Reconstructing Trajectories as Storylines: Steering LLM Agents in Narrative Space

Anonymous Author(s)

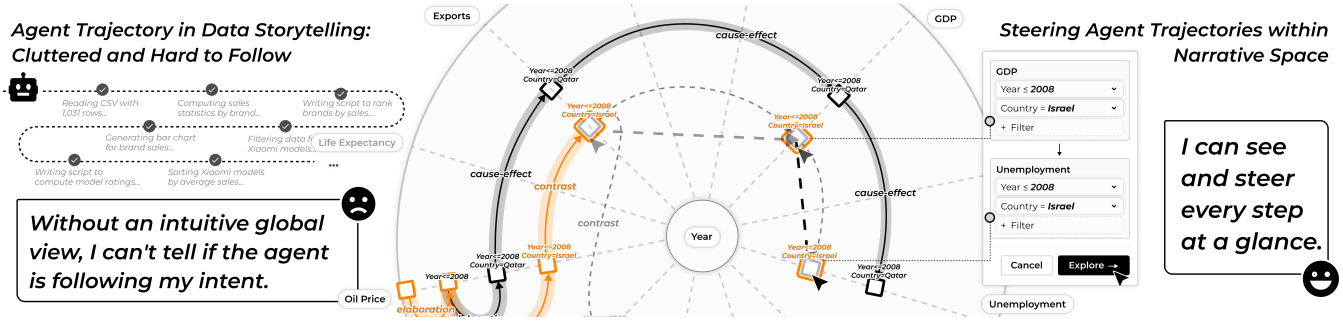

Figure 1: Steering LLM agents within narrative space. Compared to conventional agent interfaces where users are hard to maintain a clear global view of the agent's trajectory, NarraSteer reconstructs agent trajectories as storylines within a narrative space, allowing users to follow the storyline as it unfolds, identify narrative gaps, and steer the agent at a glance.

Abstract

Large language model agents produce multi-step trajectories to automate complex tasks but often leave users struggling to steer. Existing steering methods focus on single-step correction, offering no global view of the agent's trajectory that effective steering requires. To investigate this gap in data storytelling, we conducted a formative study ($N = 8$) revealing a fundamental misalignment between the agent's trajectory and the user's thinking flow that prevents effective steering. Based on these findings, we propose NarraSteer, a system that reconstructs agent trajectories as storylines and visualizes them within a narrative space, supporting users to follow the storyline as it unfolds, identify narrative gaps, and steer the agent at a glance. We evaluated NarraSteer through a user study ($N = 16$) comparing it against a tree-based baseline. Results show that NarraSteer significantly increases the insight count (+13.2%) of generated stories without compromising logicality, improves perceived control, and enables easier agent steering.

CCS Concepts

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Keywords

Do, Not, Use, This, Code, Put, the, Correct, Terms, for, Your, Paper

ACM Reference Format:

Anonymous Author(s). 2018. Reconstructing Trajectories as Storylines: Steering LLM Agents in Narrative Space. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

As Large Language Models (LLMs) have demonstrated impressive capabilities, researchers have increasingly used them to build autonomous agents across diverse domains, including software engineering [68], data analysis [18], and web navigation [74]. Unlike traditional LLMs that generate a single output, LLM agents operate over multi-step interaction **trajectories**, which consist of external observations, internal reasoning, and executable actions [43].

Although these agents can perform complex tasks autonomously, they often produce results misaligned with user intent. This stems from their multi-step black-box trajectories, where agents may gradually drift from the user's original goals [14]. To address this misalignment, one natural approach is to adapt existing steering techniques from traditional LLMs. Existing steering methods for traditional LLMs generally fall into two categories: pre-interaction and post-interaction. Pre-interaction methods enable users to better convey intent and prevent misunderstanding [4, 33, 65, 79–81]. Post-interaction methods enable users to efficiently modify or refine the results generated by LLMs [23, 61, 73, 76]. While both approaches can help users better steer LLM agents by clarifying intent, they cannot prevent agents from gradually drifting from the user's original goals across multi-step trajectories. Without an intuitive global view of what has been explored, what has been missed, and what directions remain, users struggle to identify such drift, making effective steering costly and difficult.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

In this paper, we study how to provide a global view of agent trajectories for effective steering in data storytelling. Data storytelling is particularly suited for this investigation, as its goals evolve dynamically throughout exploration [45], demanding continuous alignment between the agent's trajectory and the user's intent. To better understand what hinders users from gaining a global view of agent trajectories, we first conducted a formative study with 8 experienced data storytelling experts. Each participant was asked to guide an LLM agent in producing a slide-based data story while thinking aloud. Through observation and interviews, we identified a fundamental misalignment between the agent's trajectory and the user's thinking flow. The agent acts at the operational level, exposing only how tasks are executed (e.g., "reading CSV with 1,031 rows," "writing script to rank models by rating," "writing script to compare models"). In contrast, users think at the narrative level, reasoning about why a particular data pattern occurs (e.g., "why does Xiaomi outsell Samsung?" "what drives Poco's high ratings despite its low price?"). This misalignment prevents users from forming a global view of the agent's trajectory, making effective steering difficult. Building on these findings, we redesign the agent's action space to operate at the narrative level, enabling us to visualize the agent's trajectory within a narrative space. This provides users with an intuitive global view of what the agent has explored, what it has not, and what paths remain, allowing them to visually steer the agent-generated storyline. Finally, we evaluate our approach through a user study ($N = 16$) comparing NarraSteer against a tree-based baseline. Results show that NarraSteer significantly increases insight count without compromising logicality, improves perceived control, and enables users to steer the agent with less effort. Our contributions include:

- A formative study ($N = 8$) revealing a fundamental misalignment between the agent's trajectory and the user's narrative thinking, from which we derive three design goals.
- A proof-of-concept prototype that reconstructs agent trajectories as storylines within a narrative space, enabling users to follow, evaluate, and steer the storyline at a glance.
- A user study ($N = 16$) showing that NarraSteer increases the insight count of data stories, improves perceived control, and enables easier agent steering over a tree-based baseline.

2 Related Work

Our work builds upon research in LLM agents, LLM steering methods, and data storytelling systems. Despite progress in each area, existing approaches do not provide users with a global view of agent trajectories for effective steering, which motivates our approach.

2.1 LLM Agents

With the rapid development of LLMs, researchers have increasingly employed them as the reasoning core of autonomous agents [72]. Unlike traditional LLMs that generate a single output, LLM agents operate over multi-step interaction trajectories consisting of external observations, internal planning, and executable actions [43]. To construct such agents, two paradigms have been widely adopted:

One prominent paradigm is ReAct, which interleaves reasoning and action within an iterative loop [75]. Researchers have further enhanced ReAct agents through two main approaches. In-context learning methods enhance agent capabilities by enriching the prompting context. For example, Reflexion enables learning from past failures through episodic memory with verbal feedback [56]. CodeAct proposes executable python code as a unified action space for reasoning and tool execution [67]. Beyond in-context learning, training-based methods optimize LLM parameters to improve agent performance. For example, AgentTuning employs supervised fine-tuning on high-quality interaction trajectories data to generalize agent abilities [78]; ETO employs reinforcement learning on both successful and failed exploration trajectories to iteratively improve agent performance [57]. The other paradigm is Plan-and-Execute, which separates planning from action by decomposing tasks into structured plans before sequential execution [24, 60, 66]. To address the rigidity of static planning, recent work has introduced dynamic planning mechanisms. AdaPlanner dynamically revises its plan in response to execution failures, modifying the following plan logic to recover from unexpected environmental feedback [59]. ADaPT recursively decomposes sub-tasks as-needed when the agent fails to execute them [46].

Although these agents can perform complex tasks end-to-end, they often produce results misaligned with user intent. This misalignment arises because agents may gradually drift from user goals over multi-step trajectories. To address this, we provide users with an intuitive global view of the agent's trajectory. This global view enables users to see what has been explored and what remains, allowing them to identify drift and steer effectively.

2.2 Steering LLMs

Due to the ambiguity of natural language, LLMs often produce outputs that are misaligned with user intent. To address such misalignments, recent studies have introduced two types of steering methods: pre-interaction and post-interaction.

Pre-interaction methods enable users to clarify intent and better guide LLMs. DirectGPT, VizTA, and InterChat enable users to combine textual and visual inputs to convey intent to LLMs [4, 33, 65]; Patchview allows users to steer LLM-powered worldbuilding through a dust-and-magnet visualization [7]. LightVA and LEVA present users with recommended next steps, allowing them to guide the agent's visual analytics process through selection [80, 81]. Post-interaction methods enable users to efficiently refine the results generated by LLMs. Stepwise and Phasewise facilitate verifiable data analysis by structuring LLM output into editable phases [23]; NeuroSync externalizes the LLM's understanding of user intent as an editable interactive graph [79]. WaitGPT improves LLM data analysis by visualizing generated code as interactive graphs in real-time [73]; DynaVis facilitates visualization editing by transforming natural language commands into interactive widgets for immediate visual feedback [61]; CoLadder breaks coding tasks into steps with a tree structure for modular modification [76].

However, effective agent steering goes beyond clarifying intent at each turn. It requires an intuitive global view of the agent's trajectory. Users must be able to see where the agent has been, what it has overlooked, and what directions remain unexplored.

We therefore propose an approach that provides such a global view, enabling users to identify drift and steer effectively.

2.3 Data Storytelling

Data storytelling is an inherently iterative process involving cycles of data exploration and narrative organization [16, 45]. It requires users to extract insights from complex data and weave them into coherent narratives that effectively communicate findings. Existing surveys have provided comprehensive overviews of data storytelling authoring tools [29, 30], which allow users to move fluidly between data exploration and narrative construction.

For example, InsideInsights, Gravity, and Gravity++ map insights into visual networks in real time to help users organize complex narrative threads [34, 41, 42]. CLUE, SenseMap, GraphTrail, and Ellipsis capture exploration provenance as graphs, allowing users to reuse analysis steps when constructing data stories [11, 16, 40, 50]. More recently, LLMs have been introduced to further automate and enrich this process. Jupybara integrates LLMs into Jupyter Notebooks to transform analysis processes into data stories [64]. InsightLens employs LLMs to distill complex insights from conversation history to help users better organize data stories [71]. InReAcTable employs retrieval-augmented generation to recommend insights that are both relevant to the user's query and narratively coherent with the current analytical context [2].

Building on these capabilities, researchers have built LLM agents that generate data stories end-to-end from raw data [6, 20, 53, 54]. However, this end-to-end paradigm overlooks the iterative nature of data storytelling, where goals evolve dynamically and users need to steer the narrative as it unfolds. This motivates us to provide an intuitive global view of the agent's storyline construction process, so users can steer in time as ideas emerge.

3 Formative Study

To investigate what prevents users from gaining a global view of agent trajectories for effective steering in data storytelling, we conducted a formative study [1]. We first describe the study design, then present our key findings. Informed by these findings, we derive a set of design goals that guide the development of our system.

3.1 Study Design

3.1.1 Participants. We recruited 8 participants (3 *women*, 5 *men*; M_{age} = 24.25, SD = 4.02) with rich data storytelling experience across different disciplines (M_{exp} = 3.38, SD = 2.55). All participants have experience using LLM agents in their daily work (5+ days a week).

3.1.2 Apparatus and Tasks. We selected Manus [36] as our experimental platform, as it was a state-of-the-art publicly accessible LLM agent at the time of our study. In our study, participants were asked to create and iteratively refine data story slides based on a smartphone sales dataset [3]. The final slides needed to address two questions: (1) *Which models are the best-sellers?* (2) *Which models should be recommended to customers?* Upon completion, participants presented their final slides to the experimenter, which encouraged active engagement throughout the task.

[1] This study was approved by [redacted]

3.1.3 Procedure. The study lasted about 70 minutes and consisted of three parts. Before the study, all participants reviewed and signed an informed consent form. Participants first completed a demographic questionnaire and were introduced to the study purpose and Manus platform, including hands-on practice (~15 min). They then completed a data storytelling task using Manus while thinking aloud, iterating until satisfied, and presented their final slides to the experimenter (~40 min). Finally, based on their task performance, participants were interviewed about their interaction behaviors, steering challenges, and desired steering mechanisms (~15 min). All task sessions were video-recorded and interview sessions were audio-recorded. Each participant received $10 compensation.

3.1.4 Data Analysis. To analyze user and agent activities, two researchers independently coded half of the video recordings based on think-aloud transcripts to establish initial categories, then collaboratively refined them until reaching consensus and applied them to the remaining videos. They also transcribed the interview recordings and conducted content analysis [25], with disagreements resolved through discussion.

3.2 Study Results

Study results (see Fig. 2) show that completing a data storytelling task required an average of 9.88 iterations (SD = 3.30), with each iteration averaging 294.49 seconds (SD = 87.55). In each iteration, participants reviewed the previous output, submitted a new prompt, and Manus autonomously executed the task through a multi-step trajectory before presenting the results. We coded user activity into four categories: *Trajectory Review* (2071*s*, 9.06%), *Result Review* (6522*s*, 28.53%), *User Prompting* (5645*s*, 24.70%), and *Idle* (8619*s*, 37.71%). We coded agent activity into two categories: *Agent Working* (11386*s*, 49.81%) and *Idle* (11471*s*, 50.19%). Combining these behavioral data with interviews, we identified a fundamental misalignment between the agent's trajectory and the user's thinking flow. The agent acts at the operational level, exposing only how tasks are executed (e.g., "reading CSV with 1,031 rows," "writing script to rank models by rating," "writing script to compare models"). In contrast, users think at the narrative level, reasoning about why a particular data pattern occurs (e.g., "why does Xiaomi outsell Samsung?" "what drives Poco's high ratings despite its low price?"). We next present how this misalignment prevents users from forming a global view of the agent's trajectory and the resulting story:

3.2.1 Hard to Tell When the Story Drifts. While the agent was running, **its trajectory only exposed how the data was explored, but revealed nothing about the reasoning or findings behind it, leaving participants no chance to catch the drift in time (B1).** As P2 noted, *"I could see the agent was running Python scripts to analyze brand sales and then drilling into model sales of Poco, but I had no idea why it chose that direction or how it planned to shape the story."* Unable to connect what the agent was doing to where the story was going, participants defaulted to waiting rather than steering. As a result, **participants sat idle for 74.10% of the agent's working time, and most (5/8) did not notice misalignments until the agent had finished (O1).**

Figure 2: Activity timelines of each study session. Each timeline consists of two rows: the upper row shows user activities and the lower row shows agent activities. Timelines are scaled to equal length for comparison. Idle time is prominent across all participants, particularly during agent working phases.

Table 1: This table summarizes key findings from our formative study, highlighting observed behaviors (O1–O3), steering barriers (B1–B3), and corresponding design goals (DG1–DG3).

Observation	Steering Barrier	Design Goal
O1: Idle During Agent Working. Participants sat idle for 74.10% of agent working time; most (5/8) noticed misalignments only after the agent had finished	**B1: Hard to Tell When the Story Drifts.** The trajectory exposed only operational steps, not the narrative reasoning behind them, leaving participants no chance to catch drift in time	**DG1: Redesign Agent Actions at the Narrative Level.** Reframe agent actions at the narrative level so users can interpret and steer each step in terms of storytelling goals
O2: Repeated Trajectory-Story Switching. 4/8 participants repeatedly switched between the trajectory and the story to trace each conclusion	**B2: Hard to Trace How the Story Was Built.** The trajectory bore no explicit link to the resulting story, leaving participants struggling to trace how the agent arrived at each conclusion	**DG2: Reconstruct Agent Trajectories as Storylines.** Chain narrative-level actions into a storyline so the trajectory directly mirrors how the story was built
O3: Costly Context Building. Participants spent 29.41% of their time reviewing results to build context, exceeding the 24.70% spent formulating prompts	**B3: Hard to See Where the Story Could Go.** The trajectory carried no narrative structure, making it hard for participants to build a context of what had been explored and what remained	**DG3: Visualize Agent Trajectories within Narrative Space.** Visualize the trajectory within a narrative space so users can see explored and unexplored directions at a glance

3.2.2 Hard to Trace How the Story Was Built. After the agent presented the final story, **its trajectory bore no explicit link to the resulting story, leaving participants struggling to trace how the agent arrived at its conclusions (B2).** As P1 noted, *"The story claimed Poco had the highest average sales per model because of its processor performance, but that didn't feel right. I wanted to verify it, but I couldn't tell which step in the trajectory led to that conclusion."* Without a clear mapping between the trajectory and the story, participants could only check the trajectory step by step against the story. P1 added, *"Every conclusion had to be verified this way, which was really time-consuming."* As a result, **several participants (4/8) repeatedly switched between the trajectory and agent-generated story, trying to reconstruct how each conclusion was reached (O2).**

3.2.3 Hard to See Where the Story Could Go. **The agent's trajectory carried no narrative structure, making it hard for participants to build a context of what the agent had explored and what remained (B3).** As P7 noted, *"I couldn't get any narrative logic from the trajectory, and when the slides came out all at once, it was hard to absorb. I had to spend a lot of time going through them just to figure out what had been covered. Otherwise I wouldn't even know what to ask next."* This experience was common across participants. **Participants spent 28.53% of their time reviewing the agent-generated results, exceeding the 24.70% spent formulating prompts (O3)**, suggesting that building a narrative context from the agent's output was a significant bottleneck for deciding where to go.

Table 2: Six narrative transition types and their corresponding data changes. To ensure narrative consistency [19], each transition differs from the current node by exactly one substitution in *dim*, *measure*, or $filters$.

Relation	Narrative Meaning	Data Change	Example
similarity	The two nodes share similar narrative logic	Replace dim_i or $measure_i$	*dims*: [brand] → [model]
contrast	The two nodes present contrasting patterns	Replace dim_i or $measure_i$	*measure*: [price] → [sales]
elaboration	The latter node elaborates on the previous one with finer details	Replace dim_i with its breakdown dim; or add a condition in $filters_i$	*dim*: [brand] → [model]
generalization	The latter node generalizes the previous one to a broader level	Replace dim_i with its aggregation dim; or remove a condition from $filters_i$	*dim*: [model] → [brand]
cause-effect	There exists a causal relation between the two nodes	Replace $measure_i$ with a causal measure	*measure*: [price] → [sales]
sequence	The two nodes occur in a specific temporal order	Replace dim_i with a temporal dim; or replace a time value in $filters_i$	$filters$: [year=2023] → [year=2024]

3.3 Design Goals

Informed by the three observations (**O1–O3**) and steering barriers (**B1–B3**), we derive three design goals (**DG1–DG3**) to bridge the misalignment between the agent's trajectory and the user's thinking flow (see Table 1).

DG1: Redesign Agent Actions at the Narrative Level (derived from B1 & O1). To help users tell when the story drifts, we redesign agent actions at the narrative level, exposing the narrative reasoning behind each action. This allows users to steer in time.

DG2: Reconstruct Agent Trajectories as Storylines (derived from B2 & O2). To help users trace how the story was built, we chain narrative-level actions so that the trajectory directly mirrors the storyline. This allows users to follow the storyline as it unfolds.

DG3: Visualize Agent Trajectories within Narrative Space (derived from B3 & O3). To help users see where the story could go, we visualize trajectories within narrative space, revealing explored and unexplored directions. This allows users to steer at a glance.

4 *NarraSteer*

To address the three design goals, we introduce *NarraSteer* (Fig. 3). We first present the definition of data storyline as the agent's narrative-level action space (**DG1**, Sec. 4.1), then describe how the agent constructs it by proposing and selecting narrative transitions (**DG2**, Sec. 4.2), and visualize it within the narrative (**DG3**, Sec. 4.3). Finally, we demonstrate a usage scenario that illustrates how users steer the agent on NarraSteer (Sec. 4.4).

4.1 Definition of Data Storyline

Given a tabular dataset D and a user's exploration *scope*, *focus*, and narrative *goal*, a data storyline is constructed as a tree:

$$Story := (goal, focus, scope, r, N, E)$$

where $scope \subseteq D$ is the set of attributes (columns) selected for exploration; $focus \subseteq scope$ is the set of attributes that define the narrative focus of the story; each node $n_i \in N$ represents an exploration within *scope*; each edge $e_i \in E$ represents the narrative transition from n_j to n_i, where n_j is the parent of n_i; and $r = n_0$ denotes the root node.

4.1.1 nodes. We define each story node $n_i \in N$ as a four-tuple representing a data exploration within *scope*:

$$n_i := (measure_i, dim_i, filters_i, insights_i)$$

where $measure_i \in scope$ is a numerical attribute being analyzed (e.g., sales, price), and $dim_i \in scope$ is a categorical attribute being explored (e.g., brand, year), with $(measure_i \cup dim_i) \cap focus \neq \emptyset$ ensuring each node remains anchored to the narrative focus; $filters_i$ is an array of filter conditions defining the data subspace being explored (e.g., $brand = Samsung$); and $insights_i$ is a set of data insights discovered within this context, where each element is defined as:

$$insight := (type, score, description)$$

where *type* is the insight type, drawn from 12 types identified in prior work [28, 69]: *dominance*, *outstanding_positive*, *outstanding_negative*, *outlier*, *trend*, *evenness*, *mean_change*, *slope_change*, *correlation*, *cross_measure_correlation*, $fallback$; $score \in [0, 1]$ quantifies the significance of the insight, following the scoring method of QuickInsights [8]; and *description* is a template-based natural language description of the insight. Among these insights, $fallback$ is a special type when the user explicitly directs exploration but no other insight is found; it carries a score of 0 and an LLM-generated *description* based on the chart rendered from $n_i.measure$, $n_i.dim$, and $n_i.filters$.

4.1.2 edges. We define each edge $e_i \in E$ as a narrative transition from n_j to n_i:

$$e_i := (n_j, n_i, src\_insight_j, tgt\_insight_i, narrative_i)$$

where $src\_insight_j \in insights_j$ and $tgt\_insight_i \in insights_i$ are the insights that anchor the transition at each end; and $narrative_i := (rel_i, reason_i)$, where rel_i is the type of narrative relation drawn from previous work (Table 2) [55, 82] and $reason_i$ articulates the narrative logic behind the rel_i transition from $src\_insight_j$ to $tgt\_insight_i$.

4.2 Storyline Construction

Given a narrative *goal*, dataset D, $focus$, *scope*, and root node r, the LLM agent iteratively expands the storyline starting from r by identifying candidate transitions from the current node n_i (Sec. 4.2.1) and selecting the most appropriate edge e_k^* along with its target node n_k^* (Sec. 4.2.2). This process repeats from n_k^* until no further meaningful transition can be identified.

4.2.1 Candidate Identification. Given the current node n_i and its incoming edge e_i, along with the narrative *goal*, the agent follows the ReAct paradigm [75] to identify candidate transitions. It interleaves Chain-of-Thought reasoning [70] with four tool calls, using the current storyline and few-shot examples [10] as context: `browse_columns` and `browse_column_values` for exploring data within *scope* and determining a goal-relevant candidate node n_k; `calc_insights` for computing insights in n_k, retaining only those with score $\geq \theta$ (default 0.7 [28]) as candidate $tgt\_insight_k$; and `propose_next_step` for submitting an edge $e_k = (n_i, n_k, e_i.tgt\_insight, tgt\_insight_k, narrative_k)$ to C according to the narrative transition rules (Table 2). The agent autonomously determines the number of transitions to propose, up to three, stopping when no goal-relevant transition with valid insights can be identified.

4.2.2 Storyline Extension. Given the candidate set C, the current storyline, and the narrative *goal*, the agent selects the most appropriate transition by evaluating each candidate edge $e_k \in C$ via a multiple-choice prompt [21]. Since storylines vary considerably across users and datasets, few-shot examples are omitted; instead, the prompt provides explicit definitions of four evaluation dimensions adapted from [2]: logicality, coherence, diversity, and relevance. The agent selects the most appropriate candidate as e_k^*, and the storyline is updated as $Story' = Story \cup \{n_k^*, e_k^*\}$. Upon selecting e_k^*, the system generates a Vega-Lite chart for n_k^* based on $e_k^*.src\_insight$ and $e_k^*.tgt\_insight$, making $e_k^*.tgt\_insight$ visually salient while maintaining color consistency with the prior chart.

4.3 Visualizing Storyline within Narrative Space

To help users track how the storyline is constructed and identify unexplored narrative directions, we construct a disc with $focus$ at the center and other attribute $attr \in scope$ as an anchor distributed along the circumference. A line connects the center to each anchor, and each story node n_i is placed on the line between $n_i.measure$ and $n_i.dim$ (e.g., if $n_i.measure$ is *sales* and $n_i.dim$ is *brand*, n_i is placed on the line between *sales* and *brand*). Nodes on the same line are placed sequentially from the circumference toward the center, following the order they appear in the storyline. $n_i.filters$ do not affect node placement and are displayed as annotations directly on the node. The distribution of nodes across this space reveals the narrative directions covered by the storyline and those yet to be explored, forming a ***narrative space*** (Fig. 3d2).

Within the narrative space, each node n_i is rendered as a square marker oriented toward the center, and each edge e_i as a bezier curve labeled by $e_i.narrative.rel$. While the agent constructs the storyline, candidate edges are displayed as dashed lines and candidate nodes in gray. Once confirmed, they turn solid and are connected by filled curves. During storyline construction, users can drag the attribute anchors along the circumference to adjust the layout for better observability. Users can also steer the construction process through drag interactions. When dragging from a node, the system generates candidate nodes via attribute and filter substitution, displayed in gray within the narrative space. As the user drags through a candidate node without releasing, the system immediately generates the next set of candidates from that node, allowing users to steer the agent toward a specified trajectory in a single continuous drag (Fig. 3d4). Upon release, a confirmation dialog presents the dragged trajectory as a sequence of nodes, where users can review and adjust the filter conditions of each node before clicking *"Explore"* (Fig. 3d5). Clicking *"Explore"* to steer the agent to explore the specified storyline, determining the insights ($n_i.insights$) and narrative edges (e_i) along the way. Any node along this trajectory for which no significant insight is found will be assigned a $fallback$ insight. Together, this space allows users to follow the storyline as it unfolds, identify narrative gaps, and steer the agent at a glance.

4.4 Usage Scenario

We illustrate the envisioned usage of NarraSteer through Elena, a senior data storytelling expert. Elena is tasked with creating a data story from a Middle East oil production and economy dataset [52]. To complete the task efficiently while maintaining a sense of control, she opens NarraSteer. Elena first uploads the dataset to NarraSteer, where each attribute appears as a card showing its key statistics. To explore how oil prices have influenced economic indicators over time, she sets *Year* as the narrative *focus* (Fig. 3d1). A disc then appears with *Year* at its center and all other attributes as anchors along the circumference, forming the narrative space (Fig. 3d2).

She then prompts, *"How have rising oil prices shaped the economy of a typical country in the Middle East?"* The agent first explores [Oil_Price, Year], detecting a slope change at *Year=2012*: rising at $3.73/step before, then declining at $1.67/step after. Based on this node, proposes two candidate transitions: an *elaboration* edge to [Oil_Price, Year≤2012], capturing a steady rise from $18.0 to $105.0; and a *similarity* edge to [GDP, Year], where average GDP grew from $642B to $3,141B over the same period. After evaluating both on logicality, coherence, diversity, and relevance, the agent selects the *elaboration* edge. The agent continues, forming a storyline [Oil_Price, Year] → [Oil_Price, Year≤2012] → [Oil_Price, Year≤2012, Country=Qatar] → [Exports, Year≤2012, Country=Qatar] → [GDP, Year ≤2012, Country=Qatar] → [Unemployment, Year≤2012, Country= Qatar]: Qatar's exports climbed in lockstep with oil prices, GDP surged from $7.4B to $186.8B, and unemployment remained evenly low, averaging just 0.75% throughout the period (Fig. 3b). Throughout this process, each step of the agent's storyline construction is reflected in the narrative space, allowing Elena to easily tell whether the agent is drifting from her intent.

Having learned that Qatar's prosperity was entirely oil-driven, Elena becomes curious about a contrasting case. She prompts from [Oil_Price, Year≤2012]: *"Is there a country whose economy was less affected by rising oil prices?"* (Fig. 3c). The agent identifies Israel as the most representative case through tool calls, correctly adding [Oil_Price, Year≤2012, Country=Israel] → [Exports, Year≤2012, Country=Israel], but then selects [GDP_per_capita, Year≤2012, Country=Israel] instead of [GDP, Year≤2012, Country=Israel] as the next step. Noticing that this deviates from the parallel structure she

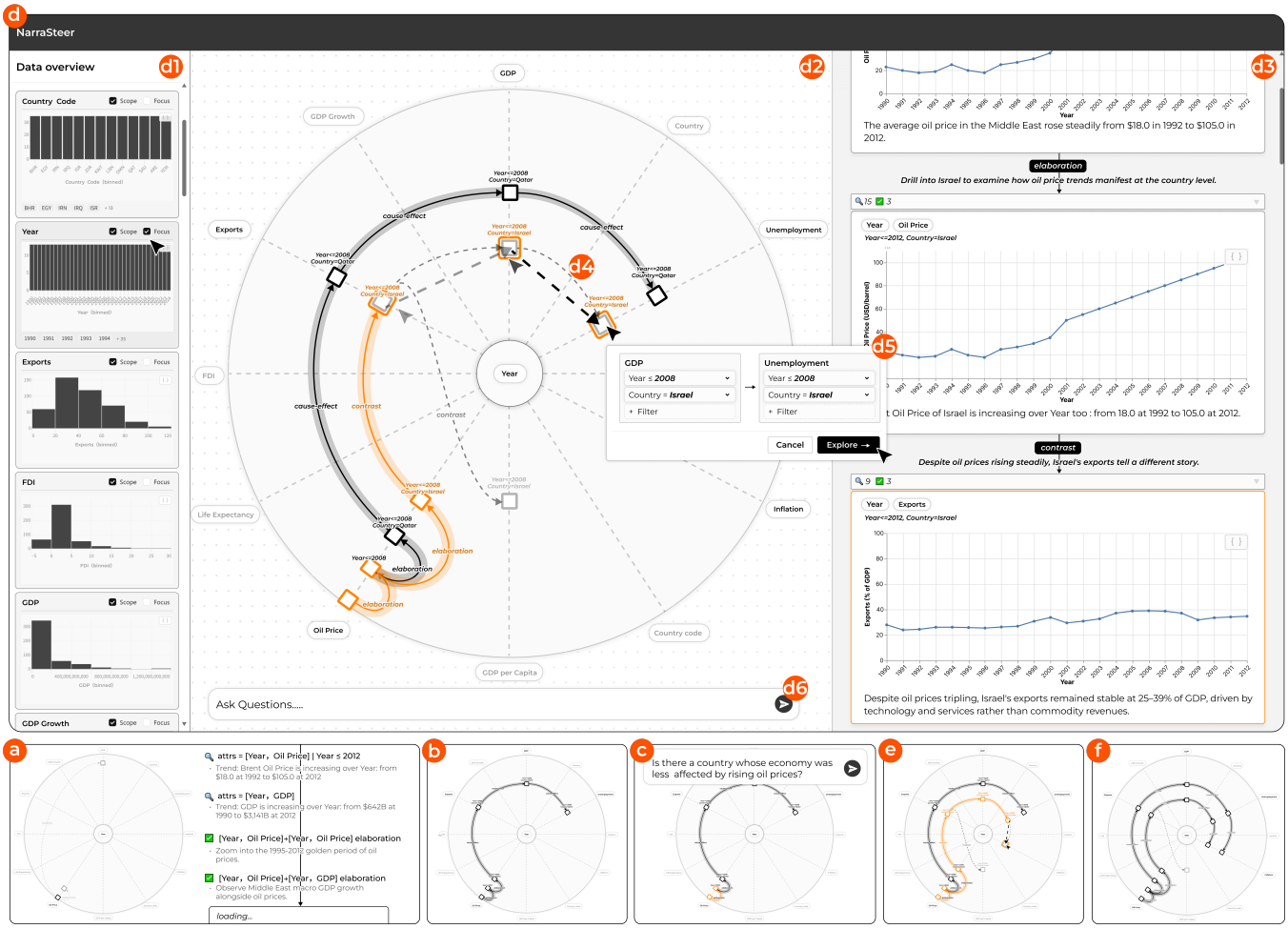

Figure 3: The interface of NarraSteer. Top (d): The interface consists of three panels: (d1) a data overview panel where users configure the narrative *scope* and *focus* for each attribute; (d2) a narrative space panel displaying the storyline as nodes placed on lines between the *focus* center and attribute anchors along the circumference, with the active storyline highlighted in orange and candidate nodes shown as gray nodes connected by dashed edges, labeled by narrative relation types; (d3) a story panel showing each node as a chart with its insight description, and each edge as a narrative transition labeled with its relation type and reasoning; (d6) a chat input box for users to prompt the agent. (d4) users can drag across candidate nodes to steer the agent toward a specified storyline; (d5) upon release, a confirmation dialog allows users to review and adjust filter conditions before clicking *"Explore"*. Bottom: Usage scenario snapshots (a–f); see Sec. 4.4 for details.

used for Qatar, Elena decides to steer the agent directly. She drags from [Exports, Year≤2012, Country=Israel] → [GDP, Year≤2012, Country=Israel] → [Unemployment, Year≤2012, Country=Israel], specifying the exact trajectory she wants (Fig. 3d4). A confirmation dialog appears showing the three-node trajectory; Elena verifies that the filters are correct and clicks *"Explore"* (Fig. 3d5). The agent abandons its exploration from [GDP_per_capita, Year≤2012, Country=Israel] and instead follows the specified path, revealing that Israel's exports remained stable at around 30% of GDP regardless of oil price fluctuations, GDP grew steadily from $62.0B to $263.6B, and unemployment improved from 13.5% in 2003 to 6.8% by 2012, driven by internal structural reform rather than oil revenues.

Satisfied with how effectively the drag interaction allowed her to steer the agent, Elena notices that the Inflation line remains unexplored for both countries (Fig. 3e). She uses the same drag interaction to add [Inflation, Year≤2012, Country=Qatar] and [Inflation, Year≤2012, Country=Israel] to complete both storylines. The resulting narrative space reveals a stark contrast: Qatar's economy was entirely oil-driven, with exports, GDP, and employment all rising in lockstep with oil prices; Israel's growth was decoupled from oil, with stable exports, steady GDP expansion, falling unemployment, and inflation declining to near zero through internal reform (Fig. 3f).

4.5 Implementation

We implemented NarraSteer with a decoupled frontend-backend architecture. The frontend uses Vue3 [63], TypeScript [37], and Vega-Lite [51] for chart rendering. The backend is built on FastAPI [47] with Google Gemini-3-flash-preview as the language model [15]. We employed LangChain [26] and LangGraph [27] to orchestrate a ReAct agent [75] that iteratively explores the dataset through tool calls, enabling structured multi-step reasoning over tabular data. Statistical insights are extracted using pandas [35] and scipy [62].

5 User Study

To evaluate whether NarraSteer's narrative space visualization enables more effective agent steering in data storytelling, we conducted a user study [2]. We first describe the study design, followed by a summary of the key findings.

5.1 Study Design

5.1.1 Participants. We recruited 16 participants (6 women, 10 men; $M_{age} = 23.44$, $SD = 2.50$) with data analysis experience ($M_{exp} = 2.44$ years, $SD = 1.36$). All participants reported using LLM agents in their daily work (5+ days per week). Three participants had also taken part in our formative study, whose prior experience enriched their post-task interview feedback. Each participant was compensated with $15 for their participation.

5.1.2 Baseline and Apparatus. Since our research goal is to investigate how to provide a global view of agent trajectories for effective steering, we do not compare against existing SOTA data storytelling systems [2]. These systems focus on story construction itself rather than steering the agent during the process. Instead, we adopt a baseline that renders the agent's trajectory as a node-link tree (see supplemental materials), as tree representations are widely used for LLM steering in prior SOTA LLM-based data storytelling work [9, 80]. To ensure a fair comparison, the baseline also adopts the redesigned storyline construction pipeline rather than general-purpose agent trajectories, as our formative study already established that the latter hinders effective steering. Both conditions support identical interaction modalities, including drag-based steering and text prompting.

5.1.3 Task. Participants completed a data storytelling task using a distinct tabular dataset: a macroeconomic indicators dataset [39] and a Middle East oil production dataset (the dataset used in Sec. 4.4) [52]. Each task consisted of two phases. participants were assigned the role of a data journalist and asked to identify a narrative focus of their choice, then steer the agent to explore story-worthy patterns about the narrative focus. In the second phase, participants reviewed their exploration history, selected the most compelling story segment, and articulated its narrative logic.

5.1.4 Procedure. The study lasted approximately 95 minutes and consisted of three phases. Participants first signed an informed consent form, completed a demographic questionnaire, and received a brief introduction to the study (~5 min). They then completed two conditions in counterbalanced order across four groups, each consisting of a system introduction (~15 min), an open-ended data exploration task (~10 min), a story consolidation task (~5 min), a post-condition questionnaire (~5 min), and a break (~10 min). Finally, participants took part in a post-study interview based on their questionnaire responses (~10 min). All sessions were video-recorded and audio-recorded with participants' consent.

5.1.5 Measures. We collected three types of measures: story quality, user perception, and user behavior. Story quality was assessed by recording insight count and insight uniqueness [48, 49], where insight uniqueness is defined as the inverse of the number of duplicated insights in the story. In addition, three experts (>= 2 years data storytelling experience) independently rated the logicality of each story on a 7-point Likert scale, evaluating whether narrative transitions between insights were coherent and well-justified. Inter-rater reliability was measured using the intraclass correlation coefficient (ICC). User perception was measured using the NASA-TLX [17] and a custom 4-item 7-point Likert scale for perceived control. User behavior was examined through screen recording analysis and post-study interviews.

5.1.6 Data Analysis. For quantitative measures, we used Wilcoxon signed-rank tests to compare NASA-TLX, perceived control, and logicality scores between conditions, and paired t-tests to compare insight count and insight uniqueness. Effect sizes were reported as rank-biserial correlations r for Wilcoxon tests and Cohen's d for paired t-tests. For qualitative data, two researchers independently conducted content analysis [25] on screen recordings and interview transcripts, with disagreements resolved through discussion.

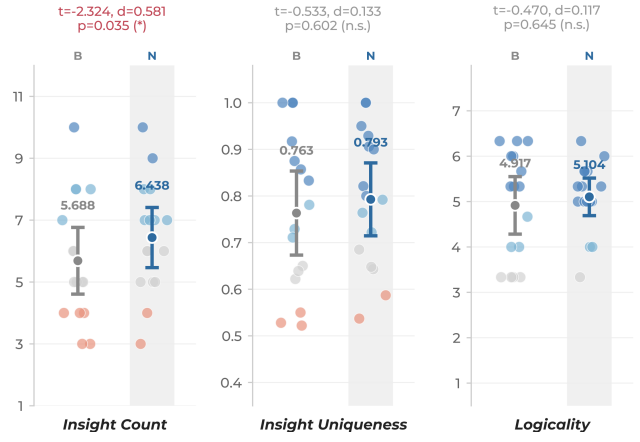

Figure 4: Insight count, insight uniqueness, and mean logicality ratings between NarraSteer (N) and baseline (B). Center dots and error bars show means with 95% CIs. Insight uniqueness is normalized by each participant's total insight count.

5.2 Study Results

Among the three participants who had also taken part in the formative study, all reported that both conditions offered greater controllability than Manus. As P1 noted, *"With Manus, I could only see things like running code or generating charts, which meant nothing to me. Here, I could see what the agent was actually looking at in the*

[2] This study was approved by [redacted]

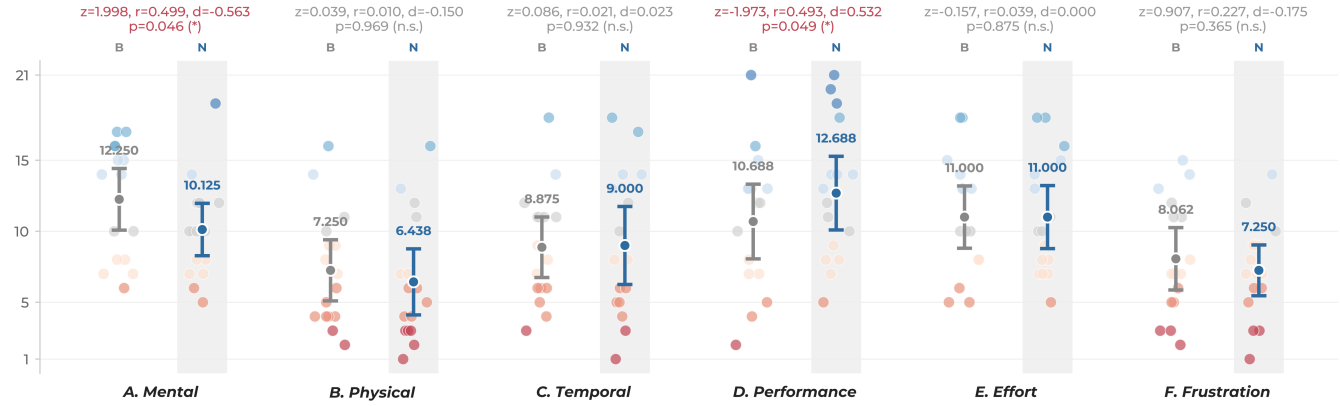

Figure 5: Ratings of Perceived workload between NarraSteer and baseline. Dots and error bars show means and 95% CIs. Performance (D) is reverse-scored, where higher values indicate better performance.

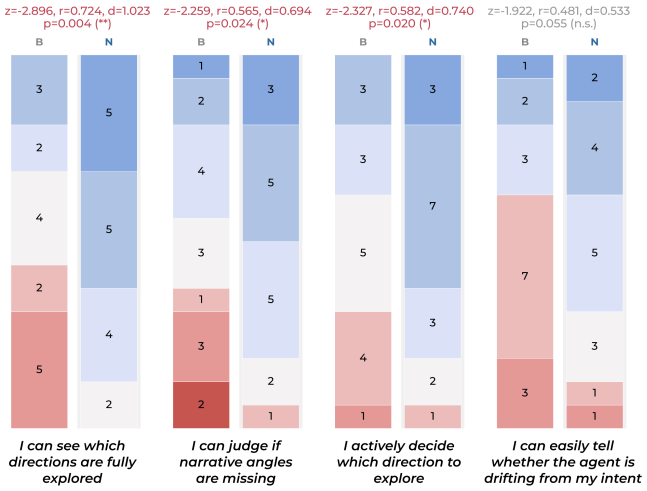

Figure 6: Ratings of perceived control between NarraSteer and baseline across four Likert-scale items. Red indicates lower agreement and blue indicates higher agreement.

data and where it was heading." P2 added, *"The way it was laid out made it much easier to read. I didn't have to dig through action logs to figure out what was going on."* The following sections report the quantitative and qualitative findings in detail.

5.2.1 Task Completion. Regarding task completion (Fig. 4), participants using NarraSteer generated significantly more insights than the baseline ($M_N = 6.438$ vs. $M_B = 5.688$; $Z = -2.324$, $p = 0.035$, $d = 0.581$), suggesting a broader narrative coverage. However, neither insight uniqueness ($p = 0.602$) nor expert-rated logicality ($p = 0.645$) showed significant differences between conditions. Inter-rater reliability for logicality was good ($\mathrm{ICC}(2,k) = 0.76$, 95% CI $[0.57, 0.87]$, $p < 0.001$). This suggests that NarraSteer enabled participants to explore more insights without compromising the logical coherence of their stories. Analysis of post-study interviews and screen recordings sheds light on the reasons behind this increase, which we discuss in Sec. 5.2.3.

5.2.2 User Perception. Regarding perceived workload (Fig. 5), participants using NarraSteer reported significantly lower mental demand ($M_N = 10.125$ vs. $M_B = 12.250$; $Z = 1.998$, $p = 0.046$, $r = 0.499$) and significantly higher perceived performance ($M_N = 12.688$ vs. $M_B = 10.688$; $Z = -1.973$, $p = 0.049$, $r = 0.493$). The lower mental demand likely stems from the synergy between narrative space visualization and drag interactions, reducing the effort of formulating prompts. The improved perceived performance may be attributed to the broader narrative coverage reflected in the higher number of insights. However, one participant (P15) reported notably higher mental demand under NarraSteer, which we discuss in Sec. 5.2.4. Other NASA-TLX dimensions showed no significant differences (all $p > 0.05$).

Regarding perceived control (Fig. 6), NarraSteer achieved significantly higher ratings for identifying fully explored directions (Q1: $Z = -2.896$, $p = 0.004$, $r = 0.724$), judging missing narrative angles (Q2: $Z = -2.259$, $p = 0.024$, $r = 0.565$), and actively deciding which direction to explore (Q3: $Z = -2.327$, $p = 0.020$, $r = 0.582$). Detecting agent drift showed a marginal trend but did not reach significance (Q4: $Z = -1.922$, $p = 0.055$, $r = 0.481$). Post-study interviews shed light on the reasons why some participants found drift detection still challenging, which we discuss Sec. 5.2.4.

5.2.3 User Behavior. Analysis of video recordings and interviews revealed distinct behavioral patterns when using NarraSteer.

NarraSteer Facilitates Easier Agent Steering. In the baseline condition, some participants (7/16) used drag interactions at most once. They relied primarily on follow-up prompts to explore new directions. As P5 explained, *"Dragging in the tree wasn't intuitive. When I tried to drag, all child nodes were just stacked below the parent regardless of what attributes they explored. Their position had nothing to do with their content, so I had no sense of where I was going."* In NarraSteer, nearly all participants (15/16) actively used drag interactions. As P5 noted, *"On the disk, I could tell what each node was about just from where it sat. So when I dragged toward*

a direction, I already knew what I was asking the agent to explore. It felt intuitive and fast." This difference suggests that NarraSteer provided a clearer global overview, enabling participants to steer the agent with less effort, more perceived control, and ultimately explore more directions in less time.

NarraSteer Facilitates timely Agent Steering. In the baseline condition, only a few participants (5/16) interrupted or redirected the agent during execution. As P9 explained, *"Relying solely on the text in the tree, it was sometimes hard to immediately judge whether the agent was heading in the right direction."* This difficulty was amplified when the dataset contained unfamiliar or similarly named attributes, which further slowed judgment of participants. In NarraSteer, this pattern shifted: more participants (10/16) actively interrupted or redirected the agent toward more relevant narrative directions. As P6 noted, *"Whenever the agent added a new node on the disk, it showed up right next to the attributes it was exploring, so I could immediately see whether it was going where I wanted. That was much easier than reading through text in the tree."* However, P6 also noted limitations: *"As the storyline grew, nodes moved closer to the center while attribute anchors stayed on the outer ring, making it a bit harder to see what each node was about."* This may explain the lack of significant improvement in Q4.

5.2.4 User Feedback. Post-study interviews revealed both positive experiences and areas for further improvement.

NarraSteer Supports Smoother Visual Navigation. Several participants (5/16) noted that NarraSteer reduced the need to switch attention between the data overview panel and the exploration panel. Unlike the baseline, where users frequently consulted the data overview to identify relevant attributes, NarraSteer encodes attribute information directly within the narrative space. As P12 noted, *"With the tree, I kept flipping back to the left panel to look up information about the dataset. With the disk, the attributes were already laid out around the circumference, so I no longer needed to do that. Everything felt more visually fluid."* This reduction in panel switching may partly account for the lower mental demand observed under NarraSteer.

NarraSteer Expands Narrative Possibilities. A few participants (4/16) noted that the baseline tree implied a fixed set of exploration paths, whereas NarraSteer made every point in the space feel like a potential direction. As P3 noted, *"With the tree, each node only had a few branches, so it felt like I had a limited set of choices. With the disk, it felt like a continuous space where I could go in any direction, which opened up my thinking."* P4 further distinguished the two: *"The tree felt more like a mind map that helped me organize existing ideas, while the disk felt more like brainstorming that opened up my thinking."*

Fast Generation Speed of Agents Hinders Effective Steering in NarraSteer. Some participants (2/16) found the agent's generation speed too fast to keep up with, leaving insufficient time to evaluate and steer before the next step was produced. As P15, who reported the highest mental demand and lowest Q4 rating, noted, *"Each prompt could trigger multiple nodes generated in quick succession. Even in the NarraSteer, things were moving so fast that I could barely tell whether the nodes were actually going in the direction I wanted."* P15 added, *"I found it easier to just wait for everything to finish and review it all at once."*

6 Discussion

Aligning Agent's Trajectory with User's Thinking Flow. Fully autonomous agents like Manus [36] perform well when task boundaries are clear, but struggle with open-ended tasks like data storytelling, where goals evolve dynamically throughout exploration. Our formative study reveals that the fundamental barrier is the misalignment between agent's trajectory and user's thinking flow, so merely exposing what the agent is doing does not improve understanding or control. NarraSteer addresses this by redesigning the agent's action space into human-interpretable narrative units that align with user thinking. This allows agent trajectories to be externalized in forms users can directly interpret and steer, resonating with recent work on externalizing LLM states [13, 58, 79].

Complexity of the Narrative Space. Data storytelling involves inherently complex narrative spaces: every combination of attributes and filters can yield distinct insights, and this combinatorial complexity grows rapidly with scope. We constrain this complexity by capping attributes per node at two, simplifying the narrative space layout for rapid comprehension. This also ensures that each insight can be conveyed through the two most effective visual encoding channels [32, 38]. Despite this simplification, the constraint covers the majority of insight types adopted from prior work [28, 69], with only one three-attribute exception: *scatterplot_clustering*. If wider coverage is needed, treating each attribute anchor as a magnet is one viable approach: nodes with three or more attributes can then be positioned through multi-magnet attraction [7, 22, 77]. This offers a path toward greater scalability.

Future Work. Our study also revealed several opportunities for improvement. Some participants found the agent too fast to steer, suggesting a need for pace control mechanisms, such as adjustable generation speed or pauses between transitions. Prior work suggests that such interaction can foster the reflective thinking that open-ended tasks demand [31, 44]. Additionally, some participants noted that longer storylines moved nodes farther from corresponding attribute anchors and made it harder to maintain a global view. This issue may be mitigated by collapsing non-active trajectories [1, 12], which could reduce visual clutter and help preserve overview clarity at scale. Beyond data storytelling, the principle of aligning the agent's trajectory with the user's thinking flow may apply to other open-ended analysis tasks, such as information foraging [58] and design ideation [5], to promote a sense of control.

7 Conclusion

We presented NarraSteer, a system that addresses the challenge of steering LLM agents in data storytelling. Through a formative study ($N = 8$), we identified a fundamental misalignment between the agent's trajectory and the user's narrative thinking. To bridge this gap, NarraSteer redesigns the agent's action space at the narrative level, reconstructs its trajectory as a storyline, and visualizes it within a narrative space. A user study ($N = 16$) showed that NarraSteer increases insight count, improves perceived control, and enables users to steer the agent with less effort. We hope that the core principle of aligning agent trajectories with user thinking flow can inform the design of future human-agent collaboration systems beyond data storytelling.

References

[1] James Abello, Frank van Ham, and Neeraj Krishnan. 2006. ASK-GraphView: A Large Scale Graph Visualization System. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 669–676. doi:10.1109/TVCG.2006.120

[2] Gerile Aodeng, Guozheng Li, Yunshan Feng, Qiyang Chen, Yu Zhang, and Chi Harold Liu. 2025. InReAcTable: LLM-Powered Interactive Visual Data Story Construction from Tabular Data. In *Proceedings of Annual ACM Symposium on User Interface Software and Technology*. 1–16. doi:10.1145/3746059.3747719

[3] Shubham Bathwal. 2026. Flipkart Mobile Dataset. Kaggle. https://www.kaggle.com/datasets/shubhambathwal/flipkart-mobile-dataset.

[4] Juntong Chen, Dongyu Liu, Jiang Wu, et al. 2025. InterChat: Enhancing Generative Visual Analytics using Multimodal Interactions. *Computer Graphics Forum* 44, 3 (2025). doi:10.1111/CGF.70112

[5] Pei Chen, Jiayi Yao, Zhuoyi Cheng, Yichen Cai, Jiayang Li, Weitao You, and Lingyun Sun. 2025. CoExploreDS: Framing and Advancing Collaborative Design Space Exploration Between Human and AI. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1–20. doi:10.1145/3706598.3713869

[6] Liqi Cheng, Dazhen Deng, Xiao Xie, Rihong Qiu, Mingliang Xu, and Yingcai Wu. 2025. SNIL: Generating Sports News From Insights With Large Language Models. *IEEE Transactions on Visualization and Computer Graphics* 31, 7 (2025), 3973–3986. doi:10.1109/TVCG.2024.3392683

[7] John Joon Young Chung and Max Kreminski. 2024. Patchview: LLM-Powered Worldbuilding with Generative Dust and Magnet Visualization. In *Proceedings of Annual ACM Symposium on User Interface Software and Technology*. 1–19. doi:10.1145/3654777.3676352

[8] Rui Ding, Shi Han, Yong Xu, Haidong Zhang, and Dongmei Zhang. 2019. QuickInsights: Quick and Automatic Discovery of Insights from Multi-Dimensional Data. In *Proceedings of ACM Conference on Management of Data*. 317–332. doi:10.1145/3299869.3314037

[9] Zijian Ding, Michelle Brachman, Joel Chan, and Werner Geyer. 2025. "The Diagram is like Guardrails": Structuring GenAI-assisted Hypotheses Exploration with an Interactive Shared Representation. In *Proceedings of ACM Conference on Creativity and Cognition*. 606–625. doi:10.1145/3698061.3726935

[10] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A Survey on In-context Learning. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. 1107–1128. doi:10.18653/v1/2024.emnlp-main.64

[11] Cody Dunne, Nathalie Henry Riche, Bongshin Lee, Ronald Metoyer, and George Robertson. 2012. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1663–1672. doi:10.1145/2207676.2208293

[12] Niklas Elmqvist and Jean-Daniel Fekete. 2010. Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics* 16, 3 (2010), 439–454. doi:10.1109/TVCG.2009.84

[13] Will Epperson, Gagan Bansal, Victor Dibia, Adam Fourney, Jack Gerrits, Erkang Zhu, and Saleema Amershi. 2025. Interactive Debugging and Steering of Multi-Agent AI Systems. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1–15. doi:10.1145/3706598.3713581

[14] Haishuo Fang, Xiaodan Zhu, and Iryna Gurevych. 2025. Preemptive Detection and Correction of Misaligned Actions in LLM Agents. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. 222–244. doi:10.18653/v1/2025.emnlp-main.12

[15] Google DeepMind. 2025. Gemini 3 Flash Preview. Website. https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview.

[16] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. 2016. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum* 35, 3 (2016), 491–500. doi:10.1111/cgf.12925

[17] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). North-Holland, 139–183. doi:10.1016/S0166-4115(08)62386-9

[18] Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Li Zhang, Lingyao Zhang, Min Yang, Mingchen Zhuge, Taicheng Guo, Tuo Zhou, Wei Tao, Robert Tang, Xiangtao Lu, Xiawu Zheng, Xinbing Liang, Yaying Fei, Yuheng Cheng, Yongxin Ni, Zhibin Gou, Zongze Xu, Yuyu Luo, and Chenglin Wu. 2025. Data Interpreter: An LLM Agent for Data Science. In *Findings of the Association for Computational Linguistics: ACL*. 19796–19821. doi:10.18653/v1/2025.findings-acl.1016

[19] Jessica Hullman, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar. 2013. A Deeper Understanding of Sequence in Narrative Visualization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2406–2415. doi:10.1109/TVCG.2013.119

[20] Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024. DataNarrative: Automated Data-Driven Storytelling with Visualizations and Texts. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. 19253–19286. doi:10.18653/v1/2024.emnlp-main.1073

[21] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. *arXiv preprint arXiv:2207.05221* (2022). https://arxiv.org/abs/2207.05221

[22] Eser Kandogan. 2001. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*. 107–116. doi:10.1145/502512.502530

[23] Majeed Kazemitabaar, Jack Williams, Ian Drosos, Tovi Grossman, Austin Zachary Henley, Carina Negreanu, and Advait Sarkar. 2024. Improving Steering and Verification in AI-Assisted Data Analysis with Interactive Task Decomposition. In *Proceedings of Annual ACM Symposium on User Interface Software and Technology*. 1–19. doi:10.1145/3654777.3676345

[24] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed Prompting: A Modular Approach for Solving Complex Tasks. In *Proceedings of International Conference on Learning Representations*. https://openreview.net/forum?id=_nGgzQjzaRy

[25] Klaus Krippendorff. 2018. In *Content Analysis: An Introduction to Its Methodology* (4th ed.). SAGE Publications, Chapter 7. doi:10.4135/9781071878781

[26] LangChain AI. 2025. LangChain. Website. https://www.langchain.com/.

[27] LangChain AI. 2025. LangGraph: Agent Orchestration Framework for Reliable AI Agents. Website. https://www.langchain.com/langgraph.

[28] Guozheng Li, Runfei Li, Yunshan Feng, Yu Zhang, Yuyu Luo, and Chi Harold Liu. 2024. CoInsight: Visual Storytelling for Hierarchical Tables With Connected Insights. *IEEE Transactions on Visualization and Computer Graphics* 30, 6 (2024), 3049–3061. doi:10.1109/TVCG.2024.3388553

[29] Haotian Li, Yun Wang, and Huamin Qu. 2024. Where Are We So Far? Understanding Data Storytelling Tools from the Perspective of Human-AI Collaboration. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. doi:10.1145/3613904.3642726

[30] Haotian Li, Yun Wang, and Huamin Qu. 2025. Reflection on Data Storytelling Tools in the Generative AI Era from the Human-AI Collaboration Perspective. In *Proceedings of the IEEE Visualization and Visual Analytics*. 21–25. doi:10.1109/VIS60296.2025.00009

[31] Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, Yiliu Tang, and Yun Huang. 2024. How AI Processing Delays Foster Creativity: Exploring Research Question Co-Creation with an LLM-based Agent. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1–25. doi:10.1145/3613904.3642698

[32] Jock Mackinlay. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics* 5, 2 (1986), 110–141. doi:10.1145/22949.22950

[33] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2024. DirectGPT: A Direct Manipulation Interface to Interact with Large Language Models. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1–16. doi:10.1145/3613904.3642462

[34] Andreas Mathisen, Tom Horak, Clemens N. Klokmose, Kaj Grønbæk, and Niklas Elmqvist. 2019. InsideInsights: Integrating Data-Driven Reporting in Collaborative Visual Analytics. *Computer Graphics Forum* 38, 3 (2019), 649–661. doi:10.1111/cgf.13717

[35] Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of Python in Science Conference*. 51–56. doi:10.25080/Majora-92bf1922-00a

[36] Meta. 2026. Manus: Hands On AI. Website. https://manus.im/.

[37] Microsoft. 2025. TypeScript: JavaScript With Syntax For Types. Website. https://www.typescriptlang.org/.

[38] Tamara Munzner. 2014. *Visualization Analysis and Design*. A K Peters/CRC Press. doi:10.1201/b17511

[39] Sergio Nefedov. 2025. Macro Indicators, 40 Years, 30 Countries. Kaggle. https://www.kaggle.com/datasets/sergionefedov/macro-indicators-40-years-30-countries.

[40] Phong H. Nguyen, Kai Xu, Andy Bardill, Betul Salman, Kate Herd, and B. L. William Wong. 2016. SenseMap: Supporting Browser-based Online Sensemaking through Analytic Provenance. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*. 91–100. doi:10.1109/VAST.2016.7883515

[41] Humphrey O. Obie, Caslon Chua, Iman Avazpour, Mohamed Abdelrazek, John Grundy, and Tomasz Bednarz. 2020. Authoring Logically Sequenced Visual Data Stories with Gravity. *Journal of Computer Languages* 58 (2020), 100961. doi:10.1016/j.cola.2020.100961

[42] Humphrey O. Obie, Dac Thanh Chuong Ho, Iman Avazpour, John Grundy, Mohamed Abdelrazek, Tomasz Bednarz, and Caslon Chua. 2022. Gravity++: A Graph-based Framework for Constructing Interactive Visualization Narratives. *Journal of Computer Languages* 71 (2022), 101125. doi:10.1016/j.cola.2022.101125

[43] Tianyue Ou, Wanyao Guo, Apurva Gandhi, Graham Neubig, and Xiang Yue. 2025. AgentDiagnose: An Open Toolkit for Diagnosing LLM Agent Trajectories. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 207–215. doi:10.18653/v1/2025.emnlp-demos.15

[44] Joon Sung Park, Rick Barber, Alex Kirlik, and Karrie Karahalios. 2019. A Slow Algorithm Improves Users' Assessments of the Algorithm's Accuracy. In *Proceedings of ACM Conference on Computer Supported Cooperative Work*. 1–15. doi:10.1145/3359204

[45] Peter Pirolli and Stuart Card. 2005. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of the International Conference on Intelligence Analysis*, Vol. 5. 2–4.

[46] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2024. ADaPT: As-Needed Decomposition and Planning with Language Models. In *Findings of the Association for Computational Linguistics: NAACL*. 4226–4252. doi:10.18653/v1/2024.findings-naacl.264

[47] Sebastián Ramírez. 2025. FastAPI. Website. https://fastapi.tiangolo.com/.

[48] Purvi Saraiya, Chris North, and Karen Duca. 2005. An Insight-Based Methodology for Evaluating Bioinformatics Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 11, 4 (2005), 443–456. doi:10.1109/TVCG.2005.53

[49] Purvi Saraiya, Chris North, Vy Lam, and Karen A. Duca. 2006. An Insight-Based Longitudinal Study of Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 12, 6 (2006), 1511–1522. doi:10.1109/TVCG.2006.85

[50] Arvind Satyanarayan and Jeffrey Heer. 2014. Authoring Narrative Visualizations with Ellipsis. *Computer Graphics Forum* 33, 3 (2014), 361–370. doi:10.1111/cgf.12392

[51] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 341–350. doi:10.1109/TVCG.2016.2599030

[52] Mehar Shanali. 2024. Middle East Economy and Oil Prices (1990–2024). Kaggle. https://www.kaggle.com/datasets/meharshanali/middle-east-economy-and-oil-prices-19902024.

[53] Leixian Shen, Haotian Li, Yun Wang, and Huamin Qu. 2024. From Data to Story: Towards Automatic Animated Data Video Creation with LLM-Based Multi-Agent Systems. In *Proceedings of IEEE VIS Workshop on Data Storytelling in an Era of Generative AI*. 20–27. doi:10.1109/GEN4DS63889.2024.00008

[54] Leixian Shen, Yizhi Zhang, Haidong Zhang, and Yun Wang. 2024. Data Player: Automatic Generation of Data Videos with Narration-Animation Interplay. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2024), 109–119. doi:10.1109/TVCG.2023.3327197

[55] Danqing Shi, Xinyue Xu, Fuling Sun, Yang Shi, and Nan Cao. 2021. Calliope: Automatic Visual Data Story Generation from a Spreadsheet. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 453–463. doi:10.1109/TVCG.2020.3030403

[56] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 36. https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html

[57] Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and Error: Exploration-Based Trajectory Optimization of LLM Agents. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7584–7600. doi:10.18653/v1/2024.acl-long.409

[58] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling Multilevel Exploration and Sensemaking with Large Language Models. In *Proceedings of ACM Symposium on User Interface Software and Technology*. 1–18. doi:10.1145/3586183.3606756

[59] Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. AdaPlanner: Adaptive Planning from Feedback with Language Models. In *Advances in Neural Information Processing Systems*, Vol. 36. https://proceedings.neurips.cc/paper_files/paper/2023/hash/b5c8c1c117618267944b2617add0a766-Abstract-Conference.html

[60] Simeng Sun, Yang Liu, Shuohang Wang, Dan Iter, Chenguang Zhu, and Mohit Iyyer. 2024. PEARL: Prompting Large Language Models to Plan and Execute Actions Over Long Documents. In *Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 469–486. doi:10.18653/v1/2024.eacl-long.29

[61] Priyan Vaithilingam, Elena L. Glassman, Jeevana Priya Inala, and Chenglong Wang. 2024. DynaVis: Dynamically Synthesized UI Widgets for Visualization Editing. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1–17. doi:10.1145/3613904.3642639

[62] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17, 3 (2020), 261–272. doi:10.1038/s41592-019-0686-2

[63] Vue.js Team. 2025. Vue.js - The Progressive JavaScript Framework. Website. https://vuejs.org/.

[64] Huichen Will Wang, Larry Birnbaum, and Vidya Setlur. 2025. Jupybara: Operationalizing a Design Space for Actionable Data Analysis and Storytelling with LLMs. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1–24. doi:10.1145/3706598.3713913

[65] Liangwei Wang, Zhan Wang, Shishi Xiao, Le Liu, Fugee Tsung, and Wei Zeng. 2025. VizTA: Enhancing Comprehension of Distributional Visualization with Visual-Lexical Fused Conversational Interface. *Computer Graphics Forum* 44, 7 (2025), e70110. doi:10.1111/cgf.70110

[66] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2609–2634. doi:10.18653/v1/2023.acl-long.147

[67] Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024. Executable Code Actions Elicit Better LLM Agents. In *Proceedings of International Conference on Machine Learning*. 50208–50232. https://proceedings.mlr.press/v235/wang24h.html

[68] Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. 2025. OpenHands: An Open Platform for AI Software Developers as Generalist Agents. In *Proceedings of International Conference on Learning Representations*. https://openreview.net/forum?id=OJd3ayDDoF

[69] Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma, and Dongmei Zhang. 2020. DataShot: Automatic Generation of Fact Sheets from Tabular Data. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 895–905. doi:10.1109/TVCG.2019.2934398

[70] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Proceedings of Conference on Neural Information Processing Systems*. 1800–1813. https://openreview.net/pdf?id=_VjQlMeSB_J

[71] Luoxuan Weng, Xingbo Wang, Junyu Lu, Yingchaojie Feng, Yihan Liu, Haozhe Feng, Danqing Huang, and Wei Chen. 2025. InsightLens: Augmenting LLM-Powered Data Analysis with Interactive Insight Management and Navigation. *IEEE Transactions on Visualization and Computer Graphics* 31, 6 (2025), 3719–3732. doi:10.1109/TVCG.2025.3567131

[72] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, Qi Zhang, and Tao Gui. 2025. The Rise and Potential of Large Language Model Based Agents: A Survey. *Science China Information Sciences* 68, 2 (2025), 121101. doi:10.1007/s11432-024-4222-0

[73] Liwenhan Xie, Chengbo Zheng, Haijun Xia, Huamin Qu, and Chen Zhu-Tian. 2024. WaitGPT: Monitoring and Steering Conversational LLM Agent in Data Analysis with On-the-Fly Code Visualization. In *Proceedings of Annual ACM Symposium on User Interface Software and Technology*. 1–14. doi:10.1145/3654777.3676374

[74] Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. 2025. AgentTrek: Agent Trajectory Synthesis via Guiding Replay with Web Tutorials. In *Proceedings of International Conference on Learning Representations*. https://openreview.net/forum?id=EEgYUccwsV

[75] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=WE_vluYUL-X

[76] Ryan Yen, Jiawen Stefanie Zhu, Sangho Suh, Haijun Xia, and Jian Zhao. 2024. CoLadder: Manipulating Code Generation via Multi-Level Blocks. In *Proceedings of Annual ACM Symposium on User Interface Software and Technology*. 1–20. doi:10.1145/3654777.3676357

[77] Ji Soo Yi, Rachel Melton, John Stasko, and Julie A. Jacko. 2005. Dust & Magnet: Multivariate Information Visualization Using a Magnet Metaphor. *Information Visualization* 4, 4 (2005), 239–256. doi:10.1057/palgrave.ivs.9500099

[78] Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2024. AgentTuning: Enabling Generalized Agent Abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL*. 3053–3077. doi:10.18653/v1/2024.findings-acl.181

[79] Wenshuo Zhang, Leixian Shen, Shuchang Xu, Jindu Wang, Jian Zhao, Huamin Qu, and Lin-Ping Yuan. 2025. NeuroSync: Intent-Aware Code-Based Problem Solving via Direct LLM Understanding Modification. In *Proceedings of Annual*

ACM Symposium on User Interface Software and Technology. 1–19. doi:10.1145/3746059.3747668

[80] Yuheng Zhao, Junjie Wang, Linbin Xiang, Xiaowen Zhang, Zifei Guo, Cagatay Turkay, Yu Zhang, and Siming Chen. 2025. LightVA: Lightweight Visual Analytics with LLM Agent-Based Task Planning and Execution. *IEEE Transactions on Visualization and Computer Graphics* 31, 9 (2025), 6162–6177. doi:10.1109/TVCG.2024.3496112

[81] Yuheng Zhao, Yixing Zhang, Yu Zhang, Xinyi Zhao, Junjie Wang, Zekai Shao, Cagatay Turkay, and Siming Chen. 2025. LEVA: Using Large Language Models to Enhance Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 31, 3 (2025), 1830–1847. doi:10.1109/TVCG.2024.3368060

[82] C. Zheng, T. Gao, S. Guo, et al. 2025. A Design Space of Animating Data-Driven Transitions in Data Videos. *Journal of Visualization* 28 (2025), 819–836. doi:10.1007/s12650-025-01066-5

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009