InsightLens: Discovering and Exploring Insights from Conversational Contexts in Large-Language-Model-Powered Data Analysis

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

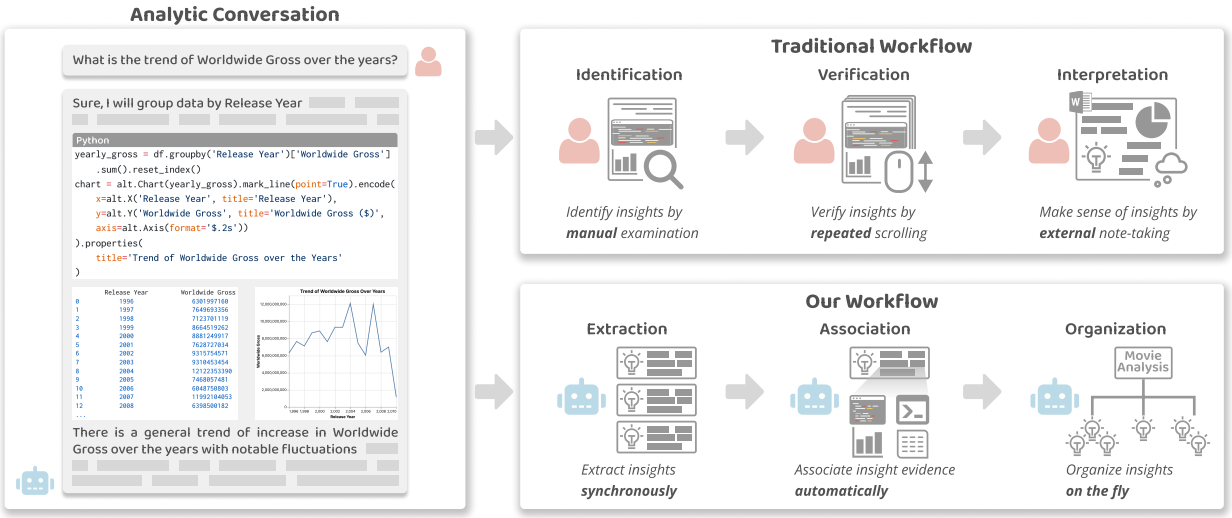

Fig. 1: In a typical workflow of conducting data analysis with large language models, users are required to identify, verify, and interpret insights from lengthy analytic conversations overwhelmed with different contexts. To alleviate the manual and cognitive load during the process, we adopt an LLM-based multi-agent framework that automates the extraction, association, and organization of 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 identification, verification, and interpretation of insights within the current chat-based interfaces of LLMs. In this paper, we first conduct a formative study with eight experienced data analysts to understand their general workflow and pain points during LLM-powered data analysis. Then, we propose an LLM-based multi-agent framework to automatically extract, associate, and organize insights along with the analysis process. Based on this, we introduce *InsightLens*, an interactive system that visualizes the intricate conversational contexts from multiple aspects to facilitate insight discovery and exploration. 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

1 Introduction

Natural language interfaces (NLIs) for data analysis [53, 56] 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 data visualizations for further analysis. Recently, large language models (LLMs), such as GPT-4 [1] and LLaMA [65], have emerged and achieved unprecedented performance in natural language understanding, reasoning, and generation. They have become the backbones for NLIs (*e.g.*, ChatGPT's Advanced Data Analysis [47]) to enhance conversational data analysis [20, 73], 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 conversation history. This process typically generates various intermediate outputs, such as code, visualizations, and NL explanations [11]. After identifying the key insights from LLMs' responses, users often need to associate them with the corresponding intermediate outputs for verification, since LLMs may sometimes provide unreliable or incorrect responses due to hallucinations [77]. As the conversations progress, users may navigate back and forth between different parts of the conversation to gather essential information for understanding the current analyses generated by LLMs. Meanwhile, they need to keep track and make sense of the previously discovered insights for making informed decisions and determining future explorations [59, 68]. Finally, users will record, organize, and report valuable insights by exploring the entire conversation history.

However, this workflow is tedious and inefficient with the current chat-based interfaces of LLMs. As analytic conversations are usually lengthy and overwhelmed with various types of context, it requires significant manual and cognitive effort to frequently navigate in the conversations to extract insights and associate them with the supporting evidence (*i.e.*, intermediate outputs). In contrast, most existing research

- *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.*

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

only tracks the provenance of a single form of analytic context, such as data [15], code [31], or visualization [80], ignoring their combinations, which impedes a comprehensive understanding of the analysis process. Furthermore, users are required to manually maintain the discovered insights either by mental recall or through external note-taking [7]. Given the large numbers of insights quickly generated by LLMs and the expanding conversational contexts, this process often causes a substantial cognitive load. Many interactive systems have been developed to help users explore LLMs' responses in various scenarios such as creative writing [62] and information seeking [29, 63]. However, they primarily focus on semantic context (*e.g.*, topic changes [34]) of LLMs' outputs, and fail to facilitate the exploration of data context during analytic conversations [25, 59]. Moreover, most such systems generally lack integrated support for recording and organizing insights.

To better understand the general workflow, challenges, and design requirements in LLM-powered data analysis, we conducted a formative interview study with eight experienced data analysts. Accordingly, we present *InsightLens*, an interactive system that facilitates efficient insight discovery and exploration. Going beyond traditional analytical chatbots that are limited to interact with a single intelligent agent [54] and require users to manually manage the conversational contexts, *InsightLens* adopts an LLM-based multi-agent framework for automatic extraction, association, and organization of insights during conversational data analysis. Moreover, *InsightLens* offers multi-level and multi-faceted visualizations to aid in exploring the organized insights. Specifically, it features an *Insight Minimap* and a *Topic Canvas* to reveal the temporal shifts of data and semantic context throughout the analysis process. They provide on-the-fly feedback to guide insight discovery and 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 our multi-agent framework in accurately extracting, associating, and organizing insights. The user study revealed that *InsightLens* can significantly reduce the manual and cognitive effort in discovering and exploring insights in LLM-powered data analysis, leading to a more efficient analysis experience.

In summary, the major contributions of our work are:

- A formative study that identifies critical challenges and summarizes design requirements for discovering and exploring insights from conversational contexts in LLM-powered data analysis.
- *InsightLens*, an interactive system that facilitates efficient insight discovery and exploration through a novel multi-agent 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 interacting with data and can significantly lower the barriers of data analysis [25]. Therefore, NLIs for data analysis have been extensively studied in multiple fields including databases [2], natural language processing (NLP) [36], and visualization [56]. Chen *et al.* [8] divided these systems into two categories: NLIs for data queries and for visualizations. We follow this categorization to review previous work and then discuss about the cutting-edge progress in LLM-powered data analysis.

NLIs for data queries, or most well known as *semantic parsing* [30], transform NL utterances into machine-readable representations like SQL and Python to execute on knowledge bases [9]. Early systems leveraged pattern-matching [78], parsing strategies [52], or rule-based methods [14] to understand the semantic structures of the input queries [2]. Later, neural approaches [22, 67] trained end-to-end neural networks to directly generate executable SQL queries from NL inputs, which overcame previous limitations like ambiguities or fuzzy linguistic coverage [28]. Recently, researchers developed training-free strategies utilizing LLMs to address issues of end-to-end neural models like low interpretability and large training data need, and achieved state-of-the-art performance [72, 79]. Binder [9], for instance, used only a few in-context examples to bind LLMs' strong reasoning abilities with programming languages to tackle with complex data queries.

NLIs for visualizations (V-NLIs) [53, 57, 58, 60] take a step further by responding with interactive visualizations based on query results. Initially introduced by Cox *et al.* [12], these systems allow users to focus more on their data rather than manipulating complex visual interfaces [56]. Many work aims at resolving the ambiguities or underspecifications in input queries [18, 53, 55]. For example, NL4DV [45] generated analytic specifications for visualizations and explicitly highlighted ambiguities in its responses. Another important line of research in V-NLIs explores analytic context to maintain a conversational flow [23, 64]. Evizeon [25] applied pragmatics principles for interacting with visualizations and defined three types of context transitions (*i.e.*, *continue*, *retain*, and *shift*). Based on this, Snowy [59] recommended context-aware utterances to support conversational visual analysis. Similarly, our work also highlights data context transitions during analysis.

Recently, **analytical assistants powered by LLMs** have emerged as a prevalent paradigm [35, 42]. Many empirical studies have been conducted to understand the conversational challenges [11] and user behaviors [20, 21] during LLM-powered data analysis. Moreover, automated tools have also been developed to better leverage LLMs' potentials. For instance, InsightPilot [40] simplified data exploration by automatically generating insights, and AI Threads [24] created and refined visualizations through a multi-threaded analytical chatbot.

Overall, the large corpus of previous studies in NLIs for data analysis provides a solid foundation for our work. We choose to focus on LLM-powered data analysis for its recent prevalence and rather immature interaction schemes. This new paradigm brings unique challenges that create high manual and cognitive overload on users. Therefore, we concentrate on investigating the pain points during conversational data analysis and enabling users to better discover and explore data insights.

2.2 Analytic Provenance in Data Analysis

Analytic provenance tracks the history and evolution of different analytic context, such as data [49], visualizations [41], and insights [19], which helps users better understand the analysis process. Ragan *et al.* [50] introduced an organizational framework to characterize different types and purposes of provenance. Based on this framework, Madanagopal *et al.* [41] further investigated the mapping between tasks and provenance types, such as knowledge transfer, validation, and sensemaking. During these processes, researchers have proposed various techniques for effective provenance management [46] and presentation [5]. For example, Berant *et al.* [4] presented a cell-based provenance with NL utterances to explain queries over data tables. And DIY [44] enabled users to evaluate NLIs' correctness on databases by visualizing representative data subset transformations. Similarly, XNLI [16] provided interactive widgets to depict visualization provenance in V-NLIs for explanation and diagnosis. Our work builds upon these endeavors by extracting and tracking the insights and other analytic context during LLM-powered data analysis. Moreover, we associate these insights with their relevant evidence (*e.g.*, code, visualizations) to facilitate user comprehension and verification.

2.3 Exploration of LLMs' Responses

Limitations of the linear conversational structures pose challenges in supporting complex information tasks with LLMs [37]. Therefore, numerous visual interfaces have been introduced to facilitate LLM response exploration [26, 38, 61]. For example, Sensecape [63] provided multi-level exploration and sensemaking for information-seeking activities, while Graphologue [29] created an interactive diagram of LLMs' responses based on named entity recognition. Both of them enhanced users' understanding of individual responses. To support structured examination of multiple responses, Luminate [62] systematically generated a multi-dimensional design space for human-AI co-creation processes. Furthermore, other work focuses on better managing LLMs' conversational contexts. C5 [34], for example, addressed the human and model contextual forgetting issues by dynamically visualizing the topic transitions during conversations. Similarly, Memory Sandbox [27] enabled transparent and interactive context management

of LLM-powered agents. Nevertheless, these interfaces are not tailored for data analysis scenarios, hence they fall short in supporting effective exploration of analytic context. Our work extends this line of research by providing multi-level and multi-faceted visualizations to facilitate insight discovery and exploration during conversational data analysis.

3 Formative Study

The target users of our system are data analysts who utilize LLMs for analytical tasks. To understand the workflow, pain points, and best practices of LLM-powered data analysis, especially how users discover and explore data insights, we conducted a formative interview study to summarize challenges with traditional chat-based interfaces. Based on our findings, we derive four design requirements to facilitate insight discovery and exploration from LLMs' conversational contexts.

3.1 Participants and Procedure

Eight experienced data analysts from various domains, including business intelligence, finance, and e-commerce were interviewed (E1-8, 3 females and 5 males, age from 25 to 32). Each participant had a minimum of 4 years' experience in data analysis, and all of them had recently used LLMs for their work. We developed a prototype system that served as a localhost analytical chatbot powered by GPT-4. Then, participants were asked to perform open-ended data analysis [16] with the system to explore the movies dataset from Vega, which consists of 709 rows and 10 columns. We encouraged participants to use think-aloud protocol to raise any questions or concerns. Finally, we collected their feedback on analysis experience, focusing on how they acquired information from the conversation history and organized the obtained data insights for summarization or further data exploration, as well as their encountered challenges and obstacles during the process. The interviews were conducted online and lasted about 60 to 80 minutes.

3.2 Findings

LLMs were prompted with analytic queries to generate code for data processing and visualization, and then interpret the execution results to provide data insights. We observed three operations that participants commonly performed during LLM-powered data analysis: *identification, verification, and interpretation* of insights. First, they identified the key insights from each response through carefully examining the entire message. Most participants (7/8) temporarily saved the insights through copy-and-paste or screenshots. Then, although they generally found the automatically derived insights relevant and accurate, most participants (7/8) still manually verified each insight by investigating the related code, code outputs, visualizations, or NL explanations. Finally, after collecting enough insights or finishing a specific analytic topic, all participants checked the previous notes or screenshots to recap their findings and determine next-step explorations. However, during the entire process, participants encountered several common challenges that decreased their analysis efficiency, which are summarized below.

For clarity, we first 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 generated by LLMs for analytic purposes (code, code outputs, visualizations, and NL explanations), and data insights derived by either LLMs or users.
- **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: Repeated and tedious insight acquisition/verification from LLMs' responses. When identifying data insights, participants needed to acquire the relevant analytic context from LLMs' responses. All participants found the process repetitive and laborious, especially given the lengthy and cumbersome conversation history. They complained that LLMs tended to *'elaborate too much on the potential reasons behind each conclusion'* (E3), which forced them to *'manually locate and summarize the key information instead of gaining intuitive takeaways'* (E1). The situation was exacerbated when verifying insights, because participants had to locate other context (*e.g.*, code and visualizations) as insight evidence and associate them with each insight manually. For example, E5 spent much time in *'scrolling back to find that particular number in code outputs'* to ensure correctness when she saw numerical values. Moreover, when participants iteratively modified their prompts for expected analysis results, the insight evidence would span across multiple responses, leading to extra manual effort for navigation.

C2: Significant overhead for insight organization. When interpreting the collected insights, most participants (7/8) managed to organize them into meaningful subgroups either based on data attributes or analytic topics. E7 explained that *'effective organizations helped him better reuse data findings in presentations and documentations'*. However, 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. As the linear chat-based interfaces suffered in effective insight management, participants resorted to external tools (*e.g.*, Typora, Word) to document their acquired insights and other analytic context in notes or screenshots. Nevertheless, as the analysis progressed, the document quickly became overwhelming and was filled with *'too much unordered text and images'* (E5), which posed further challenges to structured organization. Meanwhile, the frequent switching between different applications was highlighted as *'frustrating and time-consuming'* (E3).

C3: Inflexible and inefficient insight browsing and revisiting. Participants commonly expressed the need to revisit and explore previous findings throughout the analysis process. They reported that the lack of a high-level overview, such as a *'timeline'* (E4) or *'minimap'* (E7), hindered quick navigation and contextual understanding. The extra cognitive overload for insight exploration mainly reflected in two aspects. First, it was inconvenient to browse insights. For example, E3 maintained an outline of her discoveries in Word, but the document soon became lengthy and forced her to *'repeatedly scroll up and down to browse each section'*, which *'somewhat outweighed the advantages of organizing insights'* (E3). Moreover, as the quality of LLM-generated insights may differ, participants desired to prioritize significant insights during exploration instead of *'random meandering'* (E4), which was not supported. Second, it was cumbersome to revisit previous related insights and their supporting evidence (*e.g.*, visualizations), 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 interfaces when working with LLMs. To this end, we aim to design a novel interactive system for better extraction, association, organization, and exploration of insights to facilitate a more efficient data analysis experience. The design requirements can be summarized as follows.

R1: Support automatic insight extraction and association from LLMs' responses. Manual extraction and association of insights with relevant evidence from LLMs' lengthy responses are tedious and error-prone (**C1**). Therefore, the system should constantly monitor the conversations to automatically extract insights and insight evidence, as well as establish and maintain associations between them.

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, resorting to external tools incurs extra manual effort and cognitive overload. Hence, the system should organize insights along with the analysis process.

R3: Provide multi-level and multi-faceted insight exploration. Exploring previous insights and other analytic context from multiple aspects or levels of detail were non-intuitive and burdensome (**C3**). Therefore, the system should allow multi-faceted insight exploration (*e.g.*, temporal, data attributes, analytic topics). Additionally, insight interestingness [81] and context transitions [59] should be highlighted to help users quickly identify important insights and enhance analytic

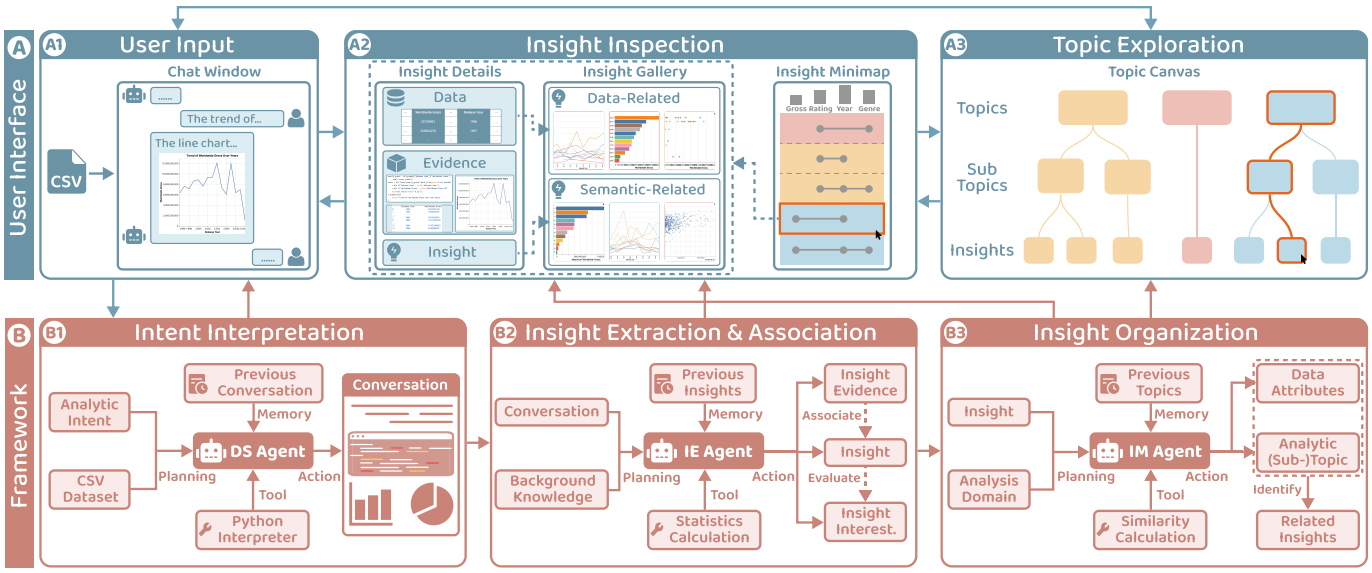

Fig. 2: *InsightLens* consists of (A) a user interface and (B) a multi-agent framework. Users (A1) upload a dataset and specify their analytic intent. The *Data Science (DS) Agent* (B1) interprets the intent, initiating a conversation cycle that is forwarded to the *Insight Extraction (IE) Agent* (B2) for insight extraction and evidence association. Following this, the *Insight Management (IM) Agent* (B3) organizes the insights by identifying their data attributes, analytic topics, and related insights. Users can then (A2) inspect the extracted insights and (A3) explore the structured topics.

comprehensiveness. To facilitate easier navigation and inspection of insights, an insight-level overview should be offered, with details on demand to reveal the supporting evidence and other related insights.

R4: Adopt familiar and unobtrusive interactions and visual designs. Users generally appreciate the conversational manner for its intuitive and user-friendly interaction with LLMs. Therefore, enhancing existing interfaces seamlessly with appropriate 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 flexible interactions that cater to different user needs, without disrupting the original chat-based workflow.

4 Framework for LLM-Powered Data Analysis

We develop a multi-agent framework (Figure 2B) to automatically extract, associate, and organize insights. Each agent, functioned by an LLM and equipped with specialized tools and in-context memory, plans and executes actionable steps to perform different tasks. Initially, the *Data Science (DS) Agent* interacts with users to complete their analytic tasks, generating a conversation cycle. This conversation cycle is then passed to the *Insight Extraction (IE) Agent*, which extracts insights from the conversation and associates them with the relevant insight evidence (**R1**). Meanwhile, the *IE Agent* evaluates the extracted insights' interestingness based on their semantic and statistical significance (**R3**). Subsequently, the *Insight Management (IM) Agent* examines the insights' data and semantic characteristics and dynamically organize them with previous insights (**R2, R3**). Throughout the conversation cycles, *InsightLens* iteratively updates the visualizations (Figure 2A) to facilitate flexible and efficient insight exploration from multiple aspects and levels of detail (**R3**). This section describes our prompting techniques, with the next section introducing our user interface.

4.1 Intent Interpretation

As the entry point of our framework, the *DS Agent* writes, executes code, and generates insights along with various intermediate outputs to address users' analytic intents (Figure 2B1). We utilize Open Interpreter [1] to provide a local code execution environment for the agent. Moreover, we adopt the ReAct (Reasoning and Acting) [74] paradigm for prompting, which requires the agent to think step-by-step and adapt its actions based on prior observations. During each conversation cycle, the agent first formulates a plan with actionable steps tailored to the dataset and analytic intent, and then sequentially executes each step to fulfill the analytic needs. At each step, the agent determines its next action (*e.g.*, refining code, generating insights) by observing previous code execution results and the current analytic stage. This process concludes when the agent has derived sufficient insights to adequately address the analytic intent. To ensure the validity and reliability of the generated insights, we instruct the agent to provide substantial intermediate outputs in its responses, such as code outputs and visualizations.

4.2 Insight Extraction and Association

To support automatic insight extraction and association (**R1**), the *IE Agent* keeps monitoring the conversation history along with the analysis process (Figure 2B2). The design of its prompt 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*, drawing from previous literature [13, 69] and our formative study. This allows the agent to be familiar with the essential domain knowledge, facilitating improved 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 information relevant to the data and analytic context, instead of extracting unrelated insights. Finally, we underscore the task and its objectives with a few demonstration examples to better leverage LLMs' in-context learning [9] abilities for desired results.

Identifying/Refining insights. For each conversation cycle, we instruct the agent to carefully examine it and determine whether it contains insights and/or other analytic context. Meanwhile, we maintain the previously identified insights as the agent's memory, which not only helps it leverage in-context learning to extract insights in a consistent manner, but also enables the refinement of previous insights. During analytic conversations, users may not always pose a new analytic intent every time; they often adjust their prompts for clarification or enhancement [11]. For example, users may request an alternative visualization to better illustrate a derived 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 conversation cycle without missing key information. Moreover, rather than replicating LLMs' lengthy responses, the extracted insights are *summarized* into concise sentences for intuitive understanding.

Associating insight evidence. To automatically bind all relevant insight evidence with each insight, the agent is required to scrutinize

[1] https://github.com/KillianLucas/open-interpreter/

the code, code outputs, visualizations, and NL explanations in each conversation cycle, 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' cognitive load in understanding and verifying insights without having to examine the entire contexts in LLMs' responses. Meanwhile, the previous insights are also taken into consideration for potential modifications or additions, in case that new evidence may emerge due to users' iterative prompting.

Evaluating insight interestingness. Inspired by QuickInsights [13], we judge the interestingness of an insight (**R3**) by two factors: its *semantic significance* (*i.e.*, the subject of it should be important, such as a best-selling product) and its *statistical significance* (*i.e.*, the relevant statistical metrics of it should be notable, such as a high standard deviation). To achieve this, the agent first evaluates each insight's semantic meaning to determine its importance. Then, it categorizes the insights and utilizes function calls to calculate their corresponding statistical metrics. We borrow ideas from previous literature for categorizing insights [69] and mapping insight categories to appropriate statistical functions [59]. Accordingly, the agent assigns a numerical interestingness score ranging from 1 to 5. To ensure scoring consistency, previous interestingness scores are also provided for reference.

4.3 Insight Organization

To organize insights from multiple aspects along with the analysis process (**R2, R3**), the *IM Agent* receives the extracted insights and examines their data and semantic characteristics to categorize them into subgroups based on data attributes and analytic topics (Figure 2B3).

Providing overall analysis domain. To ensure the generation of valid data attributes and relevant analytic topics each time, we provide an automatically identified short summary of the dataset and a list of its attributes beforehand. This enables the agent to gain an overall understanding of the analysis domain to facilitate insight organization.

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 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 the data context of each insight to support users' detailed inspection needs.

Classifying into topics/subtopics. As LLM-powered data analysis is a dynamic process, the complete set of insights cannot be predetermined, making traditional topic modeling techniques (*e.g.*, LDA) inapplicable. Therefore, we propose a novel topic classification method to sequentially assign analytic topics for each newly extracted insight.

1. First, we maintain a list of analytic topics derived from the previous conversation (and insights) as the agent's memory.
2. Then, the agent is instructed to `select` a suitable topic from the list that best describes the semantic meaning of the insight. To combine LLMs' NL understanding abilities with a best practice from prior literature [51], we provide cosine similarities between the embeddings of the insight and each existing topic for reference, which enables the agent to make more informed decisions.
3. In cases where no existing topics correspond to the insight, or when the topic list is empty at the start of each conversation, the agent is required to `generate` an appropriate analytic topic. The new topic should be under the provided analysis domain and be broad enough to encompass potentially similar subsequent insights. To avoid the generation of identical or overlapping topics as much as possible, 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 as 0.55. Once any similarity score exceeds the threshold, the agent has to generate another new candidate topic.
4. Finally, the selected or generated analytic topic for the insight is determined. We then recursively execute the above steps to classify subtopics within the assigned main topic.

We employ this method to organize the extracted insights semantically in a reliable and structured way during each conversation cycle.

Identifying related insights. After obtaining the corresponding data attributes and analytic topics of the extracted insights, we categorize them into subgroups to enable user exploration from different aspects. Moreover, we determine the related insights across two dimensions. 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. By linking these insights together, we address the common user need for easier reference or comparison of similar data findings.

5 InsightLens

We develop *InsightLens* (Figure 2A), an interactive system that builds upon the multi-agent framework to facilitate efficient insight discovery and exploration 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 in detail, including *User Input*, *Insight Inspection*, and *Topic Exploration*.

5.1 User Interface Overview

The user interface of *InsightLens* consists of five coordinated views (Figure 3). It is designed with the core principle of enhancing existing interfaces while maintaining users' original conversational workflow (**R4**). Given the unique nature of conversations which display the most amount of information at first glance, we sought advice from the data analysts in our formative study and improved our visual designs iteratively. Consequently, we choose to adopt a *'details first, overview last'* strategy [39] from left to right to make the user interface more applicable to the conversational workflow, while facilitating easy inspection and exploration of insights during the analysis process.

To achieve this, we keep the *Chat Window* (Figure 3A) similar to ChatGPT on the left, where users can input their analytic intents and view LLMs' responses. Beside it, the *Insight Details* (Figure 3B) shows an individual insight with its relevant data context and supporting evidence for thorough inspection, while the *Insight Gallery* (Figure 3C) displays its data- and semantic-related insights for convenient comparison. Additionally, we employ a matrix-based design in the *Insight Minimap* (Figure 3D) to chronologically visualize the analysis process. Each row represents a unique insight, showcasing its data and semantic characteristics. Finally, the *Topic Canvas* (Figure 3E) 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 *DS Agent* (Section 4.1) in the *Chat Window*. We adopt a streaming approach while generating LLMs' responses to mitigate system latency [73]. 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 the 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. Additionally, by clicking on each insight row, users can conveniently examine its details and navigate between conversation parts. Collectively, the visual designs and interactions support the following tasks to facilitate multi-level and multi-faceted insight exploration and fulfil various user needs (**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 satisfy different user background and preferences (*e.g.*, some analysts might not be familiar with coding and prefer to view data attributes or visualizations for verification and understanding). 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

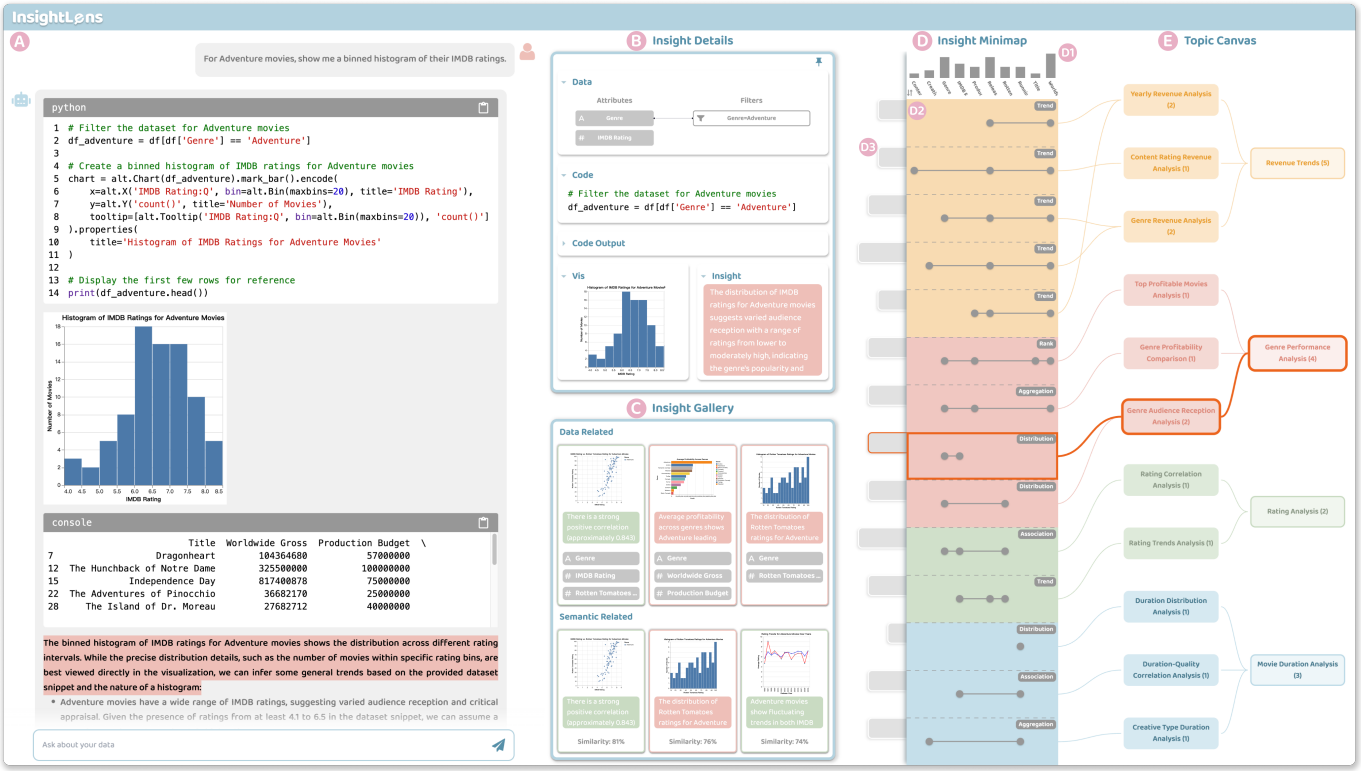

Fig. 3: The user interface of *InsightLens* consists of five views. The *Chat Window* (A) enables conversational interactions between users and LLMs. The *Insight Details* (B) displays the currently focused insight's summary with its relevant data context and supporting evidence. The *Insight Gallery* (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 throughout the conversation.

users' cognitive overload, enabling quick inspection and verification. 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 for focused examination of a specific insight.

Comparing related insights. In accordance with the currently focused insight displayed in 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 3D1) to visualize the distribution of the associated insight numbers 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 the analysis process can significantly be improved.

Understanding context transitions. In each insight row of the minimap (Figure 3D2), we represent its associated data attributes with a set of connected points (corresponding to the above histogram). This not only enables a quick review of each insight's data context, but also showcases context transitions throughout the analysis process. For example, certain visual patterns can represent different types of transitions like *continue* ( ), *retain* ( ), and *shift* ( ) [59]. In case that users expect to prioritize some attributes of interest, *e.g.*, always keeping track of '`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 intuitive and effective 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 3D3), 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 [57], 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 explore 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 number. These nodes are visually linked to their corresponding insight rows in the *Insight Minimap*. Additionally, hovering over any node will highlight its related 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 exploration across analytic topics.

6 Technical Evaluation

The effectiveness of *InsightLens* depends on whether our multi-agent framework can successfully extract, associate, and organize the gen-

erated insights during LLM-powered data analysis. Therefore, we conducted a technical evaluation focusing on (1) the *coverage* of insight extraction, (2) the *accuracy* of insight 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 labeled the key insights from the user perspective in the original responses generated by the *DS Agent*, 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 insight 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 evaluate insight organization, we focused on two aspects: data and semantic characteristics (see Section 4.3). For data context, we measured the ratio of insights with correctly identified data attributes (and analytical actions) 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 [10]. 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 [34] (*i.e.*, feeding all insights to GPT-4 for topic generation). Additionally, we 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 insight 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 produced 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 multi-agent 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 *DS Agent* 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 insight 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 *DS Agent* created new attributes for specific queries (*e.g.*, defining a `Decade` attribute from `Year`), leading to the *IM 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 LLMs 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 [77].

Summary. Despite the few failure cases, the results demonstrated our multi-agent framework's high coverage, accuracy, and quality in automatically extracting, associating, and organizing the generated insights in analytic conversations. This can significantly reduce users' manual and cognitive effort during LLM-powered data analysis, establishing a solid foundation for the interactive features of *InsightLens*.

7 User Study

To evaluate the effectiveness of *InsightLens* in facilitating insight discovery and exploration 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. Each had 4 to 8 years of experience in data analysis. 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.

For our comparative study, we set the *Baseline* as the *Chat Window* of *InsightLens*, excluding all interactive visualizations for insight inspection and exploration. This ChatGPT-like *Baseline* mirrored the systems participants currently used when interacting with LLMs. 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 [33], 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 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 in the experiments. To complement participants' qualitative feedback, we employed the following measures: (1) *number of confirmed insights*, (2) *number of unique data attributes explored*, and (3) *number of unique analytic topics explored*. These measures were informed by previous literature [15, 43, 66] 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 discovery and exploration (Figure 4). We then report *InsightLens*'s feature effectiveness, system usability, and impact on data analysis (Figure 5).

Support for Insight Discovery. The effectiveness of *InsightLens* in facilitating insight discovery was appreciated by all participants ($\mu = 4.67 > 2.67, p = .002$), while *Baseline* forced them to manually scrutinize and summarize insights from LLMs' lengthy responses. P3 expressed his favor for *'the dots below each message'* that *'reminded him of missed insights'*. Also, the highlighted NL explanations in each response were reported as *'being particularly useful for her to quickly identify key points'* (P11). Moreover, *InsightLens* significantly streamlined the verification of insights. We observed that participants constantly referred to the *Insight Details* to review the relevant insight evidence, which allowed them to *'easily see the involved attributes and visualizations without scrolling up and down'* (P10).

Additionally, one of our measures reinforced *InsightLens*'s support for insight discovery. Specifically, participants confirmed more insights using *InsightLens* compared to *Baseline* (Task 1: $\mu = 10.4 > 7.4, p = .002$; Task 2: $\mu = 11.1 > 7.3, p = .005$). By *confirming* an insight, they not only identified it, but also thoroughly verified its correctness. Therefore, we ascribed the observed significant difference to *InsightLens*'s support for reducing the time needed for verification, thereby leading to more insights discovered within a limited time frame.

Support for Insight Exploration. The effectiveness of *InsightLens* in exploring previously discovered insights received significantly higher ratings than *Baseline* ($\mu = 4.92 > 2.25, p = .002$). Participants highly appreciated *InsightLens*'s features for exploring insights from different aspects. For example, P4 commented that *'it was nice to track his findings by time order in the minimap'*, while *'using the baseline required him to navigate back and forth to grasp what he explored before'*.

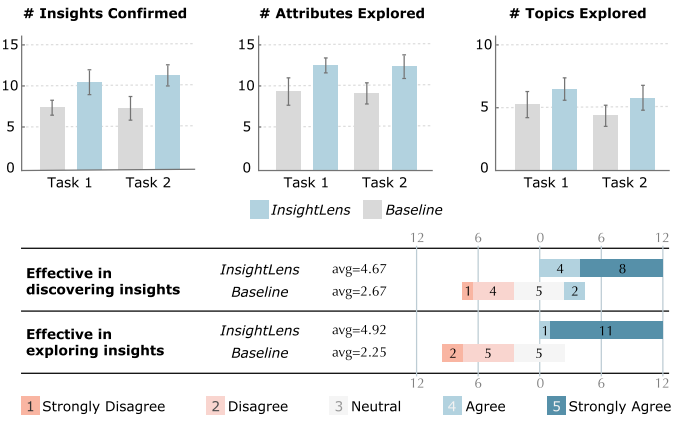

Fig. 4: The results of the measures and qualitative ratings regarding *InsightLens*'s support for insight discovery and exploration.

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

During open-ended data exploration, participants acknowledged the importance of keeping them aware of the overall analysis flow, which avoided *'repetitive analyses on a previously explored topic'* (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 many participants would check the *Insight Minimap* or *Topic Canvas* before posing their next analytic intent. Consequently, these observed significant differences implied participants' tendency to analyze more comprehensively when explicitly presenting the discovered insights organized across data and semantic dimensions.

Feature Effectiveness. Overall, the features of *InsightLens* received positive feedback from 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 check that specific line of code for data transformation'* (P6) to ensure relevance and correctness. Secondly, the *Insight Gallery* (Q2) helped participants review related insights conveniently. P8 found it particularly useful for *'understanding relationships between attributes 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. 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 agreed that *InsightLens* was easy to learn (Q5) and use (Q6), and were willing to integrate the system into their daily workflow (Q7). The visual designs and interfaces were described as *'very intuitive and user-friendly'* (P3, P7) without *'causing steep learning curves'* (P1). As stated by P9, the views looked

so natural that *'it should be easy for any professionals to understand its main features at first glance'*. Meanwhile, participants also noted some potential improvements for *InsightLens*. For example, P4 complained about LLMs' instability in analyzing complex problems, and P11 expected to *'combine certain insights for more in-depth analysis'*.

Impact on Data Analysis. We investigated *InsightLens*'s impact on the overall workflow of LLM-powered data analysis in terms of fluidity, workload, and understanding. Firstly, participants agreed that the system was unobtrusive and did not disrupt the conversational interaction (Q8). P9 commented that *'he just chatted with LLMs as usual, and the views would automatically update without any interference'*, while P7 thought *'the system was like a chat interface augmented with useful plugins'*. Secondly, participants' manual and cognitive overload could be mitigated by using *InsightLens* (Q9). The offered features alleviated the effort for *'excessive scrolling now and then'* (P2) and *'memorizing all insights in mind'* (P12). Moreover, organizing insights on the fly helped participants *'focus more on the analysis itself rather than constant context switching'* (P10). Finally, *InsightLens* could improve participants' understanding of the analyses generated by LLMs (Q10). P6 stated that *'it felt like she was participating more in the analysis process by inspecting the changes in different views to capture what was going on, instead of merely inputting a query and waiting for LLMs to handle everything'*. In other words, *InsightLens* helped strike a balance between automation and human agency, thereby increasing users' understanding and trust during LLM-powered data analysis.

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 analytic intents sequentially based on their own judgment and preferences, without being excessively intervened by the system. For example, P5 explored the colleges dataset centering around the ownership and its influence on factors such as student quality and financial condition. In such cases, the *Insight Minimap* and *Topic Canvas* primarily served as a structured and organized way for reviewing previous insights rather than inspiring new discoveries. Notably, the construction of the topic tree predominantly progressed from *bottom* (insights) to *top* (analytic topics) with much more subtopics than main topics, which revealed a depth-oriented exploration pattern.

System-Initiated Workflow. Participants without a specific aim (potentially due to their unfamiliarity with the analysis domain), on the other hand, commonly posed multiple random analytic intents at first to *'make a draft'* (P1). Then, they would inspect 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 determine their future explorations. Therefore, the construction of the topic tree was now from *top* to *bottom* with many topics scattered around and very few subtopics, which demonstrated a breadth-oriented exploration pattern.

8 Discussion

In this section, we reflect on our work and discuss its implications for designing human-LLM interfaces in data analysis, along with its limitations and potential future research directions.

8.1 Design Implications

Integrate data and semantic context for enhanced understanding. Given the inherent limitations of the linear chat-based interfaces, managing LLMs' conversational contexts for complex information tasks has emerged as a popular research topic in both VIS and HCI communities [29, 34, 63]. Going beyond existing work that primarily focuses on extracting the semantic structures of conversational utterances, *InsightLens* further integrates data context - an important factor of data analysis - including data attributes and analytical actions. We visualize the ever-changing data and semantic context simultaneously in a minimap, allowing users to quickly gain an overall understanding of the analysis process. Our user study indicates that such integration not only facilitates reviewing and navigation of different data insights, but also potentially expands data analysts' data and analytic coverage, leading to more comprehensive results during exploratory data analysis.

Provide follow-up analytic guidance for data exploration. In our user study, many participants (6/12) suggested incorporating query recommendations during analysis, especially when they faced an unfamiliar dataset (*i.e.*, the *'cold start'* issue). Providing analytic guidance has been extensively explored in previous literature [59, 68], which can further be improved with LLMs' extraordinary capabilities [20]. Meanwhile, *InsightLens*'s support for organizing insights on the fly can establish a robust foundation to inform context-aware assistance. For example, we can integrate another LLM-based agent into our framework, which receives analysts' background and goals as well as their current focused analytic topics and data attributes, and then generates appropriate suggestions to deepen or broaden their analyses.

Balance between the flexibility and complexity of interaction paradigms. The fundamental principle guiding our visual and interaction design is to maintain the conversational workflow, where the primary interaction modality is through natural language. Nevertheless, we admit the potential of leveraging other modalities or paradigms for NLI-based data analysis systems (*e.g.*, direct manipulations [60] and sticky cells [71]). One of the participants in our user study expected to modify the *Topic Canvas* by directly adding or editing nodes, similar to the operations in a mind map. Although such features could improve the flexibility of interaction with LLMs (which have been validated in many node-based LLM interfaces [3, 63]), they may also introduce increased complexity and steep learning curves [32]. Therefore, we aim to achieve a trade-off between NLIs' intuitiveness and visualizations' expressiveness. Future research could further explore how to balance these two aspects in designing interaction paradigms for LLMs.

8.2 Limitations and Future Work

Scalability. Our framework can theoretically support larger and more complex datasets without any limitations. To reduce the potential visual clutter in the *Insight Minimap* and *Topic Canvas* when very large numbers of data attributes or analytic topics are involved, we can employ graph visualization techniques such as fisheye [70], edge bundling [48], and semantic zoom [62], which we leave for future work.

Potentiality. Incorporating LLMs into data analysis is an emerging but promising paradigm. With LLMs' rapidly growing reasoning capabilities and extended context windows [6], data analysts can potentially be able to conduct longer and more in-depth analysis on intricate datasets with the help of intelligent *data copilots* [76]. Such envisions further emphasize the necessity of developing smart strategies to manage the complex conversational contexts during analysis. Therefore, we believe that our work could inspire future research in leveraging visualizations and other enhanced interaction techniques to make LLM-powered data analysis more streamlined, accessible, and productive.

Generalizability. While *InsightLens* is tailored to conversational data analysis, the design principles can be generalized to other usage scenarios of LLMs. For example, participants in our user study highly appreciated the *Insight Minimap*, whose fundamental idea is to chronologically display the entire conversation based on the visual abstraction of some domain-specific atomic units (a *data insight* in our case). Future work could adopt this minimap-based design in various applications (*e.g.*, conversational text-to-image generation [17]) with pre-defined units of each conversation cycle. Moreover, exploring the linear conversation in a non-linear, tree-based manner similar to the *Topic Canvas* is a promising paradigm worthy of further investigation in other creativity-driven processes (*e.g.*, story writing [75]).

9 Conclusion

This work presents *InsightLens*, an interactive system that visualizes the complex conversational contexts during LLM-powered data analysis to facilitate efficient insight discovery and exploration. Built upon an LLM-based multi-agent framework that streamlines the process of extracting, associating, and organizing insights in analytic conversations, *InsightLens* provides a set of interactive visualizations to enable multi-level and multi-faceted exploration. A technical evaluation and a user study demonstrate the effectiveness of our framework and system.

REFERENCES

[1] 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

[2] 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

[3] T. Angert, M. Suzara, J. Han, C. Pondoc, and H. Subramonyam. Spellburst: A node-based interface for exploratory creative coding with natural language prompts. In *Proc. UIST*. ACM, New York, NY, USA, 2023. doi: 10.1145/3586183.3606719 9

[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] Y. Chen, J. Yang, and W. Ribarsky. Toward effective insight management in visual analytics systems. In *PacificVis*, pp. 49–56, 2009. doi: 10.1109/PACIFICVIS.2009.4906837 2

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

[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, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.acl-long.870 7

[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 1, 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 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, New York, NY, USA, 2019. doi: 10.1145/3299869.3314037 4, 5

[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, Berlin, Heidelberg, 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, 8

[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] Y. Feng, X. Wang, K. K. Wong, S. Wang, Y. Lu, M. Zhu, B. Wang, and W. Chen. Promptmagician: Interactive prompt engineering for text-to-image creation. *IEEE Trans. Vis. Comput. Graph.*, 30(1):295–305, 2024. doi: 10.1109/TVCG.2023.3327168 9

[18] 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. ACM, New York, NY, USA, 2015. doi: 10.1145/2807442.2807478 2

[19] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. In *IEEE VAST*, pp. 123–130, 2008. doi: 10.1109/VAST.2008.4677365 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

[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, Florence, Italy, 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] 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

[26] 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 2

[27] Z. Huang, S. Gutierrez, H. Kamana, and S. Macneil. Memory sandbox: Transparent and interactive memory management for conversational agents. In *Proc. UIST*. ACM, New York, NY, USA, 2023. doi: 10.1145/3586182.3615796 2

[28] R. C. A. Iacob, F. Brad, E.-S. Apostol, C.-O. Truică, I. A. Hosu, and T. Rebedea. Neural approaches for natural language interfaces to databases: A survey. In *Proc. COLING*, pp. 381–395. International Committee on Computational Linguistics, Barcelona, Spain (Online), Dec. 2020. doi: 10.18653/v1/2020.coling-main.34 2

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

[30] A. Kamath and R. Das. A survey on semantic parsing. *arXiv*, 2019. doi: 10.48550/ARXIV.1812.00978 2

[31] 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, New York, NY, USA, 2019. doi: 10.1145/3290605.3300322 2

[32] 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, New York, NY, USA, 2015. doi: 10.1145/2678025.2701376 9

[33] 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

[34] 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.2309.10108 2, 7, 9

[35] 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 2

[36] 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, Singapore, Dec. 2023. doi: 10.18653/v1/2023.emnlp-industry.59 2

[37] Y. Liu, Z. Wen, L. Weng, O. Woodman, Y. Yang, and W. Chen. Sprout: Authoring programming tutorials with interactive visualization of large language model generation process. *arXiv*, 2023. doi: 10.48550/ARXIV.2312.01801 2

[38] 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 2

[39] 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

[40] 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, Singapore, Dec. 2023. doi: 10.18653/v1/2023.emnlp-demo.31 2

[41] K. Madanagopal, E. D. Ragan, and P. Benjamin. Analytic provenance in practice: The role of provenance in real-world visualization and data anal-

ysis environments. *IEEE Computer Graphics and Applications*, 39(6):30–45, 2019. doi: 10.1109/MCG.2019.2933419 2

[42] 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, New York, NY, USA, 2023. doi: 10.1145/3544548.3580940 2

[43] 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 8

[44] 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, New York, NY, USA, 2021. doi: 10.1145/3397481.3450667 2

[45] 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 2

[46] 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

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

[48] R. Pan, Z. Wang, Y. Wei, H. Gao, G. Ou, C. C. Cao, J. Xu, T. Xu, and W. Chen. Towards efficient visual simplification of computational graphs in deep neural networks. *IEEE Trans. Vis. Comput. Graph.*, pp. 1–14, 2022. doi: 10.1109/TVCG.2022.3230832 9

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

[50] 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

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

[52] 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

[53] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proc. UIST*, p. 365–377. ACM, New York, NY, USA, 2016. doi: 10.1145/2984511.2984588 1, 2

[54] V. Setlur and M. Tory. How do you converse with an analytical chatbot? revisiting gricean maxims for designing analytical conversational behavior. In *Proc. CHI*. ACM, New York, NY, USA, 2022. doi: 10.1145/3491102.3501972 2

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

[56] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang. Towards natural language interfaces for data visualization: A survey. *IEEE Trans. Vis. Comput. Graph.*, 29(6):3121–3144, 2023. doi: 10.1109/TVCG.2022.3148007 1, 2

[57] 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, 6

[58] A. Srinivasan, N. Nyapathy, B. Lee, S. M. Drucker, and J. Stasko. Collecting and characterizing natural language utterances for specifying data visualizations. In *Proc. CHI*. ACM, New York, NY, USA, 2021. doi: 10.1145/3411764.3445400 2

[59] A. Srinivasan and V. Setlur. Snowy: Recommending utterances for conversational visual analysis. In *Proc. UIST*, p. 864–880. ACM, New York, NY, USA, 2021. doi: 10.1145/3472749.3474792 1, 2, 3, 5, 6, 9

[60] 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

[61] H. Subramonyam, C. L. Pondoc, C. Seifert, M. Agrawala, and R. Pea. Bridging the gulf of envisioning: Cognitive design challenges in llm interfaces. *arXiv*, 2023. doi: 10.48550/ARXIV.2309.14459 2

[62] 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, 9

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

[64] 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

[65] 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

[66] K. Urgo and J. Arguello. Learning assessments in search-as-learning: A survey of prior work and opportunities for future research. *Information Processing Management*, 59(2):102821, 2022. doi: 10.1016/j.ipm.2021.102821 8

[67] 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, Online, July 2020. doi: 10.18653/v1/2020.acl-main.677 2

[68] 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 1, 9

[69] 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, 5

[70] Y. Wang, Y. Wang, H. Zhang, Y. Sun, C.-W. Fu, M. Sedlmair, B. Chen, and O. Deussen. Structure-aware fisheye views for efficient large graph exploration. *IEEE Trans. Vis. Comput. Graph.*, 25(1):566–575, 2019. doi: 10.1109/TVCG.2018.2864911 9

[71] 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

[72] T. Xie, C. H. Wu, P. Shi, R. Zhong, T. Scholak, M. Yasunaga, C.-S. Wu, M. Zhong, P. Yin, S. I. Wang, V. Zhong, B. Wang, C. Li, C. Boyle, A. Ni, Z. Yao, D. Radev, C. Xiong, L. Kong, R. Zhang, N. A. Smith, L. Zettlemoyer, and T. Yu. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proc. EMNLP*, pp. 602–631. ACL, Abu Dhabi, United Arab Emirates, Dec. 2022. doi: 10.18653/v1/2022.emnlp-main.39 2

[73] 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, 5

[74] 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

[75] A. Yuan, A. Coenen, E. Reif, and D. Ippolito. Wordcraft: Story writing with large language models. In *Proc. IUI*, p. 841–852. ACM, New York, NY, USA, 2022. doi: 10.1145/3490099.3511105 9

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

[77] 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 1, 7

[78] 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, New York, NY, USA, 2017. doi: 10.1145/3132847.3132977 2

[79] 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, Abu Dhabi, United Arab Emirates, Dec. 2022. doi: 10.18653/v1/2022.emnlp-main.145 2

[80] Z. Zhou, X. Wen, Y. Wang, and D. Gotz. Modeling and leveraging analytic focus during exploratory visual analysis. In *Proc. CHI*. ACM, New York, NY, USA, 2021. doi: 10.1145/3411764.3445674 2

[81] Çağatay Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Rapid data exploration through guideposts. *arXiv*, 2017. doi: 10.48550/ARXIV.1709.10513 3