DOI: 10.1111/cgf.70459

Eurographics Conference on Visualization (EuroVis) 2026
N. Andrienko, M. Chen, and B. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 3

SlideSAVR: Enabling Live Analysis during Data Presentations via Multimodal Sketching and Voice Input

C. Han[1] and M.M. Tanjim[2] and S. Guo[2] C. Dierk[2] and K.E. Isaacs[1] and J. Hoffswell[2]

[1]University of Utah, Salt Lake City, USA [2]Adobe Research, USA

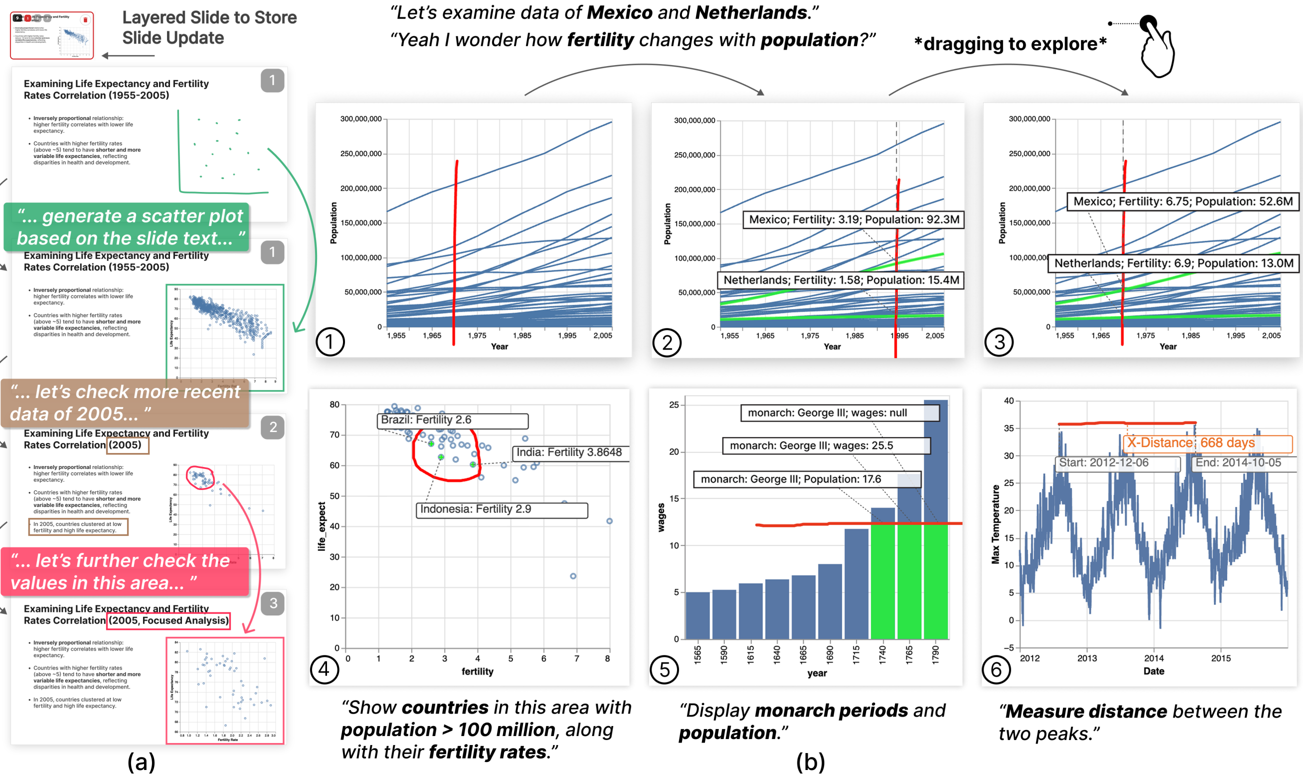

Figure 1: *SlideSAVR supports live data analysis, enabling on-the-fly Q&A between the presenter and audience. Presenters can (a) dynamically update charts and text in response to verbal prompts and sketch input; (b) create interactive handles to previously static charts.*

Abstract

Interpersonal communication in data science can yield sought-after insights, but presentation environments are often not conducive for live analysis, forcing the process to move offline. Through a formative survey with 16 participants, we identified both technical (e.g., complexity of tools) and psychological (e.g., pressure of programming during presentation) factors constraining live data analysis. To enable live analysis, we present SlideSAVR, a data-driven presentation assistant that leverages sketching and voice inputs in live discussion to support collaborative data analysis during presentations. Powered by an agentic framework that flexibly defines augmentation rules, updates slide content dynamically to match the live context, and automates backend computations, SlideSAVR enables fluid audience-presenter interaction and reduces the need for offline reanalysis and follow-up communication. We demonstrate SlideSAVR's ability to support a range of tasks through nine representative use cases. We further evaluate the system's accuracy and computation time across different settings, showing that SlideSAVR can reliably perform diverse tasks when provided with both sketch and voice inputs.

CCS Concepts

• ***Human-centered computing*** → ***Interactive systems and tools;*** *Interaction design; Visualization systems and tools;*

© 2026 The Author(s). Computer Graphics Forum published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

delivered by EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

1. Introduction

Presentation is an important part of data science work, occurring often throughout the analysis lifecycle [WAW*21] and providing an environment for data scientists and stakeholders to communicate progress and exchange insights [MLW*19, ZMW20]. Synchronous communication during data presentations enables quick exchanges and happens frequently in practice [BK21]. However, questions requiring further analysis are common and cannot be easily answered on-the-fly [PWNB22]. Analysts must often take the questions offline, perform new analyses, and then schedule another meeting, hindering the effectiveness of collaborative analysis during presentations and incurring extra time and resource costs.

To better understand the landscape of data analysis requirements and expectations during presentations, we conducted a formative survey with sixteen participants, probing their experiences with data presentations and their views on performing live data analysis while presenting. This survey revealed factors that act as barriers to live analysis during presentations; for example, answering audience questions can require accessing data and writing code, which may take too much time, break the flow of the presentation, or be difficult for audiences to follow. Some presenters also reported anxiety regarding coding or operating other tools in front of an audience.

Sketch-based interactions are promising to support live analysis in presentations. First, sketching is an intuitive and prevalent mode of communication [HI24, WHC15] and is already naturally used in presentations. Second, sketching is highly customizable, allowing it to address diverse analytical needs in live settings. However, the flexibility also makes sketch a "fuzzy" input that is challenging for computational processing. Prior work has addressed this challenge by relying on predefined shapes [LSR*15, LKS13] or by restricting sketches to a single function, such as data querying [TBJ15] or data binding in infographics [XHRC*18, SGH*25]. These strategies are not directly applicable to presentation scenarios that require multi-faceted support beyond a single function. Instead, combining sketches with voice input enhances support for live presentations: voice supplies semantic clarity and intent, while sketches contribute spatial grounding and visual specificity.

We present SlideSAVR, a slide presentation system that enables on-the-fly analysis (operated by a presenter and jointly shaped by audience questions) during data presentations through **S**ketching-**A**nd **V**oice-input **R**efinement. The system is built as a slide deck, with Multimodal LLM (MLLM) agents in the backend. All analytical operations are informed by sketches drawn on slides and/or voice inputs from both the presenter and audience. SlideSAVR leverages sketching as a user-friendly modality suited to presentation settings, and voice input as a natural part of the presentation discourse. SlideSAVR addresses diverse analytical tasks through intelligent routing of the multimodal inputs to specialized agents.

The multi-agent framework in SlideSAVR has four key agent modules: (i) the chart authoring and editing module; (ii) the sketch augmentation module that transforms sketches into interactive chart elements; (iii) the Q&A (search) module to answer questions using the provided context, as well as internet search and summary; and (iv) the statistical analysis module that computes data statistics and reports results. We design an API toolkit to give agents access to external resources and programmatic access to the sketches.

To illustrate the functional scope of SlideSAVR, we present a gallery of representative examples showing the system's ability to assist presenters in performing live data analysis operations. We also conduct a technical evaluation to assess SlideSAVR's efficacy, stress-testing it under various settings. The results show that the system can reliably perform a wide range of data-oriented actions when provided with both sketch and voice inputs.

2. Related Work

Our work aims to facilitate communication in collaborative data analysis by incorporating natural interaction modalities of speech and sketching. We discuss related work in two primary areas: (i) how communication and reporting occur in data science workflows, and (ii) how natural language and sketches have been used as an expressive medium for interaction and why they fit our task.

2.1. Presenting Data Science Results

Data science is a collaborative process, wherein data analysts are required to effectively communicate with various stakeholder groups [MLW*19, ZMW20]. A common mode of such communication is through presentations and reports [WLH19], which vary in formality, spanning executive briefings to casual brainstorming [BK21]. While interactivity tends to increase in more informal settings, formal environments tend to lack the same fluid feedback loops. Our work aims to bring the benefits of real-time feedback and exploration into more structured presentations.

Communication occurs **throughout the analysis lifecycle**, not only at the final reporting stage of a project [WAW*21]. Early on, analysts collaborate with stakeholders to gain domain knowledge and understand problem framing [KG21]. Throughout the project, analysts engage with stakeholders to present progress, build common ground, and identify misalignments [MWM*19]. In the final phase, analysts present results to facilitate stakeholders' decision-making and may iterate for new information [WLZ*21, WLH19].

Several studies have emphasized that data science is a highly iterative process. Wongsuphasawat et al. [WLH19] found that data acquisition, wrangling, modeling, and reporting are intertwined. Feedback from reporting or presentation may lead to adjustments in model settings or search for new data. Mao et al. [MWM*19] described this dynamic as constructing common ground for content (shared understanding of knowledge) and processes (shared understanding of procedures). Kross and Guo [KG21] proposed an "outer loop" involving stakeholder engagement, problem framing, trust-building, etc., in contrast to technical tasks in the inner loop.

However, existing presentation tools and workflows rarely accommodate the evolving nature of data analysis. A typical workflow incurs time and resource costs beyond analysis (i.e., arranging meetings), requiring back-and-forth exchanges [MLW*19] and repeated explanations [KG21]. Pang et al. [PWNB22] identified two challenges in synchronous communication: the pressure of peer programming and the difficulty of giving feedback on the fly. SlideSAVR aims to improve the communication aspect of such data science workflows. By enabling coding-free, on-the-fly analysis, it empowers presenters (analysts) to remain in the flow of presentation while responding to the audience's (stakeholders') questions,

© 2026 The Author(s).
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

reducing interpersonal communication costs and aligning more naturally with the iterative, conversational reality of data work.

Our work builds on growing interest in augmenting slide-based communication. Slide decks remain the primary way data science teams communicate [XWL*21, PPW*21] and thus prior tools have aided pre-hoc slide creation. NB2Slides [ZWWM22] uses neural networks to automatically generate slides from computational notebooks based on audience context, while Slide4N [WLL*23] focuses on human-AI collaboration, allowing manual content selection and post-editing. OutlineSpark [WLY*24] extends this idea to enable slide generation from user-written outlines. In contrast, SlideSAVR shifts the focus from slide generation to presentation.

SlideSAVR further draws on work from collaborative visualization [App91], which has been characterized along two dimensions: space (co-located vs. distributed) and time (asynchronous vs. synchronous) [IES*11, VW06]. Our work focuses on synchronous, co-located collaboration. Prior studies have explored similar settings using tabletop [TTP*06] and tiled displays [RJJ*06]. SlideSAVR also shares a lineage with prior work that leverages LLMs for visualization authoring and analysis [TCD*24, WTL23, WLD*25], employing natural language input to effectively convey user intents. We discuss relevant technical details of such work in Section 4.

In this paper, we use *communication* and *presentation* interchangeably to denote both interpersonal exchanges and the asymmetric presenter-audience exchange.

2.2. Sketching in Interactive Systems

Sketching is a ubiquitous practice that plays a key role in communication [Goe95], design [TSA*03], and ideation [TS09, WHC15]. Sketching is highly flexible both in what it can express and how it can be expressed. Walny et al. [WHC15] found that when sketching data, people use visuals along a continuum of numeracy to abstraction, ranging from countable marks to pictorial representations. The versatile nature of sketching makes it well-suited for conversational and interpretive settings such as live data presentations, which are fast-paced, requiring real-time responses using a wide spectrum of operations across different steps [KG21, WLH19].

The expressive capacity of sketches has inspired several sketch-based interaction systems, particularly in the domain of visualization authoring. NapkinVis [CMP10] uses pre-defined pen gestures to rapidly generate charts. DataInk [XHRC*18] uses sketches to create artistic glyphs and specify data bindings through direct manipulation. DataSelfie [KIHR*19] allows users to sketch custom visuals for their personal data. CompSketch [SGH*25] further treats sketches as objects to express design intentions. In contrast to these tools that emphasize creative authoring, SlideSAVR is optimized for speed, automation, and simplicity, adapting to the dynamic nature of data presentations by minimizing manual steps such as data selection, binding, and layout adjustment.

Sketches have also been used in data exploration and query interfaces. SketchVis [BLC*11] enables operations like selection, filtering, and aggregation on bar charts and scatterplots, and was used in a Wizard-of-Oz study [WLJ*12] to investigate the interplay of pen and touch interactions. SketchSliders [TBJ15] explored sketching for more complex queries, using differently shaped sketches as sliders to control charts. SketchStory [LKS13] and SketchInsight [LSR*15] are sketch-based systems capable of authoring narrative visualizations and supporting data exploration through predefined gestures and touch interactions. Because sketches are inherently ambiguous [LKS13], earlier systems either focused on a single functionality [TBJ15, XHRC*18, OTC25] or relied on predefined gestures and touch interactions [LKS13, LSR*15]. In contrast, SlideSAVR employs an LLM-based routing system with additional cues, such as voice transcripts, that provide critical contextual signals to disambiguate sketches. This combination makes sketches versatile enough to support diverse scenarios while ensuring the presenter's intent is captured with greater accuracy.

Beyond visualization tools, sketching has also been explored for other interactive domains [XMVS23, SKW*20, KMF*22]. InkSight [LLY*24] pairs sketches with textual information to document chart insights. Han and Isaacs [HI24] create interactive meeting notes from deictic sketching gestures. DrawTalking [RKW*24] incorporates multimodal input to construct an animated world. Code Shaping [YZV24] employs sketches to communicate code editing intentions to LLMs. Another line of work [XMVS23, SKW*20, KMF*22] explores sketches in augmented reality, binding strokes to real-world objects and applying physics to them. SlideSAVR shares a conceptual lineage but differs in both the affordances and technical use of sketches. For example, SlideSAVR generates data insights from sketches, similar to InkSight, but also uses sketches as notations for chart generation and operationalizes them as an interaction medium for data.

Sketching can express abstract ideas and is often accompanied by **natural language**. For example, when teaching geometry, one might draw a line and explain it in mathematical language [HSF*]; or when discussing a map, one might mark an area and then explain their findings [HI24]. Natural language has been used both independently and in combination with other modalities. Prior work interweaves speech with pen/touch interaction [SLS21], decomposes user questions to guide exploration [GSG*24], and examines speech interaction in collaborative visual analytics [MLBGI25].

3. Formative Study on Data Presentations Q&A

We conducted a survey to better understand current data presentation practices and investigate the potential for live data analysis. The survey focused on the interaction between presenters and audiences, with particular attention to how data-related questions are handled—whether they require additional analysis, and whether such analysis typically takes place offline or with live exploration. The survey questions are included in the supplemental materials.

We distributed the survey in a large corporate environment via several Slack channels targeting varying domain expertise. Sixteen people completed the survey. Thirteen respondents identified their role as 'researcher,' two as 'designer,' and one as 'analyst.' The participants had an average of 8.7 years of experience with data-related work (median = 6 years, range: 3–35 years). Most participants were between 25–34 years old (n=13), with some in the 45–54 (n=2) and 55–64 (n=1) brackets. Gender distribution consisted of eight men, seven women, and one participant who preferred not to disclose.

© 2026 The Author(s).
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

3.1. Presentation Practices and Context

Participants were asked to self-identify as a presenter, audience member, or both. Based on their selection, we included questions about their experience and perspectives as presenters and/or audience members. Among the sixteen participants, thirteen identified as both and responded to the questions from both perspectives. Two participants responded only from the presenter perspective, and one responded only from the audience perspective.

To help participants scope the term "data presentation" more concretely, we first asked participants to reflect on the frequency and types of presentations they normally give or attend, based on established categorizations from prior work [BK21]. The majority of participants present weekly in a casual environment (n=11). More rehearsed presentations to peers and colleagues were reported with monthly (n=7) or quarterly (n=6) frequency, while formal presentations to executives or customers were reported less often, generally at quarterly (n=5) or yearly (n=3) intervals.

For creating presentations, PowerPoint was the main tool (n=12), followed by Keynote (n=6) and Jupyter Notebook (n=4), with smaller numbers citing a variety of other design tools. For data analysis, participants reported using Python (n=12), Excel (n=9), and Jupyter Notebook (n=8) most commonly.

3.2. Communication is Heterogeneous and Multi-faceted

We asked participants to share the questions they had raised or received during data presentations. Thirteen participants report that they at least *sometimes* see data-related questions. According to our respondents, such topics range from clarifications and contextual questions, to deeper analytical probes. For example, P8 noted that technical colleagues frequently drill into methodological details: *"People are always interested in the [...] specifics around methodology, sample size, time windows, etc."*. P7 explained that audiences commonly ask for context like *"Where is this data from? How many people were in the study?"*, essentially seeking background and credibility for the results. Such questions often require the presenter to provide additional context beyond what is on the slides. Basic interpretive questions are also common, such as *"What do the x/y axes mean?"* (P10) or *"How to interpret the data?"* (P1).

Participants also recounted analytical or deep-diving questions that extended beyond the presented content. For example, P2 recalled being asked for *"additional experiments and analysis to get more specific takeaways"*. In some cases, the presenter had already conducted such analyses, but had not anticipated the question for the presentation: *"I was asked about results from the statistical tests which I had done, but did not have in the slides or have access to right away"* (P5). P7 reported that answering certain questions required accessing data not included in the presentation. These cases highlight the need to both retrieve and recompute data on the spot.

The diversity of audience inquiries reported in our survey illustrates that synchronous communication around data is multi-faceted. This heterogeneity reinforces the need for flexible presentation approaches—presenters, and the tools that try to help them, should be prepared to handle different topics in the course of a single session, often with little predictability in what will be asked.

3.3. Live Analysis Fosters Effective Discussions

Despite the challenges above, participants overwhelmingly recognized the value of being able to conduct analysis on the fly during presentations. In the audience role, P8 wrote: *"It's always nice to get answers to questions as they come up [...] but I understand that's not always realistic with current tools"*. Being able to explore data in the moment also helps the presenter maintain momentum. P11 noted that if a question arises in a weekly meeting and can be answered on the spot, it avoids waiting *"a week until the next meeting"* to see the results, enabling *"faster and more meaningful discussion."* Similarly, P10 emphasized that it is better to address questions immediately when possible *"as the audience may just forget what they asked"* and lose the context.

Participants noted additional aspects of presentations that can contribute to a collaborative environment once such a dialogue is established. P14 wrote that listeners often introduce perspectives *"closer to their own lives,"* and believed that allowing on-the-fly exploration of such ideas would *"significantly increase audience engagement,"* turning *"crowd wisdom"* into actionable insights. From presenters' perspectives, this flexibility was seen as a confidence booster: *"If I can perform real-time data analysis and visualization, I would be more confident to handle Q&A sessions"* (P16).

We used several Likert-scale items to assess participants' preferences for live analysis during presentations. From the presenter perspective, participants generally agreed that the complexity or inflexibility of current tools limits their ability to perform data analysis during a presentation (mean = 3.73 on a 5-point scale). Presenters indicated that they would perform live analysis more frequently if more flexible tools were available (mean = 3.6). From the audience perspective, most participants preferred having their questions addressed live rather than being deferred offline (mean = 3.57). Participants also emphasized the importance of understanding the presenter's actions during live analysis (mean = 4.43).

3.4. Live Programming and Audience Comprehension

However, participants also reported stress and awkwardness due to the pressures of performing live analysis during presentations, an observation also reported by [PWNB22]. P11 described the *"embarrassment of coding in front of others"* and noted it made them reluctant to perform live analysis during a talk. Participants further expressed anxiety about making mistakes, writing code slowly, or simply the awkward silence while waiting for results.

The operation of current analysis tools can contribute to presentation anxieties. Traditional data analysis environments are not designed for quick pivoting in the middle of a presentation—they may involve writing code, running scripts, or manipulating complex interfaces that disrupt the flow. AI can speed certain parts up (like coding) but imposes additional psychological barriers: *"I typically use python [...] with AI tools it is much faster, but I don't want to open up ChatGPT in front of my mentor and have it code for me on the spot"* (P11). P2 similarly noticed that doing analysis on the fly is difficult because explorative analyses involve multiple steps and *"heavily depend on intermediate results [...] it takes time to aggregate the results."* In short, current workflows often require data

© 2026 The Author(s).

wrangling and narrative preparation that cannot be rushed, leading presenters to push those tasks offline.

Participants also expressed concerns from their experience as audience members, noting that it can be difficult to comprehend what is happening during a live analysis session. If a presenter turns to raw tools or starts writing code without context, the audience may be left confused about the process or results. From the audience perspective, P11 wrote that *"it is easy to get lost during a live demo or live data analysis."* Participants also observe that taking complex queries offline is sometimes necessary because attempting them live would include the challenge of compiling the results into a *"presentable story"* for the audience (P3, P5).

Overall, the survey reveals a dual challenge: technical friction and social pressure make live analysis stressful for the presenter and potentially confusing for the audience. This observation suggests the need for efficient interactions that reduce the performance anxiety of presenters while keeping the audience oriented.

3.5. Key Takeaways from the Formative Survey

Our formative survey revealed three main findings: (i) audience members ask a wide variety of questions regarding the data and analysis, often requesting analysis beyond what was originally included in the presentation; (ii) participants agreed that being able to answer analysis questions live would accelerate their work and foster collaboration; and (iii) barriers to live analysis include anxiety regarding live-coding, delays while performing new analyses, and concerns about the audience's ability to follow along.

4. Developing SlideSAVR

To facilitate collaborative analysis in data presentations, we present SlideSAVR, a presentation system that enables presenters to use sketches and voice input to express high-level analytical intentions for live querying and visualization of data. SlideSAVR leverages an agentic framework to *identify* the intended goal and *perform* data-oriented actions to quickly update visualizations and slides.

4.1. Design Goals and Rationale

Drawing on our formative study and prior work, we describe three design goals for SlideSAVR and illustrate the underlying rationales.

A Versatile, All-in-One Presentation Workspace. Data-driven presentations are fast-paced and conversational, often involving heterogeneous and multi-faceted communication (Section 3.2), with questions ranging from clarifications to drill-down analysis. From an audience perspective, the majority of participants agreed on the importance of being able to follow along or understand the presenter's operations (Section 3.3). If a presenter dives into raw code or a complex UI that the audience is not familiar with, they can lose the narrative thread of the presentation. These concerns underscore the importance of a single tool that can handle many types of analysis while still presenting straightforward and accessible results. SlideSAVR provides an *all-in-one* workspace for presentations, enabling chart creation, data filtering, fact retrieval, and lightweight analysis *without* leaving the deck or disrupting the flow.

Code-Free, Audience-Friendly Disambiguation. Participants in our formative study described the *"embarrassment of coding in front of others"* (P11), which made them reluctant to perform live data analysis (Section 3.4). Sketching offers a coding-free, fluid medium that aligns with how presenters and audiences already communicate [Goe95]. However, this flexibility also introduces ambiguity that is hard to address via sketching alone. We thus designed SlideSAVR to use multimodal input to clarify analysis intentions. The combination of sketch and voice allows SlideSAVR to fluidly address diverse audience prompts, keeping attention on the story rather than the mechanisms behind-the-scenes.

In data-driven presentations, drawing a chart outline or circling an area on a slide are common actions, especially when paired with simple verbal expressions [B*05, Cla03, HH91]. For multimodal large language models (MLLMs), research shows that prompting with both sketches and textual prompts improves MLLM performance. Simple sketches, such as arrows or circles, can guide the model's attention [SRV23] and achieve state-of-the-art results on region-understanding tasks [C*24, ZGBFF16]. Sketches can also help LLMs solve complex reasoning tasks [HSF*]. Building on these findings, we use sketches and voice as the medium for presenters to communicate their intent to both the audience and system.

A System Controlled by the Presenter. Data-driven presentations are inherently asymmetric: the presenter leads the conversation, while the audience probes [KG21]. We thus designed SlideSAVR to assist the presenter specifically—not as a co-presenter or meeting-wide bot. By keeping all analytic operations on the presenter's side of the interface, we preserve this asymmetry and avoid confusing handoffs wherein attendees interact directly with an autonomous agent. This separation maintains presenter control, thereby preserving the status quo of asymmetric data presentation. While other forms of assistive techniques are possible, exploring them is beyond the scope of this paper and requires additional research.

4.2. SlideSAVR System Workflow

SlideSAVR consists of four components (as shown in Figure 2): (a) the *User Interface* (shown in Figure 3) captures the *Multimodal Input*, (b) *MLLM Agents*, (c) *API Tools*, and (d) *Contextual Information*. When triggered, the system uses the uploaded data, current slide context, and the sketches and transcribed voice input to form a multimodal prompt that is passed to an orchestrating MLLM agent. This *Orchestrator* infers the presenter's intent and routes to the appropriate agent to handle the individual actions. These specialized agents incorporate contextual information to infer the intended goal and perform data-oriented actions with the support of the API tools. We discuss these components and pipeline in more detail below.

4.2.1. Setup and Contextual Information in SlideSAVR

Presenters first upload slides (as Microsoft Powerpoint files), data, and any other documents they want the system to have in its context through the presentation materials pane Figure 3(*A*). Other documents might include notes from prior sessions or background material on the domain. In our prototype, all charts in the slides must be manually entered as Vega-Lite code [SMWH17]—a declarative language that can express most statistical charts. The system

© 2026 The Author(s).
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

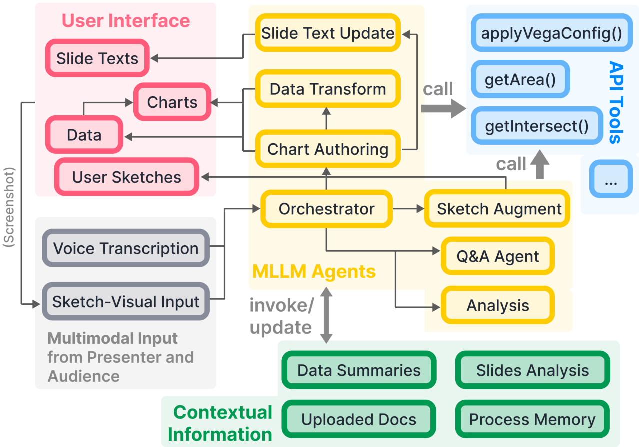

Figure 2: *The system architecture of SlideSAVR. It integrates multimodal inputs from the user interface with an MLLM-based agentic framework (seven agents), supported by customized API tools and contextual information that enable the agent to interpret the presentation context and tailor augmentation accordingly. A more detailed version is provided in the supplemental materials.*

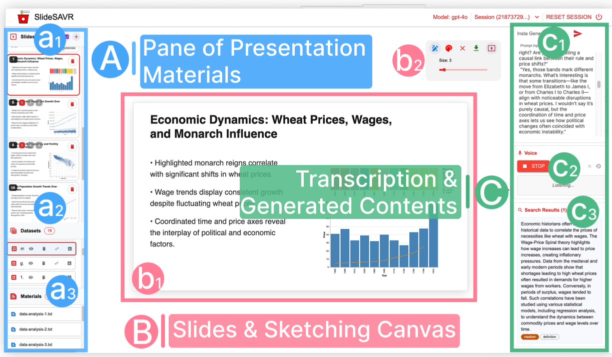

Figure 3: *The SlideSAVR user interface includes three panels. (A) The presentation materials pane includes slide thumbnails (a_1), datasets (a_2), and other relevant resources and memory files (a_3). (B) The main slide view and sketching canvas (b_1), which is the only part of the interface visible to the audience during the presentation; the sketching controls (b_2) allow the presenter to start sketching or clear the canvas. (C) The presenter pane displays the live transcription and generated content for review. A high resolution version is provided in the supplemental materials.*

organizes the materials into three panels: slide thumbnails (a_1), datasets (a_2), and documents (a_3). Users can access, add, or delete materials directly through these panels. The uploaded slides are used for the presentation content, while datasets are the basis for the system to conduct data-oriented actions later during the session. The relevant documents are used to provide context to the LLM.

After uploading the relevant documents to initialize the content, three new files are created to help maintain the context during interaction. Such external files are commonly used in LLM systems to provide long-term memory [XMG*25, JLF*23]. As shown in Figure 2, there are four types of contextual documents that help agents establish context for multimodal queries: (i) the "Uploaded Docs" mentioned in the previous paragraph; (ii) "Data Summaries" and (iii) "Slide Analysis" files that are automatically generated upon uploading the slides and data; and (iv) a "Process Memory" file that logs the presentation as it unfolds, including each agent invocation.

4.2.2. Multimodal Input as User Prompt

SlideSAVR uses both sketches and voice transcripts during presentations. The main view (b_1) in the user interface (Figure 3) serves as both the slide display and sketching canvas. A side pane (b_2) allows the presenter to adjust pen settings, erase sketches, download slides, or open a presentation view. The presentation view is shown to the audience; only the presenter can access the whole interface.

The voice transcription module (c_2) captures both presenter and audience utterances. Transcripts appear in real time in an editable textbox (c_1). The model summarizes and stores older sentences in the process memory ($t - t_{current} > 1$ mins); more recent sentences ($t - t_{current} \leq 1$ mins) are passed directly as prompts when the presenter triggers the pipeline. The presenter may make lightweight edits, such as trimming irrelevant text or adding key information.

The presenter can manually trigger the agent pipeline by pressing the "generate" button (c_1). The presenter may also configure SlideSAVR to auto-trigger two seconds after a drawing event. When triggered, the system captures a screenshot of the slide (b_1) including both the sketches and visual state of the slide as prompts. SlideSAVR also supports conventional pointing interactions as input. When the Control key is pressed, the sketches can be dragged to interactively explore the chart (Section 4.2.5).

4.2.3. MLLM Agents, Routing, and API Toolkit

A key component of SlideSAVR are the seven MLLM agents. Multimodal input first passes through the *Orchestrator*, whose role is to interpret the input, determine the appropriate course of action, and route to specialized agents responsible for distinct functionalities.

Guided by our formative survey (Section 3) and prior work, we designed four agent modules: (i) a visualization authoring and editing module that creates visualizations to support communication; (ii) a sketch augmentation module that turns sketches into interactive instruments; (iii) a Q&A agent that directly answers questions using contextual information and online search; and (iv) an analysis agent that generates Python code to conduct statistical analyses. We describe these modules in detail in the subsequent sections.

The orchestrator uses chain-of-thought prompting [WWS*22] to decide whether the user intends a chart edit or an information request; if the latter, it further classifies the request as sketch augmentation, statistical analysis, or simple Q&A without computation. The orchestrator must select exactly one module and is primed with the three sketch-augmentation types from Section 4.2.5.

The agents have access to a shared library of API tools (Figure 4) we designed to aid them in accomplishing their tasks. The API tools support interaction with data, visualizations, and contextual resources. The calls fall into two broad categories: sketch-related

© 2026 The Author(s).
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

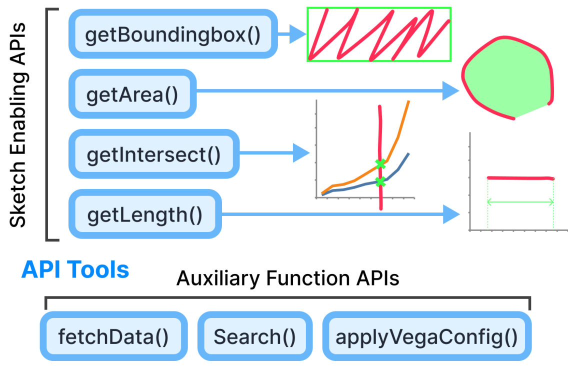

Figure 4: *Illustration of API tool categorization. (i) Sketch-enabling APIs, which support interaction on the canvas, and (ii) auxiliary function APIs, which help agents access external resources. The mechanism of sketch-enabling APIs is illustrated with thick strokes representing sketches, and thin strokes or colored areas indicating the captured data points or regions.*

tools, which interpret spatial relationships between sketches and chart elements (e.g., bounding boxes, intersections, lasso regions, etc.); and auxiliary function APIs, which access resources such as the dataset, web search, or chart specifications. A complete list of API functions is included in the supplemental materials.

4.2.4. Chart Authoring and Editing Module

In SlideSAVR, chart authoring must keep pace with live discussion and data exploration. Thus, instead of iterative editing or mixed-initiative approaches [TCD*24, WTL23, WLD*25], we prioritize generating imperfect, but readable charts in a one-shot manner. Figure 1(a) shows two scenarios triggering the chart authoring module: a presenter sketches a rough scatter plot to generate a new chart; the presenter circles an area to focus on and the chart is regenerated.

The chart generation module uses Vega-Lite [SMWH17]. Before producing code, the agent examines whether data transformation is needed based on the dataset snippet associated with the chart; if so, it requests the data transformation agent to generate and execute Python code. Following Data Formulator [WLD*25], we generate code to transform the data instead of having the agent transform it directly, which results in a quicker, more scalable pipeline.

We enforce design rules for generated charts through a Vega-Lite config file [Vegnd], encoding empirical guidelines such as avoiding color for overly large categorical sets, setting appropriate numbers of ticks, and specifying layer orders when multiple charts are present. These rules improve readability and maintain consistent quality. Although one-shot charts are not always perfect, this approach balances rapid, hands-free generation with visual clarity.

SlideSAVR also supports chart editing—modifications that do not change the chart type. For simplicity, the editing is handled by the same agent, with prompts to modify the given code. Editing often requires data transformation. For example, Figure 1(a)(2) shows filtering a scatterplot based on the vocal prompt of *"check more recent data of 2005"*. Figure 1(a)(3) shows a follow-up edit driven by both voice and sketch; in this case, the chart editing module calls the sketch-enabling APIs to locate the sketch region within the chart as input to the chart editing pipeline.

To organize and document chart edits, the slide thumbnails use a layered slide structure (Figure 3 (a_1)). In presentation, each time a chart is edited and the current slide is updated, the system generates a new layer attached to the current slide, as shown in Figure 1(a).

The chart authoring and editing agent module also has a separate slide text agent to update the text whenever data changes occur. This design ensures that the slide text matches the newly derived visual content, reducing the audience's cognitive load [MF14]. The text update agent is prompted to make minimal edits that only reflect the other changes to the slide. As shown in Figure 1(a)(2), the text is updated when the chart data changes. Note that the LLM can hallucinate [KNVZ25] when generating slide text, which is an inherent limitation of LLMs. Presenters may manually edit the text if this occurs, although doing so may interrupt the flow of the presentation. We discuss this hallucination issue further in Section 6.

4.2.5. Interactions via the Sketch Augmentation Module

Another core feature is sketch augmentation—using sketches as interactive elements, such as lassos and brushes, to flexibly query and display data on top of charts, as shown in Figure 1(b). The sketch augmentation agent is prompted to perform two main tasks: first, to determine the type of sketch augmentation, choosing from one of the following tool calls: `getArea()`, `getIntersect()`, `getLength()`, or `getBoundingBox()`; second, to specify rules for data filtering and display by generating a structured array.

The sketch-enabling API tools empower the agent to *command* the sketch's behavior. The agent first selects the appropriate API tool based on the multimodal input prompt, allowing the sketch to extract different types of information from the visual data. For instance, in Figure 1(b) ⑤ and ⑥, the same sketched horizontal line can serve multiple purposes: it functions either as an intersection tool to locate the intersection with data elements, or as a ruler to measure distances on the coordinate plane. The agent can also define what information should be displayed. In Figure 1(b) ④, for example, the agent filters for only the points in the selected region where population > 100 million, and displays the corresponding country name and fertility values as instructed by the voice prompt.

Finally, since sketches are used primarily for interactive data exploration, we also support basic user interactions. As demonstrated in Figure 1(b) ③, users can drag sketches to explore different parts of the chart. Similarly, the lasso selection sketch in Figure 1(b) ④ allows users to dynamically adjust the selected area by dragging, while the measurement tool in Figure 1(b) ⑥ enables users to reposition its starting and ending points to measure varying distances.

As an interactive exploration tool, sketching offers two main advantages: (i) Sketches are flexible and highly customizable. Presenters can control their location and shape to meet specific analytical needs that arise during collaborative exploration. (ii) Sketches are well-suited for on-the-fly creation. While such interactions could be achieved by other chart authoring approaches [SH24, ZWL*25], sketching provides a faster and lower-cost method during live presentations, especially to automate filtering and display rules.

© 2026 The Author(s).
Computer Graphics Forum published by Eurographics and John Wiley & Sons Ltd.

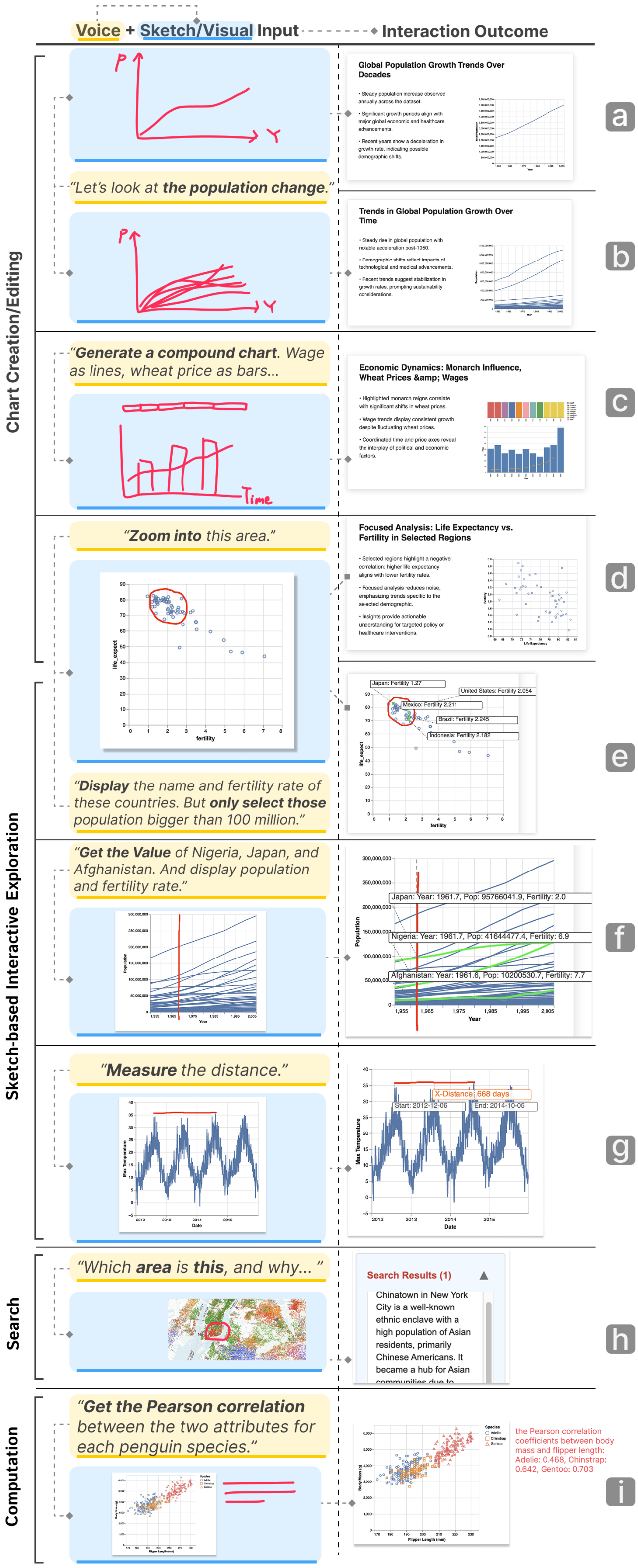

Figure 5: *Gallery of representative examples of SlideSAVR's system functionality. Notably, the same voice input can yield different outcomes when paired with different sketches (a and b), and the same sketch can support varied functions when combined with different voice inputs (d and e). See the supplemental material for an expanded figure with detailed agent operations and API calls.*

4.2.6. Q&A Support and Data Analysis Modules

Many questions during presentations stem from data but require external facts or statistical computations (e.g., averages, correlations). Using contextual information and the multimodal input, the Q&A agent can search online and summarize findings. Search results are presented in Figure 3 (c_3). Because the generated answers to knowledge-related questions may be unreliable, they are shown only to the presenter, who acts as a guardrail.

The data analysis agent performs statistical analysis by generating code, executing it in a sandbox, and summarizing the output. Results from the data analysis agent, however, can appear either in the side panel (Figure 3 (c_3)) or as text annotations directly on the slide, with sketches specifying placement. Sketching tool calls such as `getArea()` can similarly be handled by the Q&A and analysis agents, which may also call APIs to retrieve data.

We do not attempt to solve complex analytical problems requiring multi-step reasoning (e.g., building a predictive model to forecast sales), as they require careful consideration, exploration, and moderation. Unlike the simpler computations above, such analyses are not deterministic and can mislead if presented without vetting.

4.3. SlideSAVR Implementation

SlideSAVR is implemented as a web application, using React as the frontend framework and Python Flask as the backend. We use `python-pptx` [ppc] to parse the PowerPoint files. The voice recognition module is built on RealtimeSTT [con], a real-time speech-to-text library based on the Whisper model, which has approached human-level accuracy in transcribing audio [RKX*23]. For dataset parsing, we employ the implementation from Data Formulator [WTL23], which supports various formats such as CSV. We use an OpenAI API to access the language models. SlideSAVR supports touchscreens, mouse input, and stylus pens for sketching. Examples in this paper were created using a mouse cursor.

5. Evaluation of SlideSAVR

We first perform an *evaluation by demonstration* [LHV*18] showcasing SlideSAVR's functionality in a gallery of representative examples. A live demonstration case study is available in the supplemental video. We further conduct a technical evaluation to assess how well SlideSAVR can handle these examples, using varied settings to stress-test the system's capability.

5.1. Gallery of Representative Examples

Figure 5 shows the capabilities of the agents. a–d demonstrate chart creation and editing (Section 4.2.4); e–g show three sketch-based interactions (Section 4.2.5); h presents a Q&A example with online search (Section 4.2.6); and i illustrates an agent performing statistical analysis (Section 4.2.6). We provide an expanded version of the gallery and detailed analysis in the supplemental materials.

Through this gallery, we demonstrate SlideSAVR's ability to address many different types of live data analysis, illustrating that our tool meets several key design goals: the orchestrator can successfully route user requests to different functionality modules, enabling SlideSAVR to handle the heterogeneous and multi-faceted

© 2026 The Author(s).

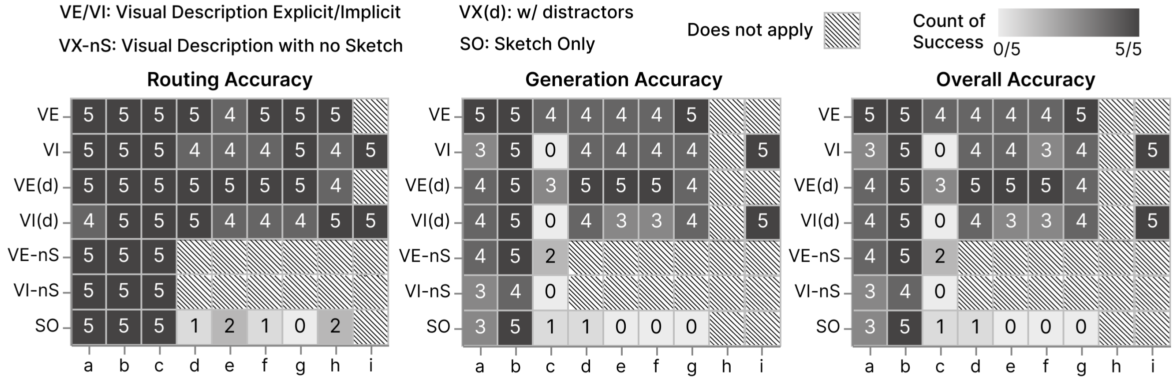

Figure 6: *Results from the technical evaluation. Each case was tested five times, with the number of successes shown in the heatmap. The rows denote seven variants formed by combinations of natural language expressions (explicit vs. implicit visual descriptions, with and without distractions) and sketch inputs (with and without sketches), and the columns correspond to the nine examples in Figure 5.*

communication that occurs during presentations. All results can be achieved using only verbal textual input and sketches, fulfilling our goal of a code-free, fast, and convenient interaction paradigm. Finally, each example can be generated within seconds (see section 5.2.3), supporting the fast-paced nature of presentations.

5.2. Technical Evaluation

We assess the efficacy of SlideSAVR's MLLM-based pipeline and our sketch+voice design to fulfill requests. Since LLM outputs can be inaccurate due to hallucination and other factors [JLF*23], we focus on SlideSAVR's accuracy in correctly routing requests (Section 4.2.3) and generating appropriate filtering and display rules for augmentation (Section 4.2.5). For generation accuracy, we manually verified that the generated slide text does not contradict the data or the query. We also measure the completion time of the system.

To our knowledge, no prior system directly addresses this presentation-focused scenario. While a general chatbot (e.g., ChatGPT) might be able to perform some of the same underlying tasks, it cannot support heterogeneous tasks effectively in a live presentation context. We include results from a simple chatbot baseline in the supplemental materials. Under the same conditions as our technical evaluation, it can produce visualizations—though with quality issues—but cannot identify or resolve data-query problems.

5.2.1. Test Cases

The evaluation covers all of the examples from Figure 5. For each example, we test two voice transcription variants (VE and VI), versions with verbal distractors VX(d), voice-only version with no sketches (VX-nS), and sketch-only (SO), resulting in seven tests cases per example. First, to assess whether MLLM agents rely on explicit visual descriptions, we compare inputs containing both data and visual details (VE: Visual-Explicit) with inputs containing only data descriptions (VI: Visual-Implicit). For example, *"Let's look at the population change."* is VI, whereas *"Draw a line chart with population on the Y and year on the X."* is VE. Second, to test the agent's ability to extract key information from casual conversation, we added textual distractors to both variants. Third, to evaluate the distinct roles of the two input modalities, we test the system using voice-only or sketch-only input. We run each test through the system five times. We exclude the two VE variants for Figure 5 i, as the query did not require visual information. We also exclude the non-sketch cases for d–h as sketch serves as an essential tool required to enable the action in these cases. In total, we ran 245 test cases $(((9 \times 7) - 14) \times 5)$. The complete list of voice inputs for each example case is provided in the supplemental materials.

5.2.2. Experiment Settings

We use GPT-4o as the base model for the evaluation. Tests are run on a MacBook Pro with M4 Pro Processor. We evaluate routing accuracy and generation accuracy separately, then combine them to obtain overall accuracy. For routing accuracy, we assess whether the orchestrator selected the correct module. For generation accuracy, we evaluate whether the outputs accurately displayed the required information. Misrouting can still show correct output information, (e.g., generating a new chart that shows the right data when it should have routed to sketch augmentation). Cases with minor style or formatting issues are considered successful if the information display was unaffected. Generation accuracy for h is excluded, as Q&A is flexible and difficult to classify as pass or fail.

5.2.3. Results

Figure 6 summarizes the results. **Routing accuracy** reaches 97.5% when the text includes both data and visual descriptions (VE). With only data descriptions (VI), accuracy drops to 91.1%, indicating that visual descriptions aid routing, though the system generally performs well with data descriptions plus sketch input. Most errors occur in the interaction augmentation examples (Figure 5 e–g), where requests are occasionally misinterpreted as chart generation. Distractors show no negative impact: routing accuracy remains 97.5% for VE(d) and 91.1% for VI(d), suggesting robustness to conversational distractors for routing. The routing accuracy for voice-only and sketch-only cases remains high for direct chart creation tasks (a–c), indicating the system is able to correctly direct this task type using either input modality. However, the routing accuracy for SO (sketch-only) cases drops significantly (24%) during sketch-based interactions (d–h). In these cases, the sketches

© 2026 The Author(s).

are simple-shaped and lack semantic meaning, which necessitates voice input to disambiguate user intentions.

Generation accuracy is 91.2% when the voice input includes both the data and visual descriptions (VE). With only data descriptions (VI), the accuracy drops significantly to 78.4%. The primary cause of this drop is seen in C, where the correspondence between the three charts in the composition and the encoded data cannot be conveyed without visual location descriptions. This example highlights the complementary roles of natural language and sketches: the former provides a general spatial framework (e.g., "the bars representing monarchies are on the top"), while the latter specifies details (e.g., showing how the monarchy bars are placed at the top). Although adding more detail to sketches could bridge this gap, drawing precise sketches is difficult in presentation environments, where most people present their data. Distractors slightly reduce the generation accuracy, with VE(d) at 88.6% and VI(d) at 77.8%,

The **overall accuracy**, combining both routing and generation, is: VE: 91.2% , VI: 78.4%, VE(d): 88.6%, VI(d): 77.8%, VE-nS: 73%, VI-nS: 46.7%, SO: 28.6%. These results suggest that the system performs best when textual input includes both data and visual descriptions, and with both voice and sketch inputs.

The **average completion time** is 8.1 seconds (range: 4.9–12.9 seconds). The main performance bottleneck is the LLM calls, with all other computation steps, combined, taking less than 200 milliseconds. Faster LLM API providers could substantially reduce latency. More advanced reasoning models may improve accuracy, though likely at the cost of longer response times.

6. Discussion, Limitations, and Future Work

We discuss the design implications of the system, the limitations of the approach, and potential directions for future work.

Authoring Visualizations with Sketches—Balancing Quality and Efficiency. In supporting SlideSAVR's goals of ease-of-use and presentation-paced response, we make trade offs in chart quality. Our constrained Vega-Lite configuration mitigates issues like excessive legends, but cannot guarantee the quality of every output. For instance, in Figure 5 C, the line stroke is too thin and the upper bar chart appears disproportionately tall relative to the sketch. Such issues reflect the limits of LLM agents in handling fine-grained visual design details. These challenges become more pronounced when visualizations extend beyond slide-based presentations—such as in storytelling contexts—where one-shot generation is often insufficient. Prior work such as ChartGPT [TCD*24] and Data Formulator [WTL23, WLD*25] has explored such iterative design problems. As an opportunity for future work, combining sketches with natural language could also support iterative refinement.

Human-AI Communication with Deictic Gestures. In linguistics, deixis refers to language where the meaning of certain expressions depends on the context of their use [Sta17], such as the phrase 'over there' in the sentence "Look at that apple tree over there." Deictic gestures are also widely used in interpersonal communications about data [HI24], e.g., *"this peak corresponds to weekday traffic."* The multimodal input in SlideSAVR can also be viewed as a form of deictic expression, where the sketch complements spoken natural language, acting as the context the agents require to understand the verbal expressions. This mode of communication is naturally understood by human audience members, and our system provides initial evidence that it can also be effectively interpreted by LLM-based AI agents. We envision that this form of interaction—combining sketches and voice—could be extended to other LLM-driven data systems, enabling users to express analytical intents in more intuitive and flexible ways.

Evaluation, Psychological Implications, and New Presentation Dynamics. Our evaluation focused on the system functionality, demonstrating the core capabilities and technical feasibility. The evaluation shows that SlideSAVR can effectively present analysis results without coding or complex manual operation—addressing one of the key psychological barriers for live analysis identified in prior work and our formative study (Section 3). However, the broader psychological effects remain unknown. The technical evaluation is confined to isolated components rather than the complete workflow. The extent to which the tool may reshape the evolving dynamics of data presentation is also still unknown. We have not yet evaluated user perceptions of LLM usage in such assistive system, for either presenters or audience members. While the lack of a user study limits our evaluation, we show that sketch-and-voice can enable live data presentation as a practical and compelling paradigm. A next step is an in-the-wild study to understand how it changes how people plan, structure, and deliver talks.

Another gap in the evaluation is the inherent diversity in natural language expressions and uncertainty of intent. Although we tested different types of verbal expression, the potential space is vast. Misinterpretation and underspecification are therefore difficult to avoid. Future work can more comprehensively assess and track how well LLMs can handle inputs across a spectrum of clarity, and how effectively users can phrase requests to reduce errors.

Limitations on APIs Capabilities and MLLM Hallucination. A limitation of the current implementation is its predefined functionality; for example, it supports only three types of sketch-based augmentations. SlideSAVR's architecture can accommodate additional augmentation and data exploration techniques as new modules. For instance, sketch-based sliders [TBJ15] for data exploration could be added by extending the API toolkit. Likewise, more complex analytical automations [MDW*23] could be encapsulated as API calls to expand the capabilities. However, a challenge is scaling the LLM's ability to reliably recognize a larger number of sketch types. Future work could investigate how many distinct tasks an LLM can support, and evaluate semantic distance between tasks.

Another limitation is the possibility of hallucination [PO25]. Introducing stronger guardrails (e.g., requiring presenters to review all content before it is shown to the audience) improves reliability but reduces automation, creating an inherent trade-off. Determining how best to balance this trade-off requires systematic user evaluation, which should be explored in future work.

Acknowledgments

We used ChatGPT and Gemini to polish grammar and language.

© 2026 The Author(s).

References

[App91] APPLEGATE L. M.: Technology support for cooperative work: A framework for studying introduction and assimilation in organizations. *Journal of Organizational Computing and Electronic Commerce, Taylor & Francis 1*, 1 (1991), 11–39. 3

[B*05] BRENNAN S. E., ET AL.: How conversation is shaped by visual and spoken evidence. *Approaches to studying world-situated language use: Bridging the language-as-product and language-as-action traditions* (2005), 95–129. 5

[BK21] BREHMER M., KOSARA R.: From jam session to recital: Synchronous communication and collaboration around data in organizations. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2021), 1139–1149. 2, 4

[BLC*11] BROWNE J., LEE B., CARPENDALE S., RICHE N., SHERWOOD T.: Data analysis on interactive whiteboards through sketch-based interaction. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2011), ITS '11, Association for Computing Machinery, p. 154–157. 3

[C*24] CAI M., ET AL.: Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *CVPR* (2024), pp. 12914–12923. 5

[Cla03] CLARK H. H.: Pointing and placing. pointing: Where language, culture and cognition meet, 2003. 5

[CMP10] CHAO W., MUNZNER T., PANNE M.: Poster: Rapid pen-centric authoring of improvisational visualizations with napkinvis. *Posters Compendium InfoVis 2* (01 2010). 3

[con] CONTRIBUTORS R.: Realtimestt: Real-time speech-to-text library with low-latency transcription. https://github.com/KoljaB/RealtimeSTT. Accessed: 2025-09-01. 8

[Goe95] GOEL V.: *Sketches of thought*. MIT press, 1995. 3, 5

[GSG*24] GUO Y., SHI D., GUO M., WU Y., CAO N., CHEN Q.: Talk2data: A natural language interface for exploratory visual analysis via question decomposition. *ACM Transactions on Interactive Intelligent Systems 14*, 2 (2024). doi:10.1145/3643894. 3

[HH91] HILL W. C., HOLLAN J. D.: Deixis and the future of visualization excellence. In *IEEE Vis* (1991), vol. 91, pp. 314–320. 5

[HI24] HAN C., ISAACS K. E.: A deixis-centered approach for documenting remote synchronous communication around data visualizations. *IEEE Trans. on Visualization and Computer Graphics* (2024). 2, 3, 10

[HSF*] HU Y., SHI W., FU X., ROTH D., OSTENDORF M., ZETTLEMOYER L., SMITH N. A., KRISHNA R.: Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*. 3, 5

[IES*11] ISENBERG P., ELMQVIST N., SCHOLTZ J., CERNEA D., MA K.-L., HAGEN H.: Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization* (2011), 310–326. 3

[JLF*23] JI Z., LEE N., FRIESKE R., YU T., SU D., XU Y., ISHII E., BANG Y. J., MADOTTO A., FUNG P.: Survey of hallucination in natural language generation. *ACM Comput. Surv. 55*, 12 (2023), 1–38. 6, 9

[KG21] KROSS S., GUO P.: Orienting, framing, bridging, magic, and counseling: How data scientists navigate the outer loop of client collaborations in industry and academia. *Proceedings of the ACM on Human-Computer Interaction 5*, CSCW2 (2021), 1–28. 2, 3, 5

[KIHR*19] KIM N. W., IM H., HENRY RICHE N., WANG A., GAJOS K., PFISTER H.: DataSelfie: Empowering people to design personalized visuals to represent their data. In *CHI '19* (2019), ACM, p. 1–12. 3

[KMF*22] KAIMOTO H., MONTEIRO K., FARIDAN M., LI J., FARAJIAN S., KAKEHI Y., NAKAGAKI K., SUZUKI R.: Sketched reality: Sketching bi-directional interactions between virtual and physical worlds with ar and actuated tangible ui. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2022), UIST '22, Association for Computing Machinery. doi:10.1145/3526113.3545626. 3

[KNVZ25] KALAI A. T., NACHUM O., VEMPALA S. S., ZHANG E.: Why language models hallucinate, 2025. URL: https://arxiv.org/abs/2509.04664, arXiv:2509.04664. 7

[LHV*18] LEDO D., HOUBEN S., VERMEULEN J., MARQUARDT N., OEHLBERG L., GREENBERG S.: Evaluation strategies for hci toolkit research. In *Proceedings of the 2018 CHI conference on human factors in computing systems* (2018), pp. 1–17. 8

[LKS13] LEE B., KAZI R. H., SMITH G.: Sketchstory: Telling more engaging stories with data through freeform sketching. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2416–2425. 2, 3

[LLY*24] LIN Y., LI H., YANG L., WU A., QU H.: Inksight: Leveraging sketch interaction for documenting chart findings in computational notebooks. *IEEE Transactions on Visualization and Computer Graphics 30*, 1 (Jan. 2024), 944–954. 3

[LSR*15] LEE B., SMITH G., RICHE N. H., KARLSON A., CARPENDALE S.: Sketchinsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. In *IEEE Pacific Visualization Symposium (PacificVis)* (2015), pp. 199–206. 2, 3

[MDW*23] MA P., DING R., WANG S., HAN S., ZHANG D.: Demonstration of insightpilot: An llm-empowered automated data exploration system. *arXiv preprint arXiv:2304.00477* (2023). 10

[MF14] MAYER R. E., FIORELLA L.: 12 principles for reducing extraneous processing in multimedia learning: Coherence, signaling, redundancy, spatial contiguity, and temporal contiguity principles. *The Cambridge handbook of multimedia learning 279* (2014), 279–315. 7

[MLBGI25] MOLINA LEÓN G., BEZERIANOS A., GLADIN O., ISENBERG P.: Talk to the wall: The role of speech interaction in collaborative visual analytics. *IEEE Trans. on Visualization and Computer Graphics 31*, 1 (2025), 941–951. 3

[MLW*19] MULLER M., LANGE I., WANG D., PIORKOWSKI D., TSAY J., LIAO Q. V., DUGAN C., ERICKSON T.: How data science workers work with data: Discovery, capture, curation, design, creation. In *Proc. CHI conference on human factors in computing systems* (2019), pp. 1–15. 2

[MWM*19] MAO Y., WANG D., MULLER M., VARSHNEY K. R., BALDINI I., DUGAN C., MOJSILOVIĆ A.: How data scientistswork together with domain experts in scientific collaborations: To find the right answer or to ask the right question? *Proceedings of the ACM on Human-Computer Interaction 3*, GROUP (2019), 1–23. 2

[OTC25] OFFENWANGER A., TSANDILAS T., CHEVALIER F.: Datagarden: Formalizing personal sketches into structured visualization templates. *IEEE Trans. on Visualization and Computer Graphics 31*, 1 (2025), 1268–1278. 3

[PO25] PANDEY S., OTTLEY A.: Benchmarking visual language models on standardized visualization literacy tests. In *Computer Graphics Forum* (2025), vol. 44, Wiley Online Library, p. e70137. 10

[ppc] PYTHON-PPTX CONTRIBUTORS: python-pptx: Python library for creating and updating powerpoint (.pptx) files. https://github.com/python-pptx/python-pptx. Accessed: 2025-09-01. 8

[PPW*21] PIORKOWSKI D., PARK S., WANG A. Y., WANG D., MULLER M., PORTNOY F.: How ai developers overcome communication challenges in a multidisciplinary team: A case study. *Proceedings of the ACM on human-computer interaction 5*, CSCW1 (2021), 1–25. 3

[PWNB22] PANG R. Y., WANG R., NELSON J., BATTLE L.: How do data science workers communicate intermediate results? In *IEEE Visualization in Data Science (VDS)* (2022), pp. 46–54. 2, 4

[RJJ*06] RENAMBOT L., JEONG B., JAGODIC R., JOHNSON A., LEIGH J., AGUILERA J.: Collaborative visualization using high-resolution tiled displays. In *ACM CHI Workshop on Information Visualization Interaction Techniques for Collaboration Across Multiple Displays* (2006), pp. 1–4. 3

[RKW*24] ROSENBERG K. T., KAZI R. H., WEI L.-Y., XIA H., PERLIN K.: Drawtalking: Building interactive worlds by sketching and speaking. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (2024), pp. 1–25. 3

© 2026 The Author(s).

[RKX*23] Radford A., Kim J. W., Xu T., Brockman G., McLeavey C., Sutskever I.: Robust speech recognition via large-scale weak supervision. In *International conference on machine learning* (2023), PMLR, pp. 28492–28518. 8

[SGH*25] Shi X., Guo S., Hoffswell J., Chan G. Y.-Y., Bursztyn V. S., Zhao J., Koh E.: Comprehensive sketching: Exploring infographic design alternatives in parallel. In *Proc. Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (2025), CHI EA '25, ACM. 2, 3

[SH24] Snyder L. S., Heer J.: DIVI: Dynamically interactive visualization. *IEEE Trans. on Visualization and Computer Graphics 30*, 1 (2024), 403–413. 7

[SKW*20] Suzuki R., Kazi R. H., Wei L.-Y., DiVerdi S., Li W., Leithinger D.: Realitysketch: Embedding responsive graphics and visualizations in ar with dynamic sketching. In *Adjunct Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2020), UIST '20 Adjunct, Association for Computing Machinery, p. 135–138. 3

[SLS21] Srinivasan A., Lee B., Stasko J.: Interweaving multimodal interaction with flexible unit visualizations for data exploration. *IEEE Trans. on Visualization and Computer Graphics 27*, 8 (2021), 3519–3533. 3

[SMWH17] Satyanarayan A., Moritz D., Wongsuphasawat K., Heer J.: Vega-lite: A grammar of interactive graphics. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (Jan. 2017), 341–350. 5, 7

[SRV23] Shtedritski A., Rupprecht C., Vedaldi A.: What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 11987–11997. 5

[Sta17] Stapleton A.: Deixis in modern linguistics. *Essex Student Journal 9*, 1 (2017). 10

[TBJ15] Tsandilas T., Bezerianos A., Jacob T.: Sketchsliders: Sketching widgets for visual exploration on wall displays. In *Proc. ACM Conference on Human Factors in Computing Systems* (2015), ACM, p. 3255–3264. 2, 3, 10

[TCD*24] Tian Y., Cui W., Deng D., Yi X., Yang Y., Zhang H., Wu Y.: ChartGPT: Leveraging LLMs to generate charts from abstract natural language. *IEEE Trans. on Visualization and Computer Graphics 31*, 3 (2024), 1731–1745. 3, 7, 10

[TS09] Tversky B., Suwa M.: *Thinking with sketches*. Oxford University Press, 2009. 3

[TSA*03] Tversky B., Suwa M., Agrawala M., Heiser J., Stolte C., Hanrahan P., Phan D., Klingner J., Daniel M.-P., Lee P., et al.: Sketches for design and design of sketches. In *Human Behaviour in Design: Individuals, Teams, Tools*. Springer, 2003, pp. 79–86. 3

[TTP*06] Tang A., Tory M., Po B., Neumann P., Carpendale S.: Collaborative coupling over tabletop displays. In *Proc. CHI conference on Human Factors in computing systems* (2006), pp. 1181–1190. 3

[Vegnd] Vega: Config. https://vega.github.io/vega/docs/config/, n.d. Accessed: 2025-09-07. URL: https://vega.github.io/vega/docs/config/. 7

[VW06] Viegas F., Wattenberg M.: Communication-minded visualization: A call to action. *IBM Systems Journal 45*, 4 (2006). 3

[WAW*21] Wang D., Andres J., Weisz J. D., Oduor E., Dugan C.: AutoDS: Towards human-centered automation of data science. In *Proc. CHI conference on human factors in computing systems* (2021), pp. 1–12. 2

[WHC15] Walny J., Huron S., Carpendale S.: An exploratory study of data sketching for visual representation. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 231–240. 2, 3

[WLD*25] Wang C., Lee B., Drucker S. M., Marshall D., Gao J.: Data formulator 2: Iterative creation of data visualizations, with ai transforming data along the way. In *Proc. CHI Conference on Human Factors in Computing Systems* (2025), pp. 1–17. 3, 7, 10

[WLH19] Wongsuphasawat K., Liu Y., Heer J.: Goals, process, and challenges of exploratory data analysis: An interview study. *arXiv preprint arXiv:1911.00568* (2019). 2, 3

[WLJ*12] Walny J., Lee B., Johns P., Henry Riche N., Carpendale S.: Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE Trans. on Visualization and Computer Graphics 18*, 12 (2012), 2779–2788. 3

[WLL*23] Wang F., Liu X., Liu O., Neshati A., Ma T., Zhu M., Zhao J.: Slide4n: Creating presentation slides from computational notebooks with human-ai collaboration. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2023), pp. 1–18. 3

[WLY*24] Wang F., Lin Y., Yang L., Li H., Gu M., Zhu M., Qu H.: Outlinespark: Igniting ai-powered presentation slides creation from computational notebooks through outlines. In *Proc. CHI Conference on Human Factors in Computing Systems* (2024), pp. 1–16. 3

[WLZ*21] Wang D., Liao Q. V., Zhang Y., Khurana U., Samulowitz H., Park S., Muller M., Amini L.: How much automation does a data scientist want? *arXiv preprint arXiv:2101.03970* (2021). 2

[WTL23] Wang C., Thompson J., Lee B.: Data formulator: Ai-powered concept-driven visualization authoring. *IEEE Trans. on Visualization and Computer Graphics 30*, 1 (2023), 1128–1138. 3, 7, 8, 10

[WWS*22] Wei J., Wang X., Schuurmans D., Bosma M., Xia F., Chi E., Le Q. V., Zhou D., et al.: Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems 35* (2022), 24824–24837. 6

[XHRC*18] Xia H., Henry Riche N., Chevalier F., De Araujo B., Wigdor D.: Dataink: Direct and creative data-oriented drawing. In *Proce. CHI Conference on Human Factors in Computing Systems* (2018), ACM, p. 1–13. 2, 3

[XMG*25] Xu W., Mei K., Gao H., Tan J., Liang Z., Zhang Y.: A-mem: Agentic memory for LLM agents. *arXiv preprint arXiv:2502.12110* (2025). 6

[XMVS23] Xia Z., Monteiro K., Van K., Suzuki R.: Realitycanvas: Augmented reality sketching for embedded and responsive scribble animation effects. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2023), UIST '23. 3

[XWL*21] Xin D., Wu E. Y., Lee D. J.-L., Salehi N., Parameswaran A.: Whither automl? understanding the role of automation in machine learning workflows. In *Proc. CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–16. 3

[YZV24] Yen R., Zhao J., Vogel D.: Code shaping: Iterative code editing with free-form sketching. In *Adjunct Proc. Symposium on User Interface Software and Technology* (2024), UIST Adjunct '24, ACM. 3

[ZGBFF16] Zhu Y., Groth O., Bernstein M., Fei-Fei L.: Visual7w: Grounded question answering in images. In *Proc. IEEE conference on computer vision and pattern recognition* (2016), pp. 4995–5004. 5

[ZMW20] Zhang A. X., Muller M., Wang D.: How do data science workers collaborate? roles, workflows, and tools. *Proc. ACM on Human-Computer Interaction 4*, CSCW1 (2020), 1–23. 2

[ZWL*25] Zhao Y., Wang Y., Luo X., Wang Y., Fekete J.-D.: Libra: An interaction model for data visualization. In *Proc. CHI Conference on Human Factors in Computing Systems* (2025), pp. 1–17. 7

[ZWWM22] Zheng C., Wang D., Wang A. Y., Ma X.: Telling stories from computational notebooks: Ai-assisted presentation slides creation for presenting data science work. In *Proc. CHI Conference on Human Factors in Computing Systems* (2022), pp. 1–20. 3

© 2026 The Author(s).