A2P-Vis: an Analyzer-to-Presenter Agentic Pipeline for Visual Insights Generation and Reporting

Shuyu Gan[*] Renxiang Wang[†] James Mooney[‡] Dongyeop Kang[§]

University of Minnesota

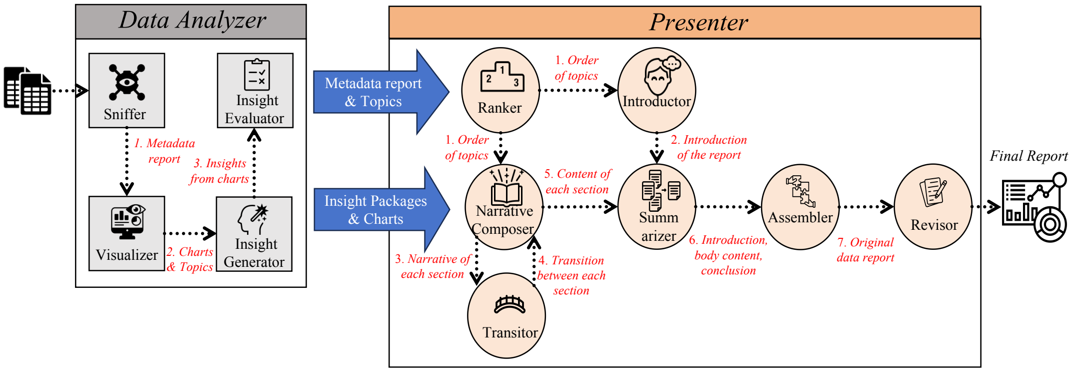


Figure 1: The workflow of ***A2P-Vis***.

Abstract

Automating end-to-end data science pipeline with AI agents still stalls on two gaps: generating insightful, diverse visual evidence and assembling it into a coherent, professional report. We present ***A2P-Vis***, a two-part, multi-agent pipeline that turns raw datasets into a high-quality data-visualization report. The **Data Analyzer** orchestrates profiling, proposes diverse visualization directions, generates and executes plotting code, filters low-quality figures with a legibility checker, and elicits candidate insights that are automatically scored for depth, correctness, specificity, depth and actionability. The **Presenter** then orders topics, composes chart-grounded narratives from the top-ranked insights, writes justified transitions, and revises the document for clarity and consistency, yielding a coherent, publication-ready report. Together, these agents convert raw data into curated materials (charts + vetted insights) and into a readable narrative without manual glue work. We claim that by coupling a quality-assured Analyzer with a narrative Presenter, A2P-Vis operationalizes co-analysis end-to-end, improving the real-world usefulness of automated data analysis for practitioners. For the complete dataset report, please see: `https://www.visagent.org/api/output/f2a3486d-2c3b-4825-98d4-5af25a819f56`.

Index Terms: Multi-agent systems, Large Language Models, Data-science automation, Insight generation, Report synthesis

1 Introduction

LLM-based AI agents have recently been applied to automate data-centric workflows, from early-stage pipeline construction such as Google's Data Science Agent [1] and related LLM-based systems [2, 9, 3] to domain-specific applications in genomics [4], machine learning benchmarking [5] and healthcare [6]. Beyond these pipelines, recent systems such as [7, 11, 10] highlight the emerging role of LLMs in structuring, verifying, and contextualizing insights for visual analytics. Yet, two persistent gaps remain: (i) producing diverse, evidence-rich visualizations with non-trivial insights, and (ii) assembling these materials into a coherent, professional report. We introduce A2P-Vis, a two-part, multi-agent workflow designed to close both gaps, as described in Figure 1. Our design is motivated by how data scientists carefully inspect initial seed data and come up with final insights from trial and errors (pick the best one among generated candidates). Our contributions are as follow:

1. **Data Analyzer** explores possible visualization directions, generates candidate plots, filters low-quality charts, finds the best insight from candidates and evaluates insights.
2. **Presenter** sequences topics, writes chart-grounded narratives with reasoned transitions, makes conclusions, and polishes the prose as a coherent visual story of findings.

2 System Overview

Now we dive into each part of the framework.

2.1 Data Analyzer

This module ingests the dataset and outputs a metadata report, a diverse set of visualization topics, executable plots, and structured insights with scores. As shown in Figure 2, the process is demonstrated on a real dataset with grounded outputs at each stage.

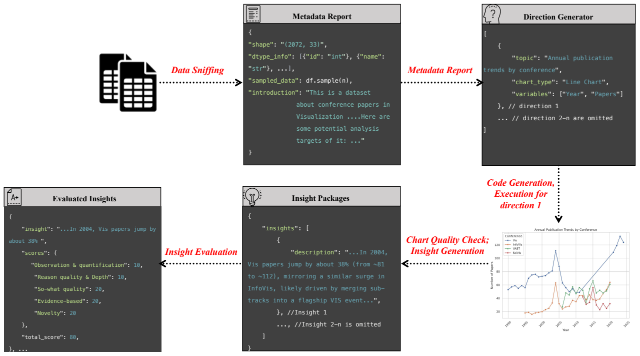
Sniffer The *Sniffer* performs lightweight dataset profiling to establish a reliable contract for downstream agents. It inspects the raw table to extract shape, column names and inferred types. From these basic features and randomly sampled data it produces a concise *metadata report*, which introduces the dataset, enumerates its attributes, and outlines plausible analysis themes. This report guides the *Visualizer* to find visualize directions and use valid columns and sensible encodings for code generation, reducing hallucinations and preventing routine failures (e.g., empty plots, degenerate scales).

[*] e-mail: gan00067@umn.edu

[†] e-mail: renxiang428@gmail.com, This work was done during Renxiang's internship at University of Minnesota.

[‡] e-mail: moone174@umn.edu

[§] e-mail: dongyeop@umn.edu



```
{
"shape": "(2072, 33)",
"dtype_info": [{"id": "int"}, {"name":
"str"}, ...],
"sampled_data": df.sample(n),
"introduction": "This is a dataset
            about conference papers in
            Visualization ....Here are
            some potential analysis
            targets of it: ..."
}
```

```
[
   {
      "topic": "Annual publication
trends by conference",
      "chart_type": "Line Chart",
      "variables": ["Year", "Papers"]
   }, // direction 1
   ... // direction 2-n are omitted
]
```

```
{
   "insights": [
      {
         "description": "...In 2004,
Vis papers jump by about 38% (from ~81
to ~112), mirroring a similar surge in
InfoVis, likely driven by merging sub-
tracks into a flagship VIS event...",
      }, //Insight 1
      ..., //Insight 2-n is omitted
   ]
}
```

```
{
   "insight": "...In 2004, Vis papers jump by
about 38% ",
   "scores": {
      "Observation & quantification": 10,
       "Reason quality & Depth": 10,
       "So-what quality": 20,
       "Evidence-based": 20,
       "Novelty": 20
   },
   "total_score": 80,
}, ...
```

Figure 2: Data Analyzer in Action (From metadata profiling to graded insights).

Additionally, it addresses context and cost: rather than streaming the full dataset into the model, downstream components consume the compact profile, which is sufficient for planning visualizations and generating code without exposing raw records. In practice, the Sniffer's output serves as a schema contract and quality gate, improving the robustness of code generation and keeping subsequent analyses aligned to a consistent data view.

Visualizer Given the metadata report, the *Visualizer* turns profile-level signals into reliable figures through a tightly coupled, four-step flow. First, the *direction generator* proposes concrete analysis targets and topics, emitting machine-readable guidance for each direction (`topic`, `chart_type`, `variables`). Next, the *code generator* compiles that guidance into directly executable script. The script is then executed by the *executor*, which logs outcomes and, upon failure. If errors are raised, it will invoke the *rectifier* to repair the code strictly according to the error trace and then execute it again. Finally, each figure passes through a chart judger that validates quality one by one, checking for meaningless charts.

Insight Generator & Evaluator. Starting from each quality gated figure, the *Insight Generator* produces 5–7 candidate insights using a *three sentence* structure: (1) an observation with chart evidence and an approximate effect size; (2) a hedged, plausible reason anchored in chart context or domain context; and (3) a "so what" that delivers either a concrete next step, a short horizon prediction, or a precise implication. The *Insight Evaluator* then scores each insight with an integer rubric aligned to this template across four criteria: *Correctness & Factuality*, *Specificity & Traceability*, *Insightfulness & Depth*, and *"So what" Quality*. It ranks candidates by total score and returns the top 3 per chart as the final output.

2.2 Presenter

This module takes the outputs of the *Data Analyzer* (the metadata report, visualization topics, and quality-gated charts with scored insights) and assembles them into a publication-ready data visualization report. Following an "overview→ sections→ summary" sequence, it orders topics, drafts the introduction, writes chart-grounded sections with brief transitions, summarizes key takeaways, and finalizes the document with a revision pass. Figure 3 illustrates the workflow and the report layout.

Ranker Given an unordered set of topics from the Data Analyzer, the *Ranker* determines a coherent sequence for narration by analyzing relationships among topics, such as shared variables, temporal order, or thematic similarity. It outputs an ordered list of topics that provides a logical flow for the report, serving as the backbone for the *Introductor* and subsequent modules.

Introductor Given the ranked topic sequence and the metadata report, the *Introductor* drafts the opening of the report and sets the roadmap. It first presents the dataset's basic properties such as size, time span, and key attributes, grounded in the metadata. It then previews the analysis in the given order, offering one–two sentence teasers for each forthcoming topic (what it examines and why it matters). The result is a concise outline that bridges the global context to the individual topic sections and hands off naturally to the *Narrative Composer*.

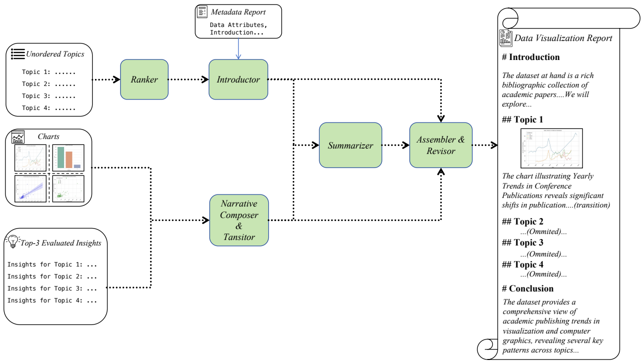

Metadata Report
Data Attributes,
Introduction...
Unordered Topics
Topic 1: ......
Topic 2: ......
Topic 3: ......
Topic 4: ......
Ranker
Introductor
Charts
Summarizer
Assembler & Revisor
Narrative Composer & Tansitor
Top-3 Evaluated Insights
Insights for Topic 1: ...
Insights for Topic 2: ...
Insights for Topic 3: ...
Insights for Topic 4: ...
Data Visualization Report
# Introduction
The dataset at hand is a rich bibliographic collection of academic papers....We will explore...
## Topic 1
The chart illustrating Yearly Trends in Conference Publications reveals significant shifts in publication....(transition)
## Topic 2
...(Ommited)...
## Topic 3
...(Ommited)...
## Topic 4
...(Ommited)...
# Conclusion
The dataset provides a comprehensive view of academic publishing trends in visualization and computer graphics, revealing several key patterns across topics...

Figure 3: Presenter in Action(From ranked topics to a narrative report).

Narrative Composer & Transitor The *Narrative Composer* integrates the top three evaluator scored insights for each chart into a coherent, chart grounded subsection. It follows a claim to evidence to implication pattern, cites the figure explicitly, retains approximate magnitudes and axis references, and standardizes terms and units across sections. The *Transitor* then adds one or two bridge sentences that connect the current section to the next by referencing shared variables, time windows, or meaningful contrasts, improving continuity without inventing links.

Summarizer Building on the introduction and per-topic narratives, the *Summarizer* produces an executive summary and closing conclusions: it consolidates key findings by topic, highlights cross-topic links or contrasts. The summary ties back to the preceding sections, echoing the main points to provide a cohesive close.

Assembler The *Assembler* compiles the introduction, section content, and summary into a complete Markdown report. It applies a consistent heading hierarchy, embeds figures at their designated locations with captions. It also generates elements such as date, headers, footers, and page numbers to standardize the report format.

Revisor Given the assembled Markdown draft, the *Revisor* applies a multi-pass, chain-of-thought [8] style revision, normalizing structure, smoothing transitions, polishing narrative and wording, and enforcing style consistency to produce the final report.

3 Conclusion

We introduce ***A2P-Vis***, an analyzer-to-presenter agentic pipeline for producing coherent, publication-ready data visualization reports. The *Data Analyzer* profiles metadata, generates and executes visualization directions, and yields quality-gated charts with evaluator-scored insights; the *Presenter* ranks topics, drafts the introduction, composes chart-grounded sections with light transitions, summarizes key takeaways, and finalizes the document via revision. This design emphasizes verifiable insight (structure + rubric), diversity in visualization directions and end-to-end reliability from raw data to narrative report.

Acknowledgments

The authors wish to thank Qianwen Wang for discussions and valuable suggestions during development of this work.

References

[1] Google Labs. Data science agent. https://labs.google.com/code/dsa. Accessed: 2025-08-18. 1

[2] S. Guo, C. Deng, Y. Wen, H. Chen, Y. Chang, and J. Wang. DS-agent: Automated data science by empowering large language models with case-based reasoning. In *Proceedings of the 41st International Conference on Machine Learning*, vol. 235 of *Proceedings of Machine Learning Research*, pp. 16813–16848. PMLR, July 2024. 1

[3] S. Hong, Y. Lin, B. Liu, B. Liu, B. Wu, C. Zhang, C. Wei, D. Li, J. Chen, J. Zhang, J. Wang, L. Zhang, L. Zhang, M. Yang, M. Zhuge, T. Guo, T. Zhou, W. Tao, X. Tang, X. Lu, X. Zheng, X. Liang, Y. Fei, Y. Cheng, Z. Gou, Z. Xu, and C. Wu. Data interpreter: An LLM agent for data science, 2024. doi: 10.48550/arXiv.2402.18679 1

[4] H. Liu, S. Chen, Y. Zhang, and H. Wang. Genotex: An LLM agent benchmark for automated gene expression data analysis, 2024. doi: 10.48550/arXiv.2406.15341 1

[5] H. Liu, S. Huang, J. Hu, Y. Zhou, and C. Tan. Hypobench: Towards systematic and principled benchmarking for hypothesis generation, 2025. doi: 10.48550/arXiv.2504.11524 1

[6] M. A. Merrill, A. Paruchuri, N. Rezaei, G. Kovacs, J. Perez, Y. Liu, E. Schenck, N. Hammerquist, J. Sunshine, S. Tailor, K. Ayush, H.-W. Su, Q. He, C. Y. McLean, M. Malhotra, S. Patel, J. Zhan, T. Althoff, D. McDuff, and X. Liu. Transforming wearable data into health insights using large language model agents, 2024. doi: 10.48550/arXiv.2406.06464 1

[7] F. Wang, B. Wang, X. Shu, Z. Liu, Z. Shao, C. Liu, and S. Chen. Chartinsighter: An approach for mitigating hallucination in time-series chart summary generation with a benchmark dataset, 2025. 1

[8] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 24824–24837, 2022. 2

[9] W. Zhang, Y. Shen, W. Lu, and Y. Zhuang. Data-copilot: Bridging billions of data and humans with autonomous workflow, 2023. doi: 10.48550/arXiv.2306.07209 1

[10] Y. Zhao, J. Wang, L. Xiang, X. Zhang, Z. Guo, C. Turkay, Y. Zhang, and S. Chen. Lightva: Lightweight visual analytics with llm agent-based task planning and execution. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):6162–6177, Sept. 2025. doi: 10.1109/tvcg.2024.3496112 1

[11] Y. Zhao, Y. Zhang, Y. Zhang, X. Zhao, J. Wang, Z. Shao, C. Turkay, and S. Chen. Leva: Using large language models to enhance visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 31(3):1830–1847, Mar. 2025. doi: 10.1109/tvcg.2024.3368060 1