX. The authors verified all content, citations, experiments, and claims, and are responsible for the final paper. AI tools were not listed as authors and were not used as sources of independent research contributions.