What Happened to Automated Visualization?An Agentic Analysis

Weiyao Meng* University of Nottingham
Yuhan Guo† Peking University
Zefei Xie‡ University of Nottingham
Kai Xu § University of Nottingham

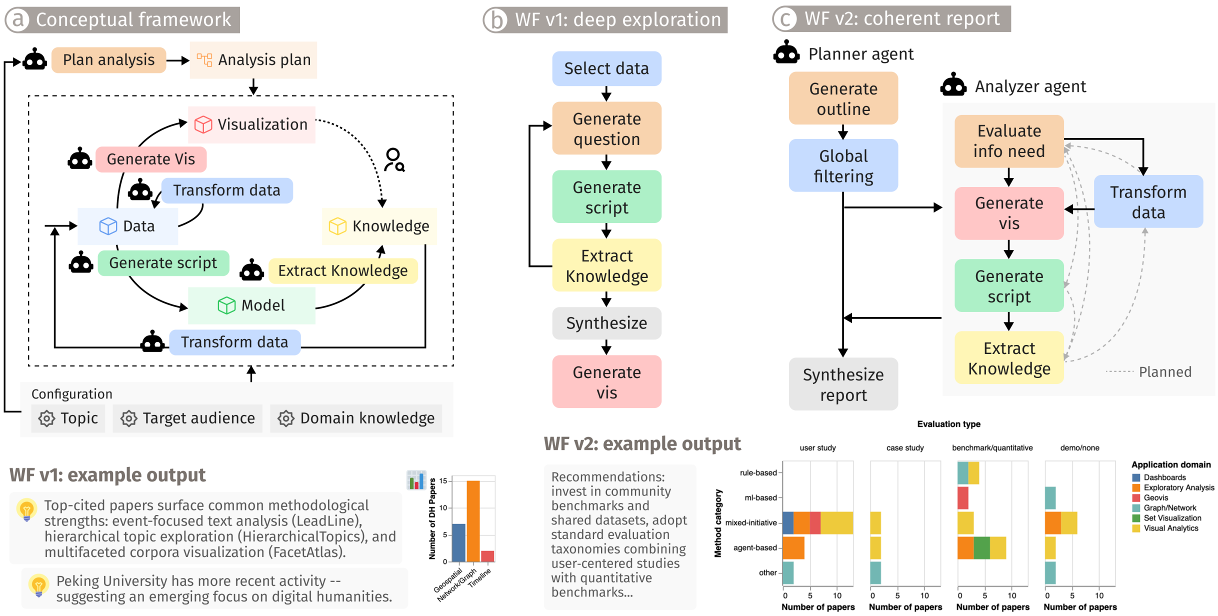

Figure 1: AgenticInsight is an agentic system driven by large language models (LLMs) that automatically analyzes a given dataset and generates a visualization report. The design of the system is guided by a conceptual framework (a) adapted from the visual analytics process model [2]. To implement the framework, the AgenticInsight system underwent iterative development. The first version (b) attempts deep analysis through an exploratory loop, while the second version (c) integrates a global planner module to ensure the coherence of the produced report.

Index Terms: AI agent, visualization, large language models, automated analysis

1 Introduction

Recently, foundation models, particularly large language models (LLMs), have shown great capabilities across a wide variety of tasks that were previously done by human, such as code generation and analytic reasoning. Such advances open new possibilities for further automating the data analysis process. This technical report presents the AgenticInsight system [1] developed for the Agentic VIS Challenge 2025 [2].

AgenticInsight is an automated system that uses an agentic workflow to generate insights and visualizations from a given dataset. The agentic workflow is designed on the basis of a conceptual framework adapted from the visual analytics process model [2]. Powered by the LLMs, the agent automates the essential steps in the visual analytics workflow, including planning, data transformation, visualization generation, analysis code generation, and knowledge extraction. Our workflow is designed to be generalizable and not tailored to a specific dataset. Meanwhile, it allows human users to specify high-level intentions, preferences, and domain knowledge in a separate configuration file, which is incorporated to guide the analytical process. As a result, our workflow can be applied to a range of analysis questions and domains. This report details the system design and decisions, demonstrates the system's performance and generalizability, and presents reflections on building an automated LLM-powered analysis system.

*e-mail: weiyao.meng2@nottingham.ac.uk
†e-mail: yuhan.guo@pku.edu.cn
‡e-mail: scyzx4@exmail.nottingham.ac.uk
§e-mail: kai.xu@nottingham.ac.uk

[1]Project repository available at https://github.com/Vis4Sense/agentic-vis-2025

[2]Challenge website: https://www.visagent.org/

2 AgenticInsight System

2.1 Conceptual Framework

In a visual analytics process, visualization is tightly integrated with automatic data analysis methods, which enables human users to gain knowledge from the analysis results and improve the computational models [2]. Our system is built on a similar process, with data, visualization, model, and knowledge serving as the key components of the analysis (Fig. 1a). The agent actively engages in the intermediate analysis steps (i.e., generating visualization and data analysis models, transforming data, and extracting knowledge), while the interpretation of the visualization is left to human analysts when reviewing the generated report. In addition, the AI agent generates analysis plans to direct the analytic process. The automatic workflow is initiated with a configuration provided by the human user, where the user may specify the topic of analysis, target audience of the report, and domain knowledge related to the dataset.

2.2 Workflow

Our final workflow (Fig. 1c) consists of two agents: (1) a planner agent that proposes an outline of the report and the corresponding analysis tasks that need to be conducted to complete the report, (2) an analyzer agent that carries out the proposed tasks.

Planner agent. The planner agent reasons upon injected analysis intention (the configuration in Fig. 1a) to formulate a structured report outline. For each section, it specifies the analysis design, detailing what to analyze and how to approach the task. After that, the Global Filter extracts a topic-relevant sub-dataset, which can be used, along with the original dataset, by the analyzer agent. This ensures consistency across all analyses while maintaining access to the original dataset for comprehensive data coverage. The planner agent calls the analyzer agent to conduct each task and finally synthesizes the results into a structured HTML report.

Analyzer agent. The process begins with the evaluation of the agent's confidence and the information need (whether data transformation or exploration is needed) to perform the analysis. For tasks that are relatively simple (e.g., identify the most cited paper), the agent adopts an efficient sequential workflow to generate visualization and knowledge. When the task is complex, such as network analysis, the agent needs to transform the data before the generation. Conceptually, the agent may also explore the data by extracting preliminary knowledge to inform data transformation and conduct iterative analysis.

2.3 Implementation

Most nodes in the workflow are implemented through prompting the LLMs, except for *Transform data* and *Generate vis* where the agent may choose between calling the LLM to generate the code or using a pre-defined tool for constructing and visualizing network data. Here we highlight the key technical components and our considerations.

Structured analysis. Both the planner agent and the analyzer agent operate on a structured state. The planner agent maintains the structured report plan and the intermediate results from the analyzer agent. The state of the anlayzer agent includes data, visualization, model, and knowledge (Fig. 1a). The analysis plan consists of a hierarchy of analysis goals (general topic, section theme, and specific task). Each task is further structured with the analysis question, primary and secondary data attributes, required transformation, and expected types of insights.

Transformation. As the visualization libraries such as Vega-Lite and Altair provide built-in functions for data transformation when producing the visualization, we implemented the node *Evaluate info need* in a simplified way, e.g., the agent uses an LLM to determine whether to perform network analysis. If the task is identified as network analysis, the agent uses a pre-defined tool that constructs the network data.

Code generation. The agent uses LLMs to generate python code for both the visualization (via the Altair library) and the analysis script (using libraries such as Pandas, Numpy, and NetworkX). We have tried providing the agent with a set of manually defined visualization or analysis tools but found it could limit the diversity in the produced visualizations and knowledge, as well as the generalizability of the analysis agent. We used Altair (the python version of Vega-Lite) instead of the native Vega-Lite because this allows us to pre-execute the code in the Python environment, thus enabling the incorporation of an error-handling mechanism. The agent asks the LLM to fix the code if an error occurs. The same mechanism is applied to the generation of the analysis script.

Knowledge extraction. In this phase, the agent executes the script generated in the previous step in a sandbox and collects the outputs, which primarily consist of data facts and findings (in the script generation step, the LLM is prompted to write a code that prints the findings during the analysis).

2.4 Iterative Development

Fig. 1b shows an earlier version of our agent, that targets deep exploration through an iterative workflow. This workflow generates very detailed and informative insights, but the following lessons led us to the second workflow.

Visualization consideration is still important in the agentic workflow. While visualization seems to be less helpful for an automated agent (if not using vision language models), it is valuable for human users to review the generated report. Decoupling visualization and automated data analysis makes it challenging to present and contextualize the findings visually, thus making it difficult for human users to validate the results.

Structured states are needed to formalize the agent's actions. The increase in the number of iterations do not necessarily lead to deeper insights. The agent may repeat similar questions or propose undoable tasks. A structured description of the potential exploration space may better guide the agent and improve the efficiency and effectiveness of the analysis.

3 Results

We evaluated our system on the Vispub dataset [1], which contains metadata on IEEE VIS publications. The topic specified for analysis was *"What happened to research on automated visualization?"* [3]. The system successfully identified the temporal trends of automated visualization research, highlighted key changes over these years, and generated a coherent visualization report.

To assess the system's generalizability and robustness, the agent was applied to three additional Kaggle datasets, i.e., publications on data science, the Auto MPG dataset, and the Pokémon dataset. The results are available at `https://vis4sense.github.io/agentic-vis-2025/`.

4 Reflections

Task-oriented vs. exploratory analysis. While our first workflow (Fig. 1a) attempts exploratory analysis, we moved to the top-down workflow (Fig. 1b) that drives the analysis through planned tasks. The latter performs better in terms of the coherency and coverage of the report. Nevertheless, we found the exploratory workflow could perform more in-depth analysis and produce unexpected and non-trivial results. We believe the agent will be more powerful if the two types of strategies are integrated, e.g., by realizing the proposed loops in the second workflow.

Generation vs. function calling. We decided to create the visualization and analysis script mainly through code generation and only included a simple tool for network visualization. The advantage is that it allows the system to be light-weight, flexible, and generalizable, and also sometimes produces creative results. However, the quality of the generated code can vary with task complexity. A hybrid approach that combines intelligent decision-making with robust fallback mechanisms may offer a good balance of reliability and flexibility.

Validation. The stochastic nature of LLMs and the automated workflow, where the agent carries out a series of analysis actions and decisions, bring a major challenge in validating the analysis process and results. We believe more dedicated visual presentation of the insights and analytic provenance would facilitate human validation as well as inform further conversation to enhance human agency in a highly-automated process.

References

[1] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. vispubdata.org: A metadata

[3] Interactive report (finalised submission on Vispub dataset)

collection about IEEE visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2199–2206, Sept. 2017. doi: 10.1109/TVCG.2016.2615308 2
[2] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. *Visual analytics: Definition, process, and challenges*. Springer, 2008. 1