InsightLens: Augmenting LLM-Powered Data Analysis with Interactive Insight Management and Navigation

Luoxuan Weng, Xingbo Wang, Junyu Lu, Yingchaojie Feng, Yihan Liu, Haozhe Feng, Danqing Huang, and Wei Chen

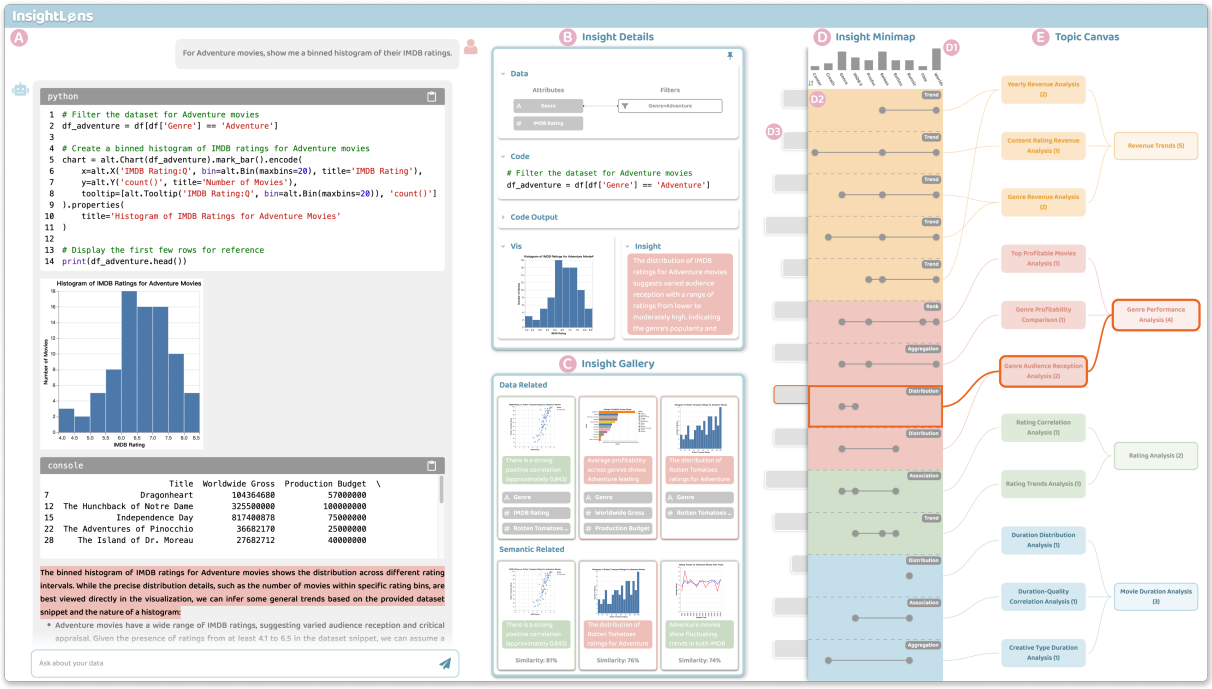

Fig. 1: The user interface of *InsightLens*. The *Chat Window* (A) enables conversational interactions between users and LLMs. The *Insight Details* (B) displays the currently focused insight's summary with its relevant data context and supporting evidence. The *Insight Gallery* (C) presents the corresponding related insights in terms of data and semantics. The *Insight Minimap* (D) visualizes the analysis process chronologically based on each insight. The *Topic Canvas* (E) provides the hierarchical topic structure of all insights.

Abstract— The proliferation of large language models (LLMs) has revolutionized the capabilities of natural language interfaces (NLIs) for data analysis. LLMs can perform multi-step and complex reasoning to generate data insights based on users' analytic intents. However, these insights often entangle with an abundance of contexts in analytic conversations such as code, visualizations, and natural language explanations. This hinders efficient recording, organization, and navigation of insights within the current chat-based LLM interfaces. In this paper, we first conduct a formative study with eight data analysts to understand their general workflow and pain points of insight management during LLM-powered data analysis. Accordingly, we introduce *InsightLens*, an interactive system to overcome such challenges. Built upon an LLM-agent-based framework that automates insight recording and organization along with the analysis process, *InsightLens* visualizes the complex conversational contexts from multiple aspects to facilitate insight navigation. A user study with twelve data analysts demonstrates the effectiveness of *InsightLens*, showing that it significantly reduces users' manual and cognitive effort without disrupting their conversational data analysis workflow, leading to a more efficient analysis experience.

Index Terms—Large language model, interactive data analysis, natural language interface, conversational contexts

- *Luoxuan Weng, Junyu Lu, Yingchaojie Feng, Yihan Liu, and Wei Chen are with the State Key Lab of CAD&CG, Zhejiang University. E-mail: {lukeweng, junyulu, fycj, liuyihan1024, chenvis}@zju.edu.cn.*
- *Xingbo Wang is with Weill Cornell Medical College, Cornell University. E-mail: xingbo.wang@med.cornell.edu.*
- *Haozhe Feng and Danqing Huang are with Tencent Inc. E-mail: {aidenhzfeng, daisyqhuang}@tencent.com.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

1 Introduction

Natural language interfaces (NLIs) for data analysis [12, 51] have received much attention in recent years. Users express their analytic intents and data-related questions in natural language (NL), prompting NLIs to generate corresponding results or visualizations for further analysis. Recently, large language models (LLMs), such as GPT-4 [2] and LLaMA [69], have achieved unprecedented performance in NL understanding, reasoning, and generation. They have become the backbones for NLIs (*e.g.*, ChatGPT's Advanced Data Analysis [53]) to enhance conversational data analysis [20, 79], hereafter referred to as *LLM-powered data analysis*.

During LLM-powered data analysis, LLMs can perform multi-step and complex reasoning to derive data insights based on users' queries about the dataset and the previous conversational contexts [59]. This process also generates various intermediate outputs, such as code, vi-

sualizations, and NL explanations [11]. In a typical round of question and answer (Q&A) within analytic conversations, users must carefully examine and understand the insights generated by LLMs, which are usually entangled with an abundance of intermediate outputs. Furthermore, data analysis is an exploratory and iterative procedure that commonly involves multiple rounds of Q&A. As such, maintaining awareness and keeping track of the entire analyses is essential for making informed decisions and determining future exploration directions [61, 73]. This emphasizes the need to *record*, *organize*, and *navigate* the insights generated throughout the analysis process.

However, recording, organizing, and navigating insights within the current chat-based LLM interfaces is tedious and inefficient, especially given the intertwined data and semantic context involved. During data analysis, insights need to be recorded with their supporting evidence (*i.e.*, intermediate outputs like visualizations) for sharing and reporting purposes [7]. This requires users to navigate back and forth in the conversation to locate the needed information. As analytic conversations are usually lengthy and overwhelmed with various contexts, this process often causes significant manual effort. Existing tools primarily focus on tracking the provenance of a single form of context (*e.g.*, data [15], code [33], or visualization [44]) and are not tailored for conversational interfaces. This limitation impedes efficient insight understanding and recording that involves multiple forms of context in the conversation. The situation is exacerbated for insight organization and navigation. Given the increasing volume of LLM-generated insights and the quickly expanding conversation length, users face a substantial cognitive load. They struggle to manage and organize these insights efficiently in a structured and readable manner, while also maintaining convenient navigation. Although numerous systems have emerged to help users organize and explore LLMs' responses in various scenarios [31, 64, 65], they often fall short in addressing the challenges in data analysis conversations. Some studies focus only on the semantic context (*e.g.*, topic changes [36]) and ignore the data context [26, 61]. Others focus on monitoring and verifying single rounds of Q&A [32, 78], which is insufficient to comprehensively explore the entire analysis process containing multi-round Q&A.

Therefore, our goal is to make conversational data analysis more trackable and navigable for users, and to support on-the-fly recording and organization of insights through a new interaction paradigm. Informed by a formative interview study with eight experts of LLM-powered data analysis, we summarize the challenges of existing chat-based LLM interfaces for data analysis. Accordingly, we present *InsightLens* , an interactive system to facilitate insight recording, organization, and navigation. Rather than burdening users with manually managing insights from the complex conversational contexts, *InsightLens* adopts an LLM-agent-based framework for automatic recording and organization of insights during conversational data analysis. Moreover, *InsightLens* augments traditional chat-based interfaces with multi-level and multi-faceted visualizations to aid in monitoring and navigating the entire conversation. Specifically, it features an *Insight Minimap* and a *Topic Canvas* that progressively evolve along with the analysis process to reveal the temporal shifts of data and semantic context. They provide on-the-fly feedback to guide data exploration without disrupting the conversational workflow. To evaluate the effectiveness of *InsightLens*, we conducted a technical evaluation and a user study. The technical evaluation demonstrated a satisfactory performance of the agent-based framework in accurately recording and organizing insights. The user study revealed that the system can significantly reduce users' manual and cognitive effort for insight management and navigation in LLM-powered data analysis, leading to an improved analysis experience.

In summary, the major contributions of our work are:

- A formative study that identifies critical challenges and summarizes design requirements for insight management and navigation during LLM-powered data analysis.
- *InsightLens*, a system that facilitates insight recording, organization, and navigation through a novel LLM-agent-based framework and interactive visualizations.
- A technical evaluation and a user study that demonstrate the effectiveness of *InsightLens*.

2 Related Work

2.1 NLIs for Data Analysis

Natural language is an intuitive modality for data interaction, significantly lowering the barriers of data analysis [26]. Therefore, NLIs for data analysis have been extensively studied in multiple fields including databases [3], NLP [38], and visualization [19]. Chen *et al.* [7] divided these systems into two types: NLIs for data queries and for visualizations. Following this categorization, we review previous works and discuss recent advancements in LLM-powered data analysis.

NLIs for data queries convert NL utterances into machine-readable formats like SQL and Python to execute on knowledge bases [9]. Early systems relied on pattern-matching [83], parsing strategies [57], or rule-based methods [14] to understand the semantic structures of queries [3]. Later, neural approaches [22,70] trained end-to-end networks to directly generate executable SQL queries from NL inputs, addressing issues like ambiguities or fuzzy linguistic coverage. Recently, training-free strategies using LLMs have emerged, achieving state-of-the-art performance [84] by leveraging LLMs' reasoning abilities with minimal in-context examples, as demonstrated by systems like Binder [9].

NLIs for visualizations (V-NLIs) [60, 62] take a step further by generating visualizations based on query results. Introduced by Cox *et al.* [12], these systems enable users to focus more on their data rather than manipulating complex visual interfaces. Many efforts aim to resolve ambiguities or underspecifications in input queries [19, 58]. For example, NL4DV [51] explicitly highlighted ambiguities in its generated visualization specifications. Other research explores analytic context to maintain a conversational flow [23, 68]. Evizeon [26] applied pragmatics principles and defined context transition types (*i.e.*, *continue*, *retain*, and *shift*). Based on this, Snowy [61] recommended context-aware utterances for conversational visual analysis. Similarly, our work also highlights data context transitions during analysis.

Recently, **analytical assistants powered by LLMs** have become a prevalent paradigm [46, 78]. Many commercial business intelligence (BI) platforms like Power BI [47] and Tableau [67] have integrated LLM support for chat-based insight discovery and dashboard generation. Empirical studies have explored conversational challenges [11] and user behaviors [20] during LLM-powered data analysis. Automated LLM-based tools have also been developed, such as InsightPilot [43] for simplifying data exploration by generating insights and AI Threads [24] for creating and refining charts through a multi-threaded chatbot.

Overall, the extensive studies on NLIs for data analysis provides a solid foundation for our work. We focus on LLM-powered data analysis for its recent prevalence and rather immature interaction schemes [21]. While conversations are natural and intuitive, this new paradigm brings unique challenges that increase manual and cognitive load on users [32]. For example, recent studies have explored cognitive issues like tedious code verification [78] and overwhelming response comprehension [11]. In this work, we aim to identify the pain points in conversational data analysis and help users better manage insights along the way.

2.2 Analytic Provenance in Data Analysis

Analytic provenance tracks the history and evolution of various analytic context, such as data [54] and visualizations [44], which helps users better understand the analysis process. Ragan *et al.* [55] introduced an organizational framework to characterize different types and purposes of provenance, which Madanagopal *et al.* [44] further expanded by mapping tasks to provenance types. Researchers have also proposed various techniques for effective provenance management [52] and presentation [5]. For example, Berant *et al.* [4] used cell-based provenance with NL utterances to explain queries over data tables, while DIY [50] enabled users to evaluate NLIs' correctness on databases by visualizing data subset transformations. XNLI [16] provided interactive widgets to depict visualization provenance in V-NLIs for explanation and diagnosis. More recently, WaitGPT [78] visualized the step-by-step generation of data code to help users monitor and verify LLM-powered data analysis. Our work extends these efforts by extracting and tracking insights along with other analytic context, binding these insights with relevant evidence (*e.g.*, visualizations) to enhance user comprehension.

2.3 Exploration of LLM Responses

Limitations of the linear conversational structures pose challenges in supporting complex information tasks with LLMs [40]. Therefore, numerous visual interfaces have been introduced to facilitate LLM response exploration [27, 41]. For example, Sensecape [65] supported multi-level exploration and sensemaking, while Graphologue [31] created interactive diagrams based on named entity recognition, both enhancing users' understanding of individual responses. Luminate [64] further supported structured examination of multiple responses by generating a multi-dimensional design space for human-AI co-creation. Additionally, C5 [36] and Memory Sandbox [28] addressed conversational context management issues by visualizing topic transitions and enabling transparent memory management, respectively. However, these interfaces are not tailored for data analysis, limiting their effectiveness in managing data insights. Our work extends this research by offering multi-level and multi-faceted visualizations to facilitate insight management and navigation in conversational data analysis.

3 Formative Study

The target users of our system are data analysts who utilize LLMs for analytical tasks. To understand the pain points and challenges of existing chat-based interfaces, we conducted a formative interview study. This study specifically examined how participants record, organize, and navigate insights generated by LLMs during conversational data analysis within a ChatGPT-like interface. Based on our findings, we derived four design requirements to facilitate insight management.

3.1 Participants and Procedure

Participants. Eight data analysts from various domains, including finance and e-commerce were interviewed (E1-8, 3 females and 5 males, age from 25 to 32). Four participants were senior data analysts, while the remaining four were juniors or intermediates. All of them had recently used LLMs for generating data visualizations or insights.

Settings. We created an analytical chatbot based on Open Interpreter [29] with GPT-4, akin to ChatGPT's Advanced Data Analysis. It could be prompted with queries to generate code for data processing and visualization, and then interpret execution results to derive insights.

Procedure. Participants were asked to perform open-ended data analysis [16] with the system to explore the movies dataset from Vega, which contains 709 rows and 10 columns. Similar to their daily work, the task was to derive and record data insights based on the dataset and produce a clear, structured report. We collected their feedback on the analysis experience, focusing on how they acquired information from the conversation and organized and navigated the insights for summarization or further exploration. We then identified challenges and obstacles they encountered. The interviews were conducted online and lasted about 60 to 80 minutes.

3.2 Findings

We observed how participants managed the generated insights throughout the analysis process. For each single round of Q&A, they first reviewed the textual response and visualizations (if any) to grasp the main idea of the message. Some of them then scrolled back to examine the code and its execution results, which were noted as being '*helpful for understanding and reproducibility*' (E5). Subsequently, participants recorded and organized insights through copy-and-paste or screenshots with documentation tools like Google Docs. After collecting enough insights or finishing a specific analytic topic, they navigated previous notes or screenshots to recap findings and plan next steps. However, during the entire process, participants faced several common challenges that decreased analysis efficiency, which are summarized below.

For clarity, we define the terminologies used in the paper.

- **Analytic Context:** Properties of the dataset (focused attributes and values), user interactions (analytic intents and data-related questions), intermediate outputs for analytic purposes (code, code outputs, visualizations, and NL explanations), and insights derived by LLMs.
- **Insight Evidence:** *Parts* of the intermediate outputs generated by LLMs that *directly* support each insight, including the *specific piece* of code, code outputs, visualizations, and NL explanations.

C1: Laborious insight recording from overwhelming conversational contexts. Recording an insight requires both tediously *'summarizing the key idea of the lengthy response'* (E1) and *'locating relevant information like visualizations as supporting evidence'* (E3). For example, E5 spent much time in scrolling back to copy code snippets and their outputs *'in case of reproducing the results in the future'*. The situation was exacerbated when participants had to iteratively modify their utterances to steer LLMs' behavior, in which case the insight and its evidence would span across multiple responses, causing extra effort for excessive scrolling. Although LLMs could be explicitly prompted to generate less verbose responses, balancing between comprehensiveness and succinctness was hard to achieve, especially during data analysis. As stated by E4, *'I prefer comprehensive analyses for high-level questions, but only need a quick answer for simple data queries.'*

C2: Significant overhead for insight organization. Most participants (7/8) organized recorded insights into meaningful subgroups based on data attributes or analytic topics with external documentation tools. This process was described as *'troublesome and painstaking'* (E4), due to the necessity of manually annotating each insight with its characteristics before synthesizing them collectively. Notably, some participants (3/8) explicitly asked LLMs to help organize insights. However, obtaining satisfactory results required iterative and nuanced prompt engineering, which could disrupt the analysis flow. As stated by E8, *'I had to start another conversation specially used for organization, otherwise the original analysis conversation would become too messy.'* The frequent switching between different conversation threads and documentation tools was *'frustrating and time-consuming'* (E3). Meanwhile, as the analysis progressed, the document itself became overwhelmed with *'many unordered texts and images'* (E5), which made it even harder for structured organization.

C3: Inflexible and inefficient insight browsing and revisiting. Participants constantly revisited and navigated previous findings throughout the analysis process. They reported that the lack of *'a high-level insight overview'* (E7) hindered quick navigation and contextual understanding, especially when the conversation became lengthy. The extra cognitive load for insight navigation mainly reflected in two aspects. First, browsing insights was inconvenient. For example, E3 maintained an outline of her discoveries in Word, but the document soon became lengthy, forcing her to *'repeatedly scroll up and down to browse each section'*, which *'somewhat outweighed the advantages of organizing insights'* (E3). Moreover, participants desired to prioritize significant insights during navigation instead of *'random meandering'* (E4), which was not supported. Second, revisiting previous related insights and their supporting evidence was cumbersome, which is a frequent need during analysis for *'comparison or reference'* (E6) and *'inspiring new discoveries'* (E8), as stated by many participants (5/8). Besides, many participants (5/8) mentioned that they sometimes unknowingly stuck in certain subsets of data attributes (E2, E5) or analytic topics (E1), leading to potential biases. Such issues could have been mitigated if users were *'more aware of the data or semantic changes'* (E1).

3.3 Design Requirements

The findings indicate that data analysts struggle with current LLM interfaces for insight management and navigation. To this end, we aim to design a novel interactive system for better recording, organization, and navigation of insights to facilitate a more efficient data analysis experience. The design requirements can be summarized as follows.

R1: Support automatic insight recording from LLMs' responses. Manual recording of insights from the overwhelming conversation requires users' tedious examination and excessive scrolling (**C1**). Therefore, the system should constantly monitor the conversation to automatically summarize and record the generated insights and bind relevant insight evidence (*e.g.*, code outputs, visualizations) with them, regardless of whether LLMs' responses are verbose or not.

R2: Facilitate effective and on-the-fly insight organization. Manual organization of insights based on data attributes or analytic topics is inefficient and troublesome (**C2**), especially when numerous insights and messy analytic context are involved. Meanwhile, the context switching between different applications or conversation threads incurs

extra cognitive load. Hence, the system should organize insights in a non-intrusive manner along with the analysis process.

R3: Provide multi-level and multi-faceted insight navigation. Browsing and revisiting previous insights from multiple aspects or levels of detail are burdensome (**C3**). Therefore, the system should support multi-faceted insight navigation (*e.g.*, temporal, data attributes, analytic topics). Additionally, insight interestingness [13] and context transitions [61] should be highlighted to help users quickly identify significant insights and enhance analytic comprehensiveness. To facilitate easier navigation of the entire conversation, an insight-level overview should be provided, with details on demand to inspect each insight with its supporting evidence and other related insights.

R4: Adopt familiar and unobtrusive interactions and visual designs for seamless data analysis. Users generally appreciate the conversational manner for its intuitiveness and user-friendliness. Therefore, augmenting existing conversational interfaces with seamlessly integrated visualizations is more favorable than creating complex new tools. To avoid steep learning curves and high switching costs, the system should adopt familiar visual designs and non-intrusive interactions without disrupting the original chat-based workflow.

4 InsightLens: Framework

Informed by the summarized challenges and design requirements, we propose automating the recording and organization of insights during analysis, and displaying these insights with on-the-fly visualizations to facilitate user navigation. To achieve this, we develop an LLM-agent-based framework (Figure 2B) that comprises two components: *Insight Extraction (IE)* and *Insight Organization (IO)*, each powered by an LLM-based agent [79]. The *IE Agent* takes each round of Q&A within the conversation as input, extracts insights from LLMs' raw responses, and associates them with relevant evidence (**R1**). It then evaluates the extracted insights' interestingness based on their semantic and statistical significance (**R3**). These insights are subsequently passed to the *IO Agent*, which examines their data and semantic characteristics and dynamically organizes them along with all previous insights (**R2, R3**). The framework automatically runs in the background throughout the analysis process without disrupting the conversational workflow (**R4**). In this section, we describe the prompt engineering techniques of our framework. Following best practices of designing LLM-based agents, we adopt the ReAct [80] paradigm for prompting and equip the agents with specialized tools and in-context memory, allowing them to plan and execute actionable steps to perform various tasks. We use OpenAI's `gpt-4-0125-preview` model for implementation.

4.1 Insight Extraction

To support automatic insight extraction (**R1**), the *IE Agent* keeps monitoring the conversation as the analysis progresses (Figure 2B1). Upon the user completing one round of Q&A with the analytical chatbot, the *IE Agent* is responsible for examining the messages and outputting a JSON-formatted insight list. Therefore, the core of the agent design lies in its step-by-step prompt engineering, which is detailed below.

Providing background knowledge. Prior to task delineation, we introduce the definitions of some key terminologies in data analysis such as *insight*, *insight evidence*, and *insight interestingness* (Figure 2B1(a)), drawing from previous literature [13, 75] and our formative study. This allows the agent to be familiar with the essential domain knowledge, facilitating task performance and output quality. Subsequently, we provide a brief description of the dataset currently in play, including its title and attributes. This ensures the agent's focus of the conversation is confined to the content relevant to the data and analytic context, instead of extracting unrelated insights. Finally, we underscore the task and the required output format with a few demonstration examples to better leverage LLMs' in-context learning [9] abilities for desired results.

Identifying/Refining insights. For each round of Q&A, we instruct the agent to carefully examine and determine whether it contains insights and output an insight list (Figure 2B1(c)). Meanwhile, we maintain the *previously* extracted insights as the agent's memory (Figure 2B1(b)), which not only helps it leverage in-context learning to extract and output insights in a consistent manner, but also enables the refinement of previous insights. During conversational data analysis, users may not always pose a new question every time; instead, they often iteratively adjust their prompts for clarification or enhancement [11]. For example, a user may request an alternative visualization to better illustrate a particular insight. Therefore, by directing the agent to choose between two actions (*i.e.*, `'identify new insight'` or `'refine existing insight'`), we ensure a comprehensive analysis of each round of Q&A without missing any follow-up information. Moreover, rather than replicating LLMs' verbose responses, the extracted insights are always *summarized* into concise sentences for intuitive understanding. This eliminates users' burden of extra prompt engineering to retrieve quick answers for simple data queries, while systematically extracting all insights for high-level questions.

Associating insight evidence. To automatically bind all relevant insight evidence with each insight (Figure 2B1(d)), the agent is required to scrutinize the code, code outputs, visualizations, and NL explanations in LLM responses, focusing on their data and semantic implications. This allows the agent to locate the *minimum* but *critical* parts that directly support each insight, which mitigates users' manual and cognitive load in understanding and recording insights without having to examine the entire contexts. We provide in-context examples for each type of insight evidence to improve the agent's awareness and performance of the task. Meanwhile, we instruct the agent to also take previous insights into consideration for potential modifications, in case that new evidence may emerge due to users' iterative prompting.

Evaluating insight interestingness. Inspired by QuickInsights [13], we judge insight interestingness (**R3**) by two factors: its *semantic significance* (*i.e.*, the subject of it should be important, such as a best-selling product) and *statistical significance* (*i.e.*, the relevant statistical metrics of it should be notable, such as a high standard deviation).

1. The agent evaluates each insight's semantic meaning and assigns a *semantic score* S_{sem} of 1 to 5 based on its overall understanding of the insight under the analytic context. For instance, if the user focuses on product profit, the 1st most profitable product is more significant than the 3rd one. We instruct the agent to consider multiple aspects (*i.e.*, significance, impact, relevance) [82] for a comprehensive assessment, and provide in-context examples (*i.e.*, insight-score pairs) and previous scores to enhance scoring performance and consistency.
2. The agent categorizes the insights and uses function calls to calculate their corresponding statistical metrics (Figure 2B1(e)). We follow prior works for categorizing insights [75] and mapping insight categories to suitable statistical metrics [61]. As insights may belong to multiple categories, we employ a *majority-vote* strategy [66] to determine the most prominent one. Then, the agent assigns a *statistical score* S_{stat} of 1 to 5 based on the calculated metrics and heuristics adopted from [60, 75]. For example, a high Pearson correlation coefficient results in a high S_{stat} for correlation insights.

We combine two scores using a weighted average: $S_{final} = S_{sem} \cdot \omega + S_{stat} \cdot (1-\omega)$, with the weight ω empirically set to 0.6. Finally, S_{final} is rounded to a scale of 1 to 5 (Figure 2B1(f)), with a rationale generated by the agent. Higher scores indicate higher insight interestingness.

4.2 Insight Organization

To organize insights from multiple aspects on the fly (**R2, R3**), the *IO Agent* receives the extracted insights (with relevant evidence) and examines their data and semantic characteristics (Figure 2B2). It is responsible for determining the corresponding data context and analytic topics/subtopics of each insight. Based on the *IO Agent*'s outputs, we sequentially categorize the insights into different subgroups. We introduce our prompt techniques and topic classification method below.

Providing overall analysis domain. To ensure the identification of valid data attributes and relevant analytic topics, we provide an automatically generated NL description of the dataset, its first five rows, and a list of its attributes beforehand (Figure 2B2(a)). This enables the agent to gain an overall understanding of the current analysis domain.

Determining data context. The agent is tasked with identifying the corresponding data attributes associated with each insight. To mitigate the risk of fabricating non-existent attributes, we explicitly instruct the

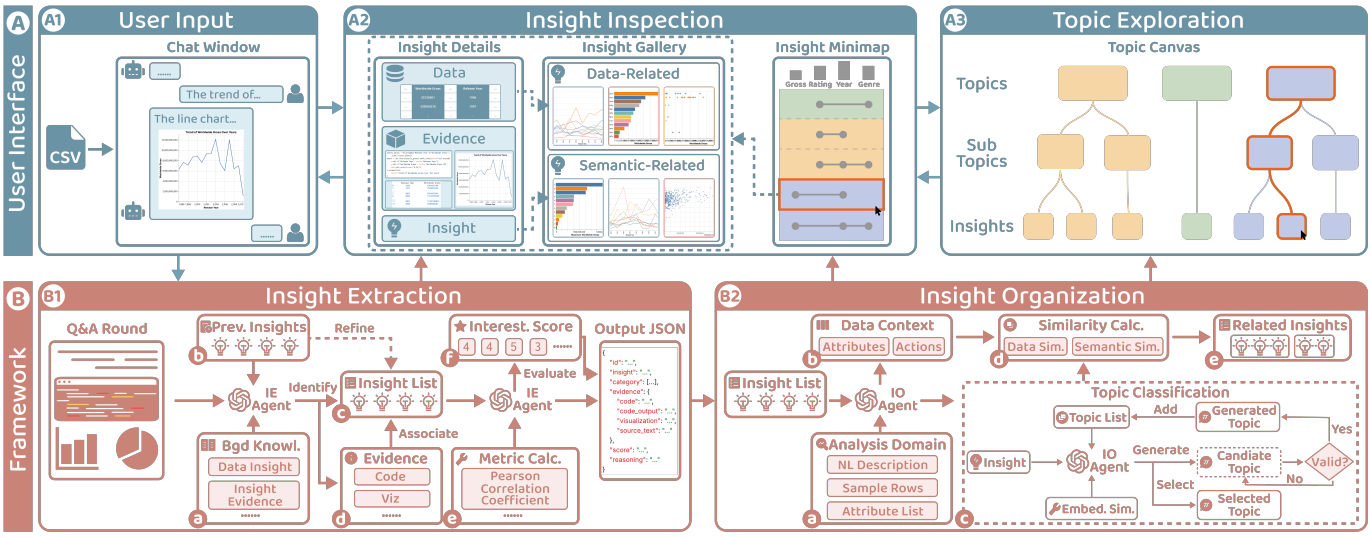

Fig. 2: *InsightLens* consists of (A) a user interface and (B) an LLM-agent-based framework. While users are (A1) interacting with the analytical chatbot, the *Insight Extraction (IE) Agent* (B1) takes each round of Q&A for insight extraction and evidence association, as well as interestingness evaluation. Following this, the *Insight Organization (IO) Agent* (B2) organizes the insights by identifying their data context, analytic topics, and related insights. Users can then (A2) inspect the extracted insights and (A3) explore the structured topics with progressively-evolving visualizations.

agent to restrict its selection to the given attribute list. Meanwhile, it is required to identify the analytical actions (*e.g.*, *filtering* and *aggregation*, if any) applied to the data subset pertinent to each insight, based on the insight evidence provided. Consequently, we can obtain each insight's data context (Figure 2B2(b)) to support users' detailed inspection needs.

Classifying into topics/subtopics. Traditional topic modeling methods (*e.g.*, LDA) are limited when handling short and sparse texts [71] like insights. Also, they generate latent topics (*i.e.*, collections of words) that lack clear semantic meanings. Inspired by a recent work [36], we adopt LLMs to sequentially assign *human-readable* topics (*e.g.*, a topic named *Climate Analysis*) for each insight (Figure 2B2(c)).

1. First, we maintain a list of current topics that are generated based on previous insights as the agent's memory.
2. For each new insight, the agent is instructed to `select` a suitable topic from the list that best describes its semantic meaning. To combine LLMs' NL understanding abilities with a best practice from prior literature [56], we provide cosine similarities between the embeddings of the insight and each existing topic for reference, enabling the agent to make more informed decisions.
3. In cases where no existing topic semantically describes the insight, or when the topic list is initially empty, the agent must `generate` an appropriate analytic topic by abstracting the insight into a concise and high-level title. We prompt the agent to ensure that the new topic falls within the provided analysis domain and is broad enough to encompass similar subsequent insights. To avoid generating identical or overlapping topics, the agent must utilize function calls to calculate the cosine similarities between the candidate new topic and each existing topic. We empirically set the similarity threshold at 0.55. If any similarity score exceeds this threshold, the agent must generate another new candidate topic. Once the new topic is determined, it is added to the topic list for future selection.
4. Finally, the selected or generated analytic topic for the newly extracted insight is determined. We then recursively execute the above steps to classify subtopics within the assigned main topic.

Notably, we use the `all-MiniLM-L6-v1` model from Sentence Transformers [56] for embedding calculation.

Identifying related insights. After obtaining the data context and analytic topics of the extracted insights, we categorize them into subgroups to enable user navigation from different aspects. We also determine related insights across two dimensions (Figure 2B2(d)). First, we identify *data-related* insights by comparing the intersections between their associated data attributes. For example, an insight associated with `[MPG, Year, Origin]` is closely related to another one associated with `[MPG, Year]`. Second, we identify *semantic-related* insights by comparing the cosine similarities between their embeddings. Consequently, two lists of related insights are derived for each insight (Figure 2B2(e)). By linking them together, we address the common user need for easier reference or comparison of similar data findings.

5 InsightLens: User Interface

Built upon the LLM-agent-based framework, *InsightLens* features a user interface (Figure 2A) to facilitate insight management and navigation during LLM-powered data analysis. In this section, we first present an overview of the user interface, and then describe its core features, visual designs, and interactions, including *User Input* (Figure 2A1), *Insight Inspection* (Figure 2A2), and *Topic Exploration* (Figure 2A3).

5.1 User Interface Overview

The user interface of *InsightLens* features five coordinated views (Figure 1). It is designed to augment existing interfaces while maintaining users' original conversational workflow (**R4**). Given the unique nature of conversations which display the most information at first glance, we sought advice from data analysts in our formative study and iteratively refined our visual designs. Consequently, we choose to adopt a *'details first, overview last'* strategy [42] from left to right to make the user interface more applicable to the conversational workflow, while facilitating easy inspection and navigation of insights during analysis.

To achieve this, we keep the *Chat Window* (Figure 1A) similar to ChatGPT on the left, where users can input their analytic intents and view LLMs' responses. Beside it, the *Insight Details* (Figure 1B) shows an individual insight with its relevant data context and supporting evidence for thorough inspection, while the *Insight Gallery* (Figure 1C) displays its data- and semantic-related insights for convenient comparison. Additionally, we employ a matrix-based design in the *Insight Minimap* (Figure 1D) to chronologically visualize the analysis process. Each row represents a unique insight, showcasing its data and semantic characteristics. Finally, the *Topic Canvas* (Figure 1E) on the right adopts a tree-based design to visualize the hierarchical topic structure, enabling users to explore their findings across different analytic topics.

5.2 User Input & Insight Inspection

As the entry point of the user interface, users upload their datasets and interact with the analytical chatbot in the *Chat Window*. Right beside it lays the *Insight Details* and *Insight Gallery* arranged vertically to enable detailed inspection for each insight. Along with the conversation flow, we provide an overview of all extracted insights in the *Insight Minimap*, which is constructed by *insight rows* vertically stacked in temporal order.

These four views are coordinated to *scroll together* seamlessly. By clicking on each insight row, users can conveniently examine its details and navigate between different parts of the conversation. Collectively, these progressively-evolving visualizations support the following tasks to facilitate multi-level and multi-faceted insight navigation without disrupting the conversational workflow (**R3, R4**).

Inspecting insight details. As the conversation progresses, the *Insight Details* updates with the latest extracted insight. It consists of five sections (*i.e.*, *Data*, *Code*, *Code Output*, *Vis*, and *Insight*) to display the insight's summary along with its associated data context and evidence. These sections are *collapsible* to enable details on demand and to satisfy different user background and preferences (*e.g.*, some analysts might be unfamiliar with coding and prefer to view only the data context or visualizations). By default, the *Code* and *Code Output* sections are collapsed to benefit non-technical users. Meanwhile, the relevant NL explanations are *highlighted* in LLMs' original responses in the *Chat Window*. All these content are the *minimum* but *critical* parts of the intermediate outputs to reduce users' cognitive load. To navigate among different insights, users can either 1) scroll in the *Chat Window* or *Insight Minimap* or 2) click on the dots ( ) below each response. Pinning ( ) is also supported to temporarily disable scrolling coordination to focus on a specific insight.

Comparing related insights. In accordance with the currently focused insight of the *Insight Details*, we present its related insights in the *Insight Gallery*, ranked by similarity (or by temporal order for ties). For simplicity, only the associated visualization and the insight's summary are displayed in each *insight card*. To enable a clear understanding of the rationales behind each recommendation, we show the relevant data attributes for data-related insights and similarity scores for semantic-related insights. Users can click on each insight card in the gallery to view its details for comparison or reference.

Revealing data coverage. On top of the minimap, we provide a histogram (Figure 1D1) to visualize the distribution of the associated insight counts across each data attribute. By observing the histogram, users can intuitively understand which attributes have already been extensively analyzed and which ones remain underexplored. Hovering and sorting are also supported to view detailed information and quickly locate the uncovered attributes. Therefore, users' awareness of their data coverage during analysis can significantly be improved.

Understanding context transitions. In each insight row of the minimap (Figure 1D2), we represent its associated data attributes with a set of connected points (corresponding to the above histogram). The horizontal connecting lines can visually indicate the holistic consideration of the involved attributes in each round of Q&A. We also provide vertical reference lines activated by hovering over any point to maintain alignment with the histogram. These insight rows not only enable a quick review of each insight's data context, but also showcase context transitions throughout the analysis process, which reveal the change of users' focused attributes. For example, certain visual patterns can represent different types of transitions like *continue* ( ), *retain* ( ), and *shift* ( ) [61]. Grasping these transitions helps users track the progress of analytic conversations [26], thereby mitigating the risk of analytical biases, such as focusing excessively on specific data subsets. In case that users expect to prioritize some attributes of interest, *e.g.*, always monitoring '`Worwide Gross`' for financial analysis, they can *drag* the bars in the above histogram to adjust column order. Additionally, we colorize each insight row to denote its analytic topic and reveal the topic changes. Overall, this simple and intuitive design can be seamlessly integrated into the conversational workflow and helps users better review their analyses across both data and semantic dimensions.

Highlighting insight interestingness. To empower users to easily identify and revisit high-quality or interesting insights, we visualize the interestingness scores of each insight as horizontal bars (Figure 1D3), as well as adding a category tag in each insight row for reference. As the 'interestingness' of an insight can be subjective and varies among users [60], the scores automatically assigned by LLMs may not accurately reflect user preferences (*i.e.*, whether they would find the insight significant). To balance this, we provide LLMs' explanations for the rationales behind each interestingness score on hovering, and also allow users to dynamically *adjust* the score by resizing the corresponding bar. Therefore, this feature offers an alternative way for users to manage and navigate previous insights, either based on automated evaluations or their own judgment, similar to a 'bookmark' for insight significance.

5.3 Topic Exploration

As the highest-level overview, the *Topic Canvas* visualizes the hierarchical topic structure of all extracted insights. We choose the tree-based design due to its simplicity and intuitiveness for topic organization and exploration (**R3, R4**). The tree (without a root node) is structured into two levels, representing main topics and their subtopics, respectively. Each node indicates a topic/subtopic, differentiated by color and labeled with its title and associated insight count. These nodes are visually linked to their corresponding insight rows in the *Insight Minimap*. Additionally, hovering over any node will highlight its included insights (and subtopics, if any) and display a brief description for quick inspection of each topic's essence. Overall, the *Topic Canvas* is automatically updated along with the analysis process and coordinated with other views to facilitate insight navigation across analytic topics.

6 Technical Evaluation

The effectiveness of *InsightLens* depends on whether our framework can successfully record and organize the LLM-generated insights. Therefore, we conducted a technical evaluation focusing on (1) the *coverage* of insight extraction, (2) the *accuracy* of insight evidence association, and (3) the *quality* and *accuracy* of insight organization.

6.1 Experiment Settings

Dataset. We collected 10 datasets from reputable sources (6 from Kaggle and 4 from Vega) with diverse analysis domains (*e.g.*, education, economics) and number of rows ($\mu = 1058, \sigma = 777$) and columns ($\mu = 14, \sigma = 5$). We manually crafted 10 analytic queries for each dataset, totaling to 100 samples. These queries, together with their corresponding datasets, were input into our system, resulting in 104 extracted insights and 50 generated analytic topics (with 70 subtopics).

Methodology. To evaluate insight extraction, we first manually identified and labeled the key insights in the original responses generated by the analytical chatbot, providing a ground truth for the insights extracted by the *IE Agent*. Then, we measured the ratio of covered labeled insights to their total number (*i.e.*, coverage). As the automatically extracted insights were summarized by the *IE Agent* for easier understanding, we considered a labeled insight as covered if its semantic meaning was *contained* in the corresponding extracted insight.

To evaluate evidence association, we measured the ratio of insights with correctly associated evidence to the total number of extracted insights (*i.e.*, accuracy). If any part of the evidence (*i.e.*, code, code outputs, visualizations, and NL explanations) was incorrect or irrelevant to its corresponding insight, we considered it as a negative sample. To ensure robustness, we adopted a four-step verification process: (1) confirming the exact match between the associated evidence and the original response; (2) evaluating the correctness of data processing through manual code review and execution results; (3) validating the appropriateness of visualizations based on the ground truth processed data; and (4) examining the relevance of the associated evidence to the corresponding insight through manual assessment.

To evaluate insight organization, we focused on two aspects: data and semantic characteristics (see Section 4.2). For data context, we measured the ratio of insights with correctly identified data attributes (and analytical actions, if any) to the total number of extracted insights (*i.e.*, accuracy). For analytic topics/subtopics, we utilized GPT-4 to rate their quality, a widely adopted method in the NLP community for assessing machine-generated texts that has proven effective in various scenarios [10, 18, 39]. Specifically, we instructed GPT-4 to consider multiple aspects of the topics (*e.g.*, relevance, clarity, adaptability) for a thorough evaluation. The detailed prompts can be found in the supplemental material. As the assignment of analytic topics is subjective and lacks a definitive ground truth, we compared the rating scores of our dynamically generated topics with a static baseline [36] (*i.e.*, feeding all insights to GPT-4 for topic generation). We then manually labeled

each insight with the topic list generated by our system as a ground truth for evaluating topic classification accuracy.

6.2 Results

Metrics. For insight extraction, the coverage of the extracted insights was **91.2%** (*i.e.*, covered 176 out of 193 labeled insights). For evidence association, the accuracy of the associated insight evidence was **88.5%** (*i.e.*, 92 corrects and 12 errors). For insight organization, the accuracy of the identified data context was **88.5%** (*i.e.*, 92 corrects and 12 errors). Additionally, analytic topics generated by our system received an average quality rating of **7.6** on a 10-point scale, surpassing the static baseline (5.9). The accuracy of topic classification was **91.3%** (*i.e.*, 95 corrects and 9 errors). Overall, these statistical metrics demonstrated the effectiveness and robustness of our LLM-agent-based framework.

Failure Cases Analysis. For insight extraction, we categorized the 17 failure cases into two types: (1) *Missing Insights* (8/17) and (2) *Missing Details* (9/17). The *IE Agent* sometimes failed to extract all the key insights; instead, it tended to only focus on the most significant ones. For instance, with the query *'compute the average discount percentage offered by each smartphone brand'*, only the brands with the highest and lowest discounts were highlighted, while the analytical chatbot actually mentioned numerous intermediate brands in its response. In other cases, the agent over-summarized the information, omitting critical details. An example of this is an extracted insight that merely acknowledged the *'top 10 most profitable movies'* without specifying their titles.

For evidence association, we observed two failure modes: (1) *No Code/Code Output* (5/12) and (2) *Incorrect NL Explanations* (7/12). In the former, the *IE Agent* did not include any associated code or code output in its responses. In the latter, it provided incorrect NL explanations that did not align with the insights, arising from either fabricated sentences or an oversimplification of the original output.

For insight organization, we evaluated failures in terms of data context accuracy and topic classification accuracy. Data context errors primarily stemmed from *Fabricating Attributes* (9/12), with the remainder due to *Missing Attributes* (3/12). The former occurred when the analytical chatbot created new attributes for specific queries (*e.g.*, defining a `Decade` attribute from `Year`), leading to the *IO Agent*'s inability to correctly identify the original dataset attributes. In contrast, the latter was due to the agent's occasional failure to fully deduce the associated attributes. Regarding topic classification, the predominant issue was *Topic Disagreement* (9/9), where humans and GPT-4 focused on different aspects. Since insights could span multiple topics, such cases were technically not 'errors' but rather outcomes of varying labeling criteria.

Overall, most failure cases discussed above can be ascribed to LLMs' hallucinations. Such issues are particularly evident given the intricate nature of our targeted tasks and the complex prompting techniques we employ for our framework, which often lead to LLMs' generation of unexpected outputs. To mitigate this, we can incorporate more effective instructions to make LLMs' behavior more reliable and robust [81].

Summary. Despite the few failure cases, the results demonstrated our framework's high coverage, accuracy, and quality in automated insight recording and organization. This can significantly reduce users' manual and cognitive effort during conversational data analysis, establishing a solid foundation for the interactive features of *InsightLens*.

7 User Study

To evaluate the effectiveness of *InsightLens* in facilitating insight management and navigation during LLM-powered data analysis, we conducted a within-subjects user study. Specifically, we aimed to collect users' feedback on the effectiveness and usability of *InsightLens*'s features, as well as its impact on the overall data analysis process.

7.1 Experiment Design

Participants and Setup. We recruited 12 data analysts (P1-12, 4 females and 8 males, age from 24 to 29) from the business intelligence department of a local technology company. Their expertise levels in data analysis ranged from junior/medium (8/12, < 5-year experience) to senior (4/12, > 5-year experience). Their daily tasks included analyzing datasets and reporting data findings, with proficiency in various tools like Excel (12/12), Python (10/12), and Microsoft Power BI (8/12). All of them had experience using LLMs (*e.g.*, ChatGPT, Claude, Qwen) for their work with varying frequencies (6 often, 4 sometimes, 2 rarely). Each participant received $25 as compensation upon completion.

InsightLens's visual support for insight management and navigation primarily relies on the four coordinated views (*i.e.*, *Insight Details*, *Insight Gallery*, *Insight Minimap*, and *Topic Canvas*) to function as a whole. Therefore, we set the comparative *Baseline* as the *Chat Window* of *InsightLens* excluding all interactive features to evaluate their effects, similar to prior studies on LLM data analysis interfaces [78]. This ChatGPT-like *Baseline* mirrored the systems familiar to participants for LLM-powered data analysis and maintained the same appearance and chat functionality as *InsightLens* for a fair comparison. We also provided a document editor for participants to record their findings.

Tasks and Datasets. Participants were asked to use both *InsightLens* and *Baseline* to analyze two datasets: (1) a housing dataset (15 columns, 1460 rows) and (2) a colleges dataset (14 columns, 1214 rows). They were instructed to perform open-ended data exploration on each dataset to provide insights into (1) the housing market dynamics for real estate agents, and (2) the various factors of US colleges for student applicants, as if they were to provide a comprehensive data report within a week. To mitigate learning effects while ensuring comparability of collected data across different experiment sessions, we split each dataset into two parts [35], each of which was allocated to one of the systems.

Procedure. Initially, participants were asked to sign a consent form and fill out a pre-study questionnaire to collect their demographic information. After that, we conducted a tutorial using an example dataset to introduce the features of both systems. Participants were then given adequate time to familiarize themselves with each system, during which they were encouraged to raise any questions or concerns.

Then, participants were requested to use both systems across two datasets (and tasks). We counterbalanced the order of the systems and datasets (4=2x2 sessions in total) to mitigate learning effects. Each session lasted 15 minutes and was screen- and audio-recorded as system logs. Participants were also encouraged to think aloud about their thoughts and findings during the analysis process.

Finally, participants were required to complete a post-study questionnaire using a 5-point Likert scale, followed by a semi-structured interview to comprehend their ratings and collect qualitative feedback on the effectiveness, usability, and potential impact of the system on their daily workflow. The entire study lasted about 120 minutes.

Measures. We collected 48(=12x4) recordings and system logs. To complement participants' qualitative feedback, we employed the following measures: (1) *number of recorded insights*, (2) *number of unique data attributes explored*, and (3) *number of unique analytic topics explored*. These measures were informed by previous literature [15, 49] and offered quantitative evidence for our analysis. To ensure methodological consistency, we utilized the same prompting techniques of *InsightLens* on *Baseline* for data processing.

7.2 Results

All participants completed four experiment sessions successfully. Based on their qualitative feedback and the collected quantitative measures, we discuss the effectiveness of *InsightLens* in facilitating insight management and navigation (Figure 3). We then report *InsightLens*'s feature effectiveness, system usability, and impact on data analysis (Figure 4).

Support for Insight Management. The effectiveness of *InsightLens* in facilitating insight management was appreciated by all participants ($\mu = 4.67 > 2.67, p = .002$). Recording insights was much easier in *InsightLens*, whereas *Baseline* forced participants to manually scrutinize and summarize LLMs' lengthy responses. P3 expressed his favor for *'the dots below each message'* that *'reminded him of missed out insights'*. We also observed that participants constantly referred to the *Insight Details* to review and record the relevant insight evidence, which allowed them to *'easily see the involved attributes and charts without scrolling up and down'* (P10). For organizing insights, the progressively updating *Topic Canvas* and *Insight Minimap* significantly eased participants' burden, mitigating the need for *'resorting to tools like Word'* (P5) and *'summarizing an insight outline'* (P6).

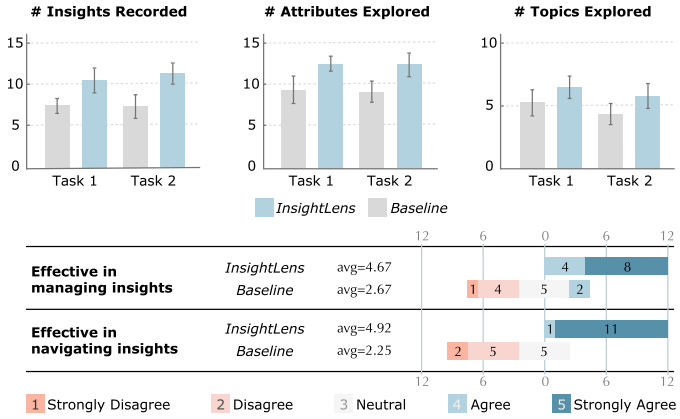

Fig. 3: The results of the measures and qualitative ratings regarding *InsightLens*'s support for insight management and navigation.

No.	Question	Average	Score Distribution
Q1	The *Insight Details* helps me inspect the insight and its relevant data context and evidence.	4.50	6 6
Q2	The *Insight Gallery* helps me browse and revisit previous relevant insights.	4.17	4 2 6
Q3	The *Insight Minimap* helps me review and navigate between different insights during analysis.	4.75	1 1 10
Q4	The *Topic Canvas* helps me organize and summarize different insights during analysis.	4.00	2 8 2
Q5	It is easy to learn the system.	4.67	1 2 9
Q6	It is easy to use the system.	4.50	6 6
Q7	I will use the system again.	4.67	4 8
Q8	The system does not disrupt my original workflow of conversational data analysis.	4.67	1 2 9
Q9	The system reduces my manual and cognitive effort during LLM-powered data analysis.	4.75	3 9
Q10	The system improves my understanding of the analyses generated by LLMs.	4.33	1 6 5

1 Strongly Disagree 2 Disagree 3 Neutral 4 Agree 5 Strongly Agree

Fig. 4: The results of the questionnaire regarding *InsightLens*'s effectiveness, usability, and impact on data analysis.

Additionally, one of our measures reinforced *InsightLens*'s support for insight recording. Specifically, participants recorded more insights using *InsightLens* compared to *Baseline* (Task 1: $\mu = 10.4 > 7.4, p = .002$; Task 2: $\mu = 11.1 > 7.3, p = .005$). We ascribed the observed significant difference to *InsightLens*'s support for reducing the time needed for locating insights and their relevant evidence, thereby leading to more insights recorded within a limited time frame.

Support for Insight Navigation. *InsightLens* was rated as highly effective in reviewing and navigating previous insights ($\mu = 4.92 > 2.25, p = .002$). Participants highly valued *InsightLens*'s features for navigating insights from different aspects. For example, P4 appreciated *'tracking her findings by time order in the minimap'*, while *'using the baseline required her to scroll back and forth to grasp what she explored before'*. During open-ended data exploration, participants recognized the importance of maintaining awareness of the overall analysis flow, which avoided *'repetitive analyses on previously explored topics'* (P8).

Interestingly, the quantitative measures revealed the potential expansion on participants' data and analytic coverage due to their improved awareness of the analyses. When using *InsightLens*, they explored more data attributes (Task 1: $\mu = 12.4 > 9.3, p = .006$; Task 2: $\mu = 12.3 > 9.1, p = .012$) and analytic topics (Task 1: $\mu = 6.5 > 5.3, p = .03$; Task 2: $\mu = 5.8 > 4.3, p = .035$) than *Baseline*. During the experiments, we constantly noticed that participants checked and navigated in the *Insight Minimap* or *Topic Canvas* before posing their next query. Consequently, these observed significant differences implied participants' tendency to analyze more comprehensively when provided with easier navigation of the recorded insights organized across data and semantic dimensions.

Feature Effectiveness. Overall, the features of *InsightLens* were well-received by most participants. Firstly, the *Insight Details* (Q1) was appreciated by participants for allowing them to *'quickly obtain an insight summary without manually reading every piece of messages'* (P5, P7). Also, the associated insight evidence such as code snippets eliminated their need to *'scroll back to find that specific line of code for data transformation'* (P6) to comprehensively record the insight. Secondly, the *Insight Gallery* (Q2) helped participants review related insights conveniently. P8 found it particularly useful for *'understanding attribute relationships when dealing with multiple similar insights'*, while P3 likened it to *'a menu tool'* that enabled him to review different visualization types for similar insights. However, some participants found it less beneficial (P2, P4) due to the rather short analysis time allotted for the experiments. Thirdly, the *Insight Minimap* (Q3) was constantly praised by most participants (8/12) as *'the most useful feature'* (P1). P9 described it as *'being very innovative and reminded him of the minimap in VS Code'*, while others favored its *'clear presentation of covered data attributes'* (P2, P4, P7, P12) and *'color encodings to reveal topic changes'* (P5). This made the analysis process *'more structured and thorough'* (P11). Additionally, the interestingness bars enabled participants to discard trivial insights. For example, P4 identified an insight with an extremely low interestingness score about a negligible attribute relationship *'caused by an accidental query'*. Finally, the *Topic Canvas* (Q4) reduced participants' manual and cognitive effort to organize insights. The generated topics were reported as *'being reasonable and intuitive'* that *'decreased the chaos of the overwhelming conversation'* (P10). Moreover, viewing the tree-based topic structure gave P3 a sense of *'solving the open-ended task from various angles'* - aiding comprehensive thinking - though some preferred relying on personal judgment rather than *'being disturbed by the organized topics'* (P5).

System Usability. All participants found *InsightLens* easy to learn (Q5) and use (Q6), and were willing to integrate it into their daily workflow (Q7). The visual designs were praised as *'very intuitive and user-friendly'* (P3, P7) without *'causing steep learning curves'* (P1). P9 noted that the views looked so natural that *'any professionals could understand its main features at first glance'*. Meanwhile, participants also suggested improvements for *InsightLens*. For example, P4 wished to intervene the organization process by *'proposing her own topics'*, and P11 expected to *'combine certain insights for more in-depth analysis'*.

Impact on Data Analysis. We examined *InsightLens*'s impact on LLM-powered data analysis workflows for fluidity, workload, and understanding. Firstly, participants agreed that *InsightLens* was unobtrusive and did not disrupt their conversational interactions (Q8). P9 noted, *'he just chatted with LLMs as usual, and the views updated automatically without interference'*, while P7 described it as *'essentially a chat interface augmented with useful plugins'*. Secondly, *InsightLens* reduced manual and cognitive load (Q9), alleviating issues like *'excessive scrolling'* (P2) and *'memorizing insights in mind'* (P12). Recording and organizing insights on the fly helped participants *'focus more on the analysis itself rather than constant context switching'* (P10). Finally, *InsightLens* improved participants' understanding of LLM-generated analyses (Q10). P6 remarked, *'it felt like she was more involved in the analysis process by inspecting progressive changes in views, instead of merely inputting queries and waiting for LLMs to handle everything'*. Thus, *InsightLens* helped strike a balance between automation and human agency, thereby increasing users' understanding and engagement.

7.3 Observed Behaviors

We observed two prominent workflow patterns adopted by different participants when using *InsightLens* for data analysis.

User-Initiated Workflow. Participants with a clear analysis goal often posed sequential queries based on their own judgment and preferences with minimal system intervention. For example, P5 explored the colleges dataset focusing on how college ownership influenced factors like student quality and financial condition. Here, the *Insight Minimap* and *Topic Canvas* primarily served as structured and organized ways for reviewing previous insights rather than inspiring new discoveries. The construction of the topic tree mainly progressed from *bottom* (insights) to *top* (topics) with more subtopics than main topics, revealing a depth-oriented exploration pattern.

System-Initiated Workflow. Participants without a specific aim, often due to unfamiliarity with the analysis domain, initially posed

multiple random queries to *'make a draft'* (P1). They then inspected the *Insight Minimap* and *Topic Canvas* to gain an overview of their analyses and observe potential biases (*e.g.*, certain attributes/topics may have been thoroughly explored while others remain overlooked) to plan future explorations. Therefore, the construction of the topic tree was now from *top* to *bottom* with many topics scattered around and few subtopics, reflecting a breadth-oriented exploration pattern.

8 Discussion

8.1 Design Implications

Integrate data and semantic context for enhanced understanding. Given the limitations of linear chat-based interfaces, managing LLMs' contexts for complex tasks has gained popularity in VIS and HCI [31, 36,65]. Unlike existing works that primarily extract semantic structures, *InsightLens* further integrates data context - crucial for data analysis - including data attributes and analytical actions. We visualize dynamic data and semantic context simultaneously in a minimap, allowing users to quickly grasp the analysis process. Our user study shows that this integration not only aids in reviewing and navigating insights but also potentially expands data analysts' data and analytic coverage, leading to more comprehensive results in exploratory data analysis.

Provide follow-up analytic guidance for data exploration. In our user study, many participants (6/12) suggested incorporating query recommendations, particularly for unfamiliar datasets (*i.e.*, the *'cold start'* issue). Prior research has extensively explored analytic guidance [61, 73], which can be improved with LLMs' capabilities [20]. *InsightLens*'s support for organizing insights on the fly can establish a robust foundation for context-aware assistance. For example, integrating another agent into our framework can generate tailored suggestions by considering analysts' background, goals, and current focused topics and attributes, thereby deepening or broadening their analyses.

Balance between flexibility and complexity of interaction paradigms. Our design principle maintains a conversational workflow primarily through natural language. However, we recognize the potential of other modalities for NLI-based data analysis (*e.g.*, direct manipulations [62] and sticky cells [76]). For example, some participants in our user study expected to modify the *Topic Canvas* by adding or editing nodes, akin to mind maps. While such features could enhance LLM interaction flexibility [65], they may also introduce complexities [34]. Therefore, we aim to achieve a trade-off between NLIs' intuitiveness and visualizations' expressiveness. Future research could further explore how to balance these aspects in designing LLM interaction paradigms for data analysis.

8.2 More Application Scenarios

Incorporating LLMs into data analysis is an emerging but promising paradigm. With LLMs' growing reasoning capabilities and extended context windows [6], data analysts can potentially conduct longer and more in-depth analyses on intricate datasets. Such envisions necessitate smart strategies to manage complex analytic contexts. While *InsightLens* focuses on augmenting conversational interfaces, its design rationales can be adapted to traditional data analysis workspaces, such as BI platforms [47] and computational notebooks [37]. For example, in BI platforms like Tableau [67], users conduct visual analysis through a dashboard and a sidebar-based analytical assistant. An *Insight Minimap* can serve as an analytic timeline, allowing users to revisit previous visualization states and maintain awareness of the entire analysis process. Additionally, within Jupyter Notebooks, users typically interact with LLMs via code comments or magic commands [46] to perform exploratory data analysis. Constructing a *Topic Canvas* that reflects the semantics of code and markdown text can enable a non-linear, tree-based navigation of notebook cells, offering an innovative way to organize messy notebooks. Therefore, we believe that our work could inspire future research in making LLM-powered data analysis more streamlined, accessible, and productive through visualizations.

8.3 Limitations and Future Work

Hallucination. LLMs can generate incorrect or misleading insights [21]. While our main focus is on insight management and navigation, *InsightLens* can inherently support some degree of verification by displaying relevant evidence like code snippets or outputs, allowing users to identify potential errors. For example, during our user study, participants frequently reviewed the *Data* section within the *Insight Details* to verify data attributes and analytical actions. This helped them quickly determine if LLMs had correctly utilized and transformed the data, without tediously sifting through lengthy responses. To further improve *InsightLens*'s reliability, we can fine-tune LLMs for data analysis tasks [81] and explore advanced agent designs [25]. Moreover, incorporating methods like code verification [78] and task decomposition [32] can help users proactively diagnose issues in LLM responses. We can also examine data transformation errors and factual contradictions via self-reflection [30] and external knowledge bases [74], and highlight them with visual cues like warning icons [8].

Bias. *InsightLens* relies on LLMs for fully automated insight extraction and topic generation. The lack of interactive control may foster an excessive reliance on LLM outputs, potentially introducing biases [72]. Users might overlook critical insights requiring human interpretation or feel constrained by LLM-generated topics. Moreover, biases inherent in LLMs' training corpus can be exacerbated by the patterns of the current dataset, which may further propagate into the extracted insights or generated topics [17]. To mitigate these issues, we can integrate visual alert mechanisms (*e.g.*, highlighting uncertainties in LLM responses [48]) to improve user awareness of potential biases. Additionally, incorporating user feedback loops, such as enabling adjustments or merging of topic nodes [65], can allow users to provide background knowledge or specify personal preferences. This can enhance the customization of insight extraction and topic generation.

Scalability. To handle complex queries and large numbers of insights/topics, *InsightLens* employs a streaming strategy [79] to enable real-time parsing and rendering of LLM responses, thereby maintaining system responsiveness. This allows for immediate presentation of processed insights/topics and dynamic updating of visualizations to significantly reduce user wait times. Nevertheless, we recognize that it requires additional engineering efforts to support vast numbers of attributes or topics in large-scale data analysis scenarios. Currently, the system's response time is primarily affected by the LLM inference latency, which can be mitigated through hardware acceleration solutions such as Groq [1]. To address potential visual clutter in the user interface, hierarchical semantic zoom [64] for nodes within the *Topic Canvas* and adaptive visual filtering [77] for columns in the *Insight Minimap* can enhance user interaction and prevent interface overload.

Cognitive Impact. *InsightLens* utilizes color encoding to differentiate analytic topics, which may introduce cognitive trade-offs like color fatigue [63] or visual clutter, especially with extensive use or among users with color vision deficiencies. Future work should explore alternative salience strategies to complement color encoding and evaluate how different visual design choices might affect *InsightLens*'s effectiveness. For instance, incorporating additional visual encodings (*e.g.*, size, shape) or multimodal cues [45] can diversify highlighting methods and improve system accessibility.

Design and Evaluation. The participant groups in our formative and user studies were rather small and lacked diversity in age, gender, and domain, potentially introducing biases and limiting representativeness. A larger, more diverse participant pool would enhance the robustness of our design and evaluation. Additionally, conducting a between-subject study and an ablation study on different user interface components would mitigate individual effects and provide a thorough assessment.

9 Conclusion

This work presents *InsightLens*, an interactive system that visualizes the complex conversational contexts during LLM-powered data analysis to facilitate insight management and navigation. Built on an LLM-agent-based framework that automates the recording and organization of insights in analytic conversations, *InsightLens* provides a set of progressively-evolving visualizations to enable multi-level and multi-faceted insight navigation. A technical evaluation and a user study demonstrate the effectiveness of our framework and system.

References

[1] D. Abts, G. Kimmell, A. Ling, J. Kim, M. Boyd, A. Bitar, S. Parmar, I. Ahmed, R. DiCecco, D. Han, J. Thompson, M. Bye, J. Hwang, J. Fowers, P. Lillian, A. Murthy, E. Mehtabuddin, C. Tekur, T. Sohmers, K. Kang, S. Maresh, and J. Ross. A software-defined tensor streaming multiprocessor for large-scale machine learning. In *Proc. ISCA*, p. 567–580. ACM, 2022. doi: 10.1145/3470496.3527405 9

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv*, 2023. doi: 10.48550/ARXIV.2303.08774 1

[3] K. Affolter, K. Stockinger, and A. Bernstein. A comparative survey of recent natural language interfaces for databases. *VLDB J.*, 28:793–819, 2019. doi: 10.1007/S00778-019-00567-8 2

[4] J. Berant, D. Deutch, A. Globerson, T. Milo, and T. Wolfson. Explaining queries over web tables to non-experts. In *ICDE*, pp. 1570–1573, 2019. doi: 10.1109/ICDE.2019.00144 2

[5] M. Chakhchoukh, N. Boukhelifa, and A. Bezerianos. Understanding how in-visualization provenance can support trade-off analysis. *IEEE Trans. Vis. Comput. Graph.*, 29(9):3758–3774, 2023. doi: 10.1109/TVCG.2022.3171074 2

[6] S. Chen, S. Wong, L. Chen, and Y. Tian. Extending context window of large language models via positional interpolation. *arXiv*, 2023. doi: 10.48550/ARXIV.2306.15595 9

[7] Z. Chen and H. Xia. Crossdata: Leveraging text-data connections for authoring data documents. In *Proc. CHI*. ACM, 2022. doi: 10.1145/3491102.3517485 2

[8] F. Cheng, V. Zouhar, S. Arora, M. Sachan, H. Strobelt, and M. El-Assady. Relic: Investigating large language model responses using self-consistency. In *Proc. CHI*. ACM, 2024. doi: 10.1145/3613904.3641904 9

[9] Z. Cheng, T. Xie, P. Shi, C. Li, R. Nadkarni, Y. Hu, C. Xiong, D. Radev, M. Ostendorf, L. Zettlemoyer, N. A. Smith, and T. Yu. Binding language models in symbolic languages. In *ICLR*, 2023. doi: 10.48550/ARXIV.2210.02875 2, 4

[10] C.-H. Chiang and H.-y. Lee. Can large language models be an alternative to human evaluations? In *Proc. ACL*, pp. 15607–15631. ACL, July 2023. doi: 10.18653/v1/2023.acl-long.870 6

[11] B. Chopra, A. Singha, A. Fariha, S. Gulwani, C. Parnin, A. Tiwari, and A. Z. Henley. Conversational challenges in ai-powered data science: Obstacles, needs, and design opportunities. *arXiv*, 2023. doi: 10.48550/ARXIV.2310.16164 2, 4

[12] K. Cox, R. E. Grinter, S. L. Hibino, L. J. Jagadeesan, and D. Mantilla. A multi-modal natural language interface to an information visualization environment. *International Journal of Speech Technology*, 4:297–314, 2001. doi: 10.1023/A%3A1011368926479 1, 2

[13] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In *Proc. SIGMOD*, p. 317–332. ACM, 2019. doi: 10.1145/3299869.3314037 4

[14] M. Dubey, S. Dasgupta, A. Sharma, K. Höffner, and J. Lehmann. Asknow: A framework for natural language query formalization in sparql. In *Proc. International Conference on The Semantic Web*, p. 300–316. Springer, 2016. doi: 10.1007/978-3-319-34129-3_19 2

[15] W. Epperson, V. Gorantla, D. Moritz, and A. Perer. Dead or alive: Continuous data profiling for interactive data science. *IEEE Trans. Vis. Comput. Graph.*, 30(1):197–207, 2024. doi: 10.1109/TVCG.2023.3327367 2, 7

[16] Y. Feng, X. Wang, B. Pan, K. K. Wong, Y. Ren, S. Liu, Z. Yan, Y. Ma, H. Qu, and W. Chen. Xnli: Explaining and diagnosing nli-based visual data analysis. *IEEE Trans. Vis. Comput. Graph.*, pp. 1–14, 2023. doi: 10.1109/TVCG.2023.3240003 2, 3

[17] I. O. Gallegos, R. A. Rossi, J. Barrow, M. M. Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, and N. K. Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179, 09 2024. doi: 10.1162/coli_a_00524 9

[18] M. Gao, X. Hu, J. Ruan, X. Pu, and X. Wan. Llm-based nlg evaluation: Current status and challenges. *arXiv*, 2024. doi: 10.48550/ARXIV.2402.01383 6

[19] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proc. UIST*, p. 489–500, 2015. doi: 10.1145/2807442.2807478 2

[20] K. Gu, M. Grunde-McLaughlin, A. M. McNutt, J. Heer, and T. Althoff. How do data analysts respond to ai assistance? a wizard-of-oz study. *arXiv*, 2023. doi: 10.48550/ARXIV.2309.10108 1, 2, 9

[21] K. Gu, R. Shang, T. Althoff, C. Wang, and S. M. Drucker. How do analysts understand and verify ai-assisted data analyses? *arXiv*, 2023. doi: 10.48550/ARXIV.2309.10947 2, 9

[22] J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J.-G. Lou, T. Liu, and D. Zhang. Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proc. ACL*, pp. 4524–4535. ACL, July 2019. doi: 10.18653/v1/P19-1444 2

[23] M. Hearst and M. Tory. Would you like a chart with that? incorporating visualizations into conversational interfaces. In *IEEE VIS*, pp. 1–5, 2019. doi: 10.1109/VISUAL.2019.8933766 2

[24] M.-H. Hong and A. Crisan. Conversational ai threads for visualizing multidimensional datasets. *arXiv*, 2023. doi: 10.48550/ARXIV.2311.05590 2

[25] 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. *arXiv*, 2024. doi: 10.48550/ARXIV.2402.18679 9

[26] E. Hoque, V. Setlur, M. Tory, and I. Dykeman. Applying pragmatics principles for interaction with visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 24(1):309–318, 2018. doi: 10.1109/TVCG.2017.2744684 2, 6

[27] M. N. Hoque, T. Mashiat, B. Ghai, C. Shelton, F. Chevalier, K. Kraus, and N. Elmqvist. The hallmark effect: Supporting provenance and transparent use of large language models in writing with interactive visualization. *arXiv*, 2024. doi: 10.48550/ARXIV.2311.13057 3

[28] Z. Huang, S. Gutierrez, H. Kamana, and S. Macneil. Memory sandbox: Transparent and interactive memory management for conversational agents. In *Proc. UIST*. ACM, 2023. doi: 10.1145/3586182.3615796 3

[29] O. Interpreter. Open interpreter. `https://github.com/OpenInterpreter/open-interpreter`, 2024. 3

[30] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii, and P. Fung. Towards mitigating LLM hallucination via self reflection. In *Findings of EMNLP*, pp. 1827–1843. Association for Computational Linguistics, Dec. 2023. doi: 10.18653/v1/2023.findings-emnlp.123 9

[31] P. Jiang, J. Rayan, S. P. Dow, and H. Xia. Graphologue: Exploring large language model responses with interactive diagrams. In *Proc. UIST*. ACM, 2023. doi: 10.1145/3586183.3606737 2, 3, 9

[32] M. Kazemitabaar, J. Williams, I. Drosos, T. Grossman, A. Z. Henley, C. Negreanu, and A. Sarkar. Improving steering and verification in ai-assisted data analysis with interactive task decomposition. In *Proc. UIST*. ACM, 2024. doi: 10.1145/3654777.3676345 2, 9

[33] M. B. Kery, B. E. John, P. O'Flaherty, A. Horvath, and B. A. Myers. Towards effective foraging by data scientists to find past analysis choices. In *Proc. CHI*, p. 1–13. ACM, 2019. doi: 10.1145/3290605.3300322 2

[34] S. Lallé, D. Toker, C. Conati, and G. Carenini. Prediction of users' learning curves for adaptation while using an information visualization. In *Proc. IUI*, p. 357–368. ACM, 2015. doi: 10.1145/2678025.2701376 9

[35] Q. Li, H. Lin, C. F. Tang, X. Wei, Z. Peng, X. Ma, and T. Chen. Exploring the "double-edged sword" effect of auto-insight recommendation in exploratory data analysis. In *Proc. IUI Workshop*, CEUR Workshop Proceedings, 2021. 7

[36] P. Liang, D. Ye, Z. Zhu, Y. Wang, W. Xia, R. Liang, and G. Sun. C5: Towards better conversation comprehension and contextual continuity for chatgpt. *arXiv*, 2023. doi: 10.48550/ARXIV.2308.05567 2, 3, 5, 6, 9

[37] Y. Lin, H. Li, L. Yang, A. Wu, and H. Qu. Inksight: Leveraging sketch interaction for documenting chart findings in computational notebooks. *IEEE Trans. Vis. Comput. Graph.*, 30(1):944–954, 2024. doi: 10.1109/TVCG.2023.3327170 9

[38] S.-C. Liu, S. Wang, T. Chang, W. Lin, C.-W. Hsiung, Y.-C. Hsieh, Y.-P. Cheng, S.-H. Luo, and J. Zhang. JarviX: A LLM no code platform for tabular data analysis and optimization. In *Proc. EMNLP*, pp. 622–630. ACL, Dec. 2023. doi: 10.18653/v1/2023.emnlp-industry.59 2

[39] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proc. EMNLP*, pp. 2511–2522. ACL, Dec. 2023. doi: 10.18653/v1/2023.emnlp-main.153 6

[40] Y. Liu, Z. Wen, L. Weng, O. Woodman, Y. Yang, and W. Chen. Sprout: an interactive authoring tool for generating programming tutorials with the visualization of large language models. *IEEE Trans. Vis. Comput. Graph.*, pp. 1–15, 2024. doi: 10.1109/TVCG.2024.3410523 3

[41] J. Lu, B. Pan, J. Chen, Y. Feng, J. Hu, Y. Peng, and W. Chen. Agentlens: Visual analysis for agent behaviors in llm-based autonomous systems. *arXiv*, 2024. doi: 10.48550/ARXIV.2402.08995 3

[42] T. Luciani, A. Burks, C. Sugiyama, J. Komperda, and G. E. Marai. Details-first, show context, overview last: Supporting exploration of viscous

fingers in large-scale ensemble simulations. *IEEE Trans. Vis. Comput. Graph.*, 25(1):1225–1235, 2019. doi: 10.1109/TVCG.2018.2864849 5

[43] P. Ma, R. Ding, S. Wang, S. Han, and D. Zhang. InsightPilot: An LLM-empowered automated data exploration system. In *Proc. EMNLP*, pp. 346–352. ACL, Dec. 2023. doi: 10.18653/v1/2023.emnlp-demo.31 2

[44] K. Madanagopal, E. D. Ragan, and P. Benjamin. Analytic provenance in practice: The role of provenance in real-world visualization and data analysis environments. *IEEE Computer Graphics and Applications*, 39(6):30–45, 2019. doi: 10.1109/MCG.2019.2933419 2

[45] M. R. Mahmud, A. Cordova, and J. Quarles. Multimodal feedback methods for advancing the accessibility of immersive virtual reality for people with balance impairments due to multiple sclerosis. *IEEE Trans. Vis. Comput. Graph.*, 30(11):7193–7202, 2024. doi: 10.1109/TVCG.2024.3456189 9

[46] A. M. Mcnutt, C. Wang, R. A. Deline, and S. M. Drucker. On the design of ai-powered code assistants for notebooks. In *Proc. CHI*. ACM, 2023. doi: 10.1145/3544548.3580940 2, 9

[47] Microsoft. Power bi. https://www.microsoft.com/en-us/power-platform/products/power-bi, 2024. 2, 9

[48] D. Munechika, Z. J. Wang, J. Reidy, J. Rubin, K. Gade, K. Kenthapadi, and D. H. Chau. Visual auditor: Interactive visualization for detection and summarization of model biases. In *IEEE VIS*, pp. 45–49, 2022. doi: 10.1109/VIS54862.2022.00018 9

[49] A. Narechania, A. Coscia, E. Wall, and A. Endert. Lumos: Increasing awareness of analytic behavior during visual data analysis. *IEEE Trans. Vis. Comput. Graph.*, 28(1):1009–1018, 2022. doi: 10.1109/TVCG.2021.3114827 7

[50] A. Narechania, A. Fourney, B. Lee, and G. Ramos. Diy: Assessing the correctness of natural language to sql systems. In *Proc. IUI*, p. 597–607. ACM, 2021. doi: 10.1145/3397481.3450667 2

[51] A. Narechania, A. Srinivasan, and J. Stasko. Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Trans. Vis. Comput. Graph.*, 27(2):369–379, 2021. doi: 10.1109/TVCG.2020.3030378 1, 2

[52] P. H. Nguyen, K. Xu, A. Wheat, B. W. Wong, S. Attfield, and B. Fields. Sensepath: Understanding the sensemaking process through analytic provenance. *IEEE Trans. Vis. Comput. Graph.*, 22(1):41–50, 2016. doi: 10.1109/TVCG.2015.2467611 2

[53] OpenAI. Chatgpt plugins. https://openai.com/blog/chatgpt-plugins#code-interpreter, 2024. 1

[54] X. Pu, S. Kross, J. M. Hofman, and D. G. Goldstein. Datamations: Animated explanations of data analysis pipelines. In *Proc. CHI*. ACM, 2021. doi: 10.1145/3411764.3445063 2

[55] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Trans. Vis. Comput. Graph.*, 22(1):31–40, 2016. doi: 10.1109/TVCG.2015.2467551 2

[56] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. EMNLP-IJCNLP*, pp. 3982–3992. ACL, Nov. 2019. doi: 10.18653/v1/D19-1410 5

[57] D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan. Athena: an ontology-driven system for natural language querying over relational data stores. *Proc. VLDB Endow.*, 9(12):1209–1220, aug 2016. doi: 10.14778/2994509.2994536 2

[58] V. Setlur, M. Tory, and A. Djalali. Inferencing underspecified natural language utterances in visual analysis. In *Proc. IUI*, p. 40–51. ACM, 2019. doi: 10.1145/3301275.3302270 2

[59] L. Shen, H. Li, Y. Wang, and H. Qu. From data to story: Towards automatic animated data video creation with llm-based multi-agent systems. *arXiv*, 2024. doi: 10.48550/ARXIV.2408.03876 1

[60] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Trans. Vis. Comput. Graph.*, 25(1):672–681, 2019. doi: 10.1109/TVCG.2018.2865145 2, 4, 6

[61] A. Srinivasan and V. Setlur. Snowy: Recommending utterances for conversational visual analysis. In *Proc. UIST*, p. 864–880. ACM, 2021. doi: 10.1145/3472749.3474792 2, 4, 6, 9

[62] A. Srinivasan and J. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE Trans. Vis. Comput. Graph.*, 24(1):511–521, 2018. doi: 10.1109/TVCG.2017.2745219 2, 9

[63] H. Strobelt, D. Oelke, B. C. Kwon, T. Schreck, and H. Pfister. Guidelines for effective usage of text highlighting techniques. *IEEE Trans. Vis. Comput. Graph.*, 22(1):489–498, 2016. doi: 10.1109/TVCG.2015.2467759 9

[64] S. Suh, M. Chen, B. Min, T. J.-J. Li, and H. Xia. Structured generation and exploration of design space with large language models for human-ai co-creation. *arXiv*, 2023. doi: 10.48550/ARXIV.2310.12953 2, 3, 9

[65] S. Suh, B. Min, S. Palani, and H. Xia. Sensecape: Enabling multilevel exploration and sensemaking with large language models. In *Proc. UIST*. ACM, 2023. doi: 10.1145/3586183.3606756 2, 3, 9

[66] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, and G. Wang. Text classification via large language models. In *Findings of EMNLP*, pp. 8990–9005. Association for Computational Linguistics, Dec. 2023. doi: 10.18653/v1/2023.findings-emnlp.603 4

[67] Tableau. Tableau einstein. https://www.tableau.com/, 2024. 2, 9

[68] M. Tory and V. Setlur. Do what i mean, not what i say! design considerations for supporting intent and context in analytical conversation. In *IEEE VAST*, pp. 93–103, 2019. doi: 10.1109/VAST47406.2019.8986918 2

[69] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv*, 2023. doi: 10.48550/ARXIV.2302.13971 1

[70] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proc. ACL*, pp. 7567–7578. ACL, July 2020. doi: 10.18653/v1/2020.acl-main.677 2

[71] H. Wang, N. Prakash, N. K. Hoang, M. S. Hee, U. Naseem, and R. K.-W. Lee. Prompting large language models for topic modeling. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 1236–1241, 2023. doi: 10.1109/BigData59044.2023.10386113 5

[72] N. Wang and L. Chen. User bias in beyond-accuracy measurement of recommendation algorithms. In *Proc. RecSys*, p. 133–142. ACM, 2021. doi: 10.1145/3460231.3474244 9

[73] X. Wang, F. Cheng, Y. Wang, K. Xu, J. Long, H. Lu, and H. Qu. Interactive data analysis with next-step natural language query recommendation. *arXiv*, 2022. doi: 10.48550/ARXIV.2201.04868 2, 9

[74] X. Wang, R. Huang, Z. Jin, T. Fang, and H. Qu. Commonsensevis: Visualizing and understanding commonsense reasoning capabilities of natural language models. *IEEE Trans. Vis. Comput. Graph.*, 30(1):273–283, 2024. doi: 10.1109/TVCG.2023.3327153 9

[75] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE Trans. Vis. Comput. Graph.*, 26(1):895–905, 2020. doi: 10.1109/TVCG.2019.2934398 4

[76] Z. J. Wang, K. Dai, and W. K. Edwards. StickyLand: Breaking the Linear Presentation of Computational Notebooks. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. ACM, 2022. doi: 10.1145/3491101.3519653 9

[77] C. Weaver. Cross-filtered views for multidimensional visual analysis. *IEEE Trans. Vis. Comput. Graph.*, 16(2):192–204, 2010. doi: 10.1109/TVCG.2009.94 9

[78] L. Xie, C. Zheng, H. Xia, H. Qu, and C. Zhu-Tian. Waitgpt: Monitoring and steering conversational llm agent in data analysis with on-the-fly code visualization. In *Proc. UIST*. ACM, 2024. doi: 10.1145/3654777.3676374 2, 7, 9

[79] T. Xie, F. Zhou, Z. Cheng, P. Shi, L. Weng, Y. Liu, T. J. Hua, J. Zhao, Q. Liu, C. Liu, L. Z. Liu, Y. Xu, H. Su, D. Shin, C. Xiong, and T. Yu. Openagents: An open platform for language agents in the wild. *arXiv*, 2023. doi: 10.48550/ARXIV.2310.10634 1, 4, 9

[80] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023. doi: 10.48550/ARXIV.2210.03629 4

[81] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv*, 2023. doi: 10.48550/ARXIV.2309.01219 7, 9

[82] 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 Trans. Vis. Comput. Graph.*, pp. 1–17, 2024. doi: 10.1109/TVCG.2024.3368060 4

[83] W. Zheng, H. Cheng, L. Zou, J. X. Yu, and K. Zhao. Natural language question/answering: Let users talk with the knowledge graph. In *Proc. CIKM*, p. 217–226. ACM, 2017. doi: 10.1145/3132847.3132977 2

[84] F. Zhou, M. Hu, H. Dong, Z. Cheng, F. Cheng, S. Han, and D. Zhang. TaCube: Pre-computing data cubes for answering numerical-reasoning questions over tabular data. In *Proc. EMNLP*, pp. 2278–2291. ACL, Dec. 2022. doi: 10.18653/v1/2022.emnlp-main.145 2