SketchGPT: A Sketch-based Multimodal Interface for Application-Agnostic LLM Interaction

Zeyuan Huang[*]
Institute of Software, Chinese Academy of Sciences
Beijing, China
zeyuan2020@iscas.ac.cn

Cangjun Gao[*]
Institute of Software, Chinese Academy of Sciences
Beijing, China
gaocangjun23@mails.ucas.ac.cn

Yaxian Shan
Institute of Software, Chinese Academy of Sciences
Beijing, China
iris210805@126.com

Haoxiang Hu[*]
Institute of Software, Chinese Academy of Sciences
Beijing, China
huhaoxiang22@mails.ucas.ac.cn

Qingkun Li
Institute of Software, Chinese Academy of Sciences
Beijing, China
liqingkun@iscas.ac.cn

Xiaoming Deng[*]
Institute of Software, Chinese Academy of Sciences
Beijing, China
xiaoming@iscas.ac.cn

Cuixia Ma[*][†]
Institute of Software, Chinese Academy of Sciences
Beijing, China
cuixia@iscas.ac.cn

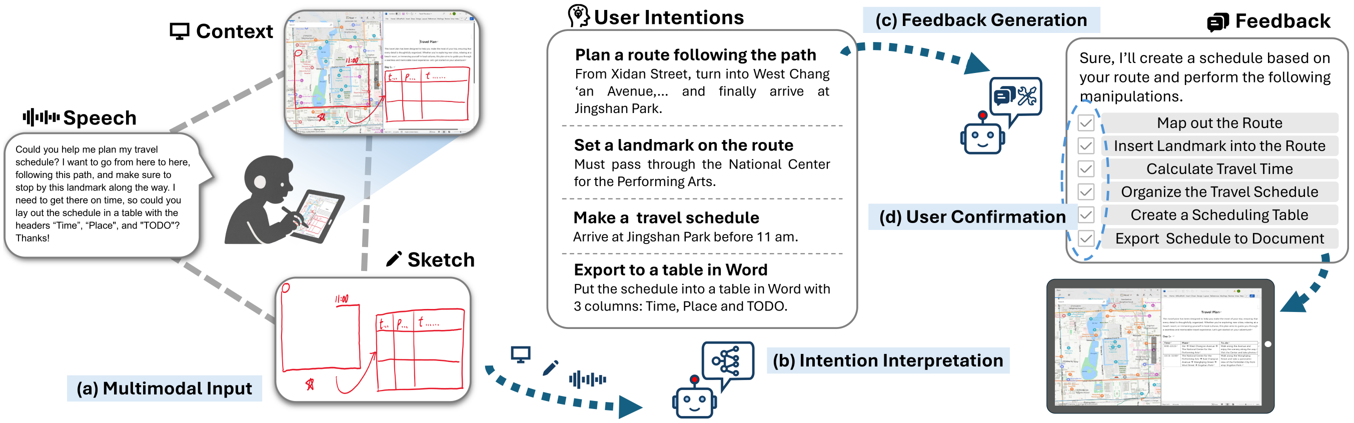
Yu-Kun Lai
School of Computer Science and Informatics
Cardiff University
Cardiff, Wales, United Kingdom
LaiY4@cardiff.ac.uk

Yong-Jin Liu
CS Department, MOE-Key Laboratory of Pervasive Computing
Tsinghua University
Beijing, China
liuyongjin@tsinghua.edu.cn

Feng Tian[‡]
Institute of software, Chinese Academy of Sciences
Beijing, China
tianfeng@iscas.ac.cn

Guozhong Dai
Institute of software, Chinese Academy of Sciences
Beijing, China
dgz@iscas.ac.cn

Hongan Wang[*]
Institute of Software, Chinese Academy of Sciences
Beijing, China
hongan@iscas.ac.cn



Could you help me plan my travel schedule? I want to go from here to here, following this path, and make sure to stop by this landmark along the way. I need to get there on time, so could you lay out the schedule in a table with the headers "Time", "Place", and "TODO"? Thanks!

Plan a route following the path
From Xidan Street, turn into West Chang 'an Avenue,... and finally arrive at Jingshan Park.

Set a landmark on the route
Must pass through the National Center for the Performing Arts.

Make a travel schedule
Arrive at Jingshan Park before 11 am.

Export to a table in Word
Put the schedule into a table in Word with 3 columns: Time, Place and TODO.

Sure, I'll create a schedule based on your route and perform the following manipulations.

- ☑ Map out the Route
- ☑ Insert Landmark into the Route
- ☑ Calculate Travel Time
- ☑ Organize the Travel Schedule
- ☑ Create a Scheduling Table
- ☑ Export Schedule to Document

Figure 1: SketchGPT introduces a multimodal interaction that enables communication with large language models through sketches and speech. It effectively interprets users' open-ended and abstract interaction intentions within context, supporting direct and unrestricted expression at the system level.

[*] Zeyuan Huang, Cangjun Gao, Haoxiang Hu, Xiaoming Deng, Cuixia Ma, and Hongan Wang are also with the School of Computer Science and Technology, University of Chinese Academy of Sciences.
[†] Corresponding author.
[‡] Feng Tian is also with the School of Artifcial Intelligence, University of Chinese Academy of Sciences.

UIST '25, Busan, Republic of Korea
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2037-6/25/09
https://doi.org/10.1145/3746059.3747598

Abstract

Human interaction with large language models (LLMs) is typically confined to text or image interfaces. Sketches offer a powerful medium for articulating creative ideas and user intentions, yet their potential remains underexplored. We propose SketchGPT, a novel interaction paradigm that integrates sketch and speech input directly over the system interface, facilitating open-ended, context-aware communication with LLMs. By leveraging the complementary strengths of multimodal inputs, expressions are enriched with semantic scope while maintaining efficiency. Interpreting user intentions across diverse contexts and modalities remains a key challenge. To address this, we developed a prototype based on a multi-agent framework that infers user intentions within context and generates executable context-sensitive and toolkit-aware feedback. Using Chain-of-Thought techniques for temporal and semantic alignment, the system understands multimodal intentions and performs operations following human-in-the-loop confirmation to ensure reliability. User studies demonstrate that SketchGPT significantly outperforms unimodal manipulation approaches, offering more intuitive and effective means to interact with LLMs.

CCS Concepts

• **Human-centered computing → Interaction techniques**; **Interactive systems and tools**.

Keywords

sketch input, multimodal interaction, large language models

ACM Reference Format:
Zeyuan Huang, Cangjun Gao, Yaxian Shan, Haoxiang Hu, Qingkun Li, Xiaoming Deng, Cuixia Ma, Yu-Kun Lai, Yong-Jin Liu, Feng Tian, Guozhong Dai, and Hongan Wang. 2025. SketchGPT: A Sketch-based Multimodal Interface for Application-Agnostic LLM Interaction. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3746059.3747598

1 Introduction

Large language models (LLMs) are increasingly integrated into various tasks, from recording [11, 47] and reading [30] to healthcare [32, 71, 74], voice processing [54], data analysis [87, 88, 97], etc. As these models evolve into capable agents handling complex challenges [104], more general AI agents have been emerging recently [3, 58, 90]. Text prompting serves as the major "language" for human-LLM interaction, which provides an indirect means of conveying intentions. Achieving alignment between human intent and machine understanding requires precise contextual articulation [10, 55]—a process demanding both deliberate effort [76] and expertise [109]. Consequently, when expressing intentions and context proves difficult, users tend to favor more direct interactions [34].

Recent research has explored more seamless interactions with LLMs. To support natural intention expression, non-verbal cues such as eye gaze [11, 49], gestures [111], touch [26], and direct interface manipulation [59] have been employed to leverage contextual cues for target reference, reducing ambiguity and streamlining communication. Moreover, direct interactions within the GUI are constrained by the available interface elements to express intent. These modalities are limited in conveying richer and more abstract semantics such as spatial, relational, symbolic, and graphical details, which are also difficult to articulate clearly through text.

Sketching is regarded as an accessible and efficient way to express ideas [12, 96]. However, previous work has confined the use of sketches to specific applications or sketch vocabularies, such as handwritten notes, drawing, pen gestures, and handwriting recognition, without integrating them into broader system interfaces. Understanding sketches and their context was previously limited, but the advent of multimodal large language models (MLLMs) has opened up new possibilities for sketch interactions. Google introduced the "Circle to Search" feature [29], which allows users to search within designated areas. This interaction method, initially applied to image searches and other specific tasks [37, 56], has now been expanded to the system level. Yen et al. [106] conducted an exploratory study on code editing using sketches layered on the code interface. Expressing user intentions at the system level through sketching—across broader application scenarios involving more general and complex contexts—remains a promising yet underexplored research direction.

Pushing sketches to system-level and open-ended scenarios offers a flexible "language" to communicate with LLMs. However, the inherent ambiguity [4, 6] and input inefficiency of sketches pose challenges. Although sketches can freely convey a range of intentions and address gaps in LLM interaction, the same sketch can indicate different intents in different contexts, for example, an arrow may indicate pointing to an element, upward direction, or movement. To disambiguate their intent, users often resort to supplementary handwritten text, which further constrains input efficiency. Text constitutes an effective supplement to sketches [81, 117]; given that users' hands are frequently occupied during sketching, speech—widely employed in intelligent assistant interfaces—serves as a crucial modality for disambiguating intent and conveying supplementary information.

In this paper, we propose SketchGPT, a novel interaction paradigm with LLMs that enables users to communicate through sketches and speech directly on top of system interfaces. The core concept of SketchGPT is to support concurrent multimodal intention expression by leveraging the tight coupling between context, free-form sketch, and speech for more natural and efficient interaction, as shown in Figure 1(a). A Wizard-of-Oz study was first conducted under the assumption of fully capable LLMs, to investigate how individuals express their intentions in open-ended scenarios. The formative insights on patterns and preferences highlight the potential of the concept and further inspire the subsequent prototype. The SketchGPT system was developed in a multi-agent framework with three stages: intention interpretation, feedback generation, and user confirmation. The multimodal intentions are dynamically interpreted based on the context and temporal-semantic correlations through multiple Chain-of-Thought (CoT) agents (Figure 1(b)). Another agent generates responses from user intentions and, using an extensible toolkit, produces a concrete list of operations to bridge user input and system execution (Figure 1(c)). Following this, the user is engaged in the loop where confirmed intentions are automatically executed (Figure 1(d)).

SketchGPT complements existing interactions by enabling higher-level, more natural intention expression rather than replacing them. The evaluation of our concept and system focuses on modality influence in this novel interaction setting rather than direct baseline comparisons. We conducted a user study comparing SketchGPT with unimodal interactions (speech or sketch only), demonstrating the advantages of multimodal intention expression in terms of comprehension accuracy, expression efficiency, and user experience. In exploratory scenarios similar to daily tasks, users reported high satisfaction, highlighting the value and acceptance of the system. Our findings from observations and interviews further revealed valuable insights into SketchGPT. The studies received ethical approval from the ethics board of our institution.

In summary, our contributions are as follows: (1) We propose SketchGPT, a multimodal interaction paradigm that integrates sketches and speech for open-ended, system level interaction with large language models, enhancing the efficiency and naturalness of intention expression by leveraging the strengths of both modalities. (2) We implement a prototype system within a multi-agent framework that interprets ambiguous user intents in contexts and allows user confirmed execution through a human-in-the-loop mechanism. (3) Insights and findings from our user study underscore the advantages of our multimodal interaction over unimodal interactions.

2 Related Work

2.1 Interacting with LLMs

The increasing use of LLMs has brought the optimization of human-LLM interaction to the forefront of academic discourse. Present research predominantly centers on improving this interaction through three principal approaches.

One line of research focuses on optimizing the user's initial prompt [8, 98, 100], designing prompt workflows [103] or multi-agent processes [113] to enhance task performance. These methods do not require active user involvement during the prompt process, making them widely adopted in LLM-based systems. We apply such techniques in the SketchGPT framework for key steps like parsing user intent and executing manipulations.

Another group of studies design graphical user interfaces (GUIs) tailored for specific task types to improve LLM interaction. By introducing GUI operations [40, 59, 83, 99, 102] and visual designs [41, 83], these methods enhance prompt quality and the presentation of LLM outputs. Tasks covered include LLM chain design [102], prototype design [40], multi-step retrosynthetic route planning [83], text writing [21, 41], image generation [99], and element manipulation [59]. While improving interaction efficiency and user experience for specific tasks, these approaches face scalability limitations for broader task types. Additionally, human-LLM interaction in these works still relies on text-based prompts, with GUI operations serving a supplementary role. As a result, representing semantics like spatial or shape-related concepts, remains challenging.

The third direction examines incorporating modalities beyond text in human–LLM interaction, including eye-tracking [49], speech [54], gestures [111], sketch [106, 107] and sensor data [16]. The introduction of these modalities enhances LLM's understanding of environmental context [16, 49, 111], as well as improving the naturalness [16, 49, 111] and efficiency [54] of interactions, or providing alternative interaction methods for traditional tasks [106, 107]. A few studies have explored multimodal fusion for human-LLM interaction, combining eye-tracking with speech [11, 49] or speech with touch [26]. However, the potential of integrating sketches and speech for LLM interaction remains unexplored.

Current research on LLMs has predominantly focused on specific task scenarios, where these models have proven effective in solving challenges that traditional methods struggle with. However, existing works generally restrict LLMs to individual applications, without considering their integration into broader, multi-application workflows. Although efforts have been made to simplify LLM interactions through point-and-click and referential actions combined with text input [59], using hand-drawn sketches together with speech to convey user intent to LLMs has yet to be investigated.

2.2 Sketch Interactions

Sketch interaction, valued for its ability to support cognitive processes and foster creative expression, has been extensively researched and applied across a variety of disciplines. For example, it has been used in tasks such as drawing [46, 115], retrieving images [68], 3D models [72], audio [22], code editing [70, 80, 106, 107], document annotation [51, 52, 101], image generation [82], mobile manipulator teleoperation [39] and video content generation [57]. The applications in these fields demonstrate the capability of sketches in expressing various semantics.

Despite significant advancements, existing sketch-based interaction methods face several limitations. One challenge is the limited options for stylus input modes. Some studies enable manual switching between modes, such as toggling between solid and dashed lines to represent 3D model visibility [72], or allowing users to alternate between drawing and manipulation modes [93]. Although the work by [39] allows for free-form sketching, switching between the modes for controlling manipulator movement and grasping still requires an explicit toggle via a UI element. A more flexible approach is found in RichReview [108], which automatically adjusts pen modes based on stroke characteristics and position. However, this still relies on predefined rules rather than interpreting the user's intent based on the semantics of the sketch. Furthermore, when the task involves understanding the intent from a sketch, most systems rely on predefined gestures, which limits the flexibility of interaction. For instance, several works focusing on sketch-based document editing and annotation [51, 52, 101] designed various pen gestures for common interaction behaviors in document scenarios. However, users needed to memorize these gestures before using the system, rather than being able to apply their own preferred sketching methods for immediate use. Additionally, sketches often contain multiple elements with distinct meanings, yet many approaches treat them as a single entity, resulting in a loss of important nuances. Some methods rely on recognition techniques to identify predefined units or treat individual strokes as basic predefined elements [14, 31, 110]. Camba et al. [13] proposed interpreting groups of strokes as reusable elements that can be assembled into CAD objects. However, this is only a conceptual method for the future and has not been implemented yet.

Leveraging the capabilities of MLLMs, SketchGPT facilitates a more comprehensive understanding of sketch semantics by incorporating speech input. This enables the system to distinguish between different types of strokes, such as those used for writing, drawing or annotation, without relying on predefined gestures or input modes.

2.3 Multimodal Interface Input

Multimodal interaction allows users to engage with computers through various input methods, such as speech, eye gaze, gestures, touch, and sketches. Combining input modalities offers benefits like disambiguation, robustness, contextual adaptability [65, 66], convenience, naturalness, and efficiency [9, 67]. Early systems like "Put-That-There" [9] and QuickSet [18, 42] demonstrated the advantages of integrating speech with gestures or pen input. However, these systems used predefined or referential gestures, limiting their expressiveness and range of applications. Incorporating gaze tracking into mobile devices often requires additional hardware [15, 24, 61, 69], and built-in camera solutions still lack accuracy [48]. Touch interaction on these devices is limited by "fat finger" imprecision and difficulty in targeting specific elements [86].

Sketching can simplify language and reduce ambiguity in communication [118], but relying solely on sketches may lead to misinterpretation [6]. Verbal expression can enrich the symbolic nature of sketches and provide clarity [25]. These modalities are complementary and often used together in various settings [5, 6, 25, 44]. Recent work by Rosenberg et al. [79] developed a sketching and speaking interface for storytelling and interactive world creation, but did not parse sketch semantics and relied on traditional natural language understanding methods for speech processing. Existing research is dedicated to developing multimodal interfaces for specific applications or scenarios, such as for conversation flow control [73], 3D model retrieval [28], handling user queries on tablets [45], makeup tutorial assistance [92], robot navigation [116], and image generation [53]. These involve inputs from various modalities, including gestures [73], language [28, 45, 53, 73, 92, 116], touch [45, 73], and sketch [28, 45, 53, 116]. In multimodal processing, researchers have also proposed methods to align brushstrokes with audio in painting tutorials [62] and developed a framework to understand user intent from multimodal contexts [33]. Users tend to provide redundant speech alongside sketches [5, 6], with a temporal correlation between the two [2, 44]. Existing systems align speech and sketches based on temporal and spatial relationships [25, 43]. However, current studies on people's behavioral patterns when using speech and sketches to express ideas primarily focus on human-to-human communication in whiteboard scenarios. The main emphasis of these studies is on the temporal relationship and content redundancy between speech and sketches. There is still a lack of systematic research on how people tend to use speech and sketches to express their intentions, what types of intentions they express, and what kinds of content are included in sketches and speech respectively when interacting with LLMs.

3 Formative Study

Existing research on speech and sketch expression predominantly focuses on human-to-human communication. There remains a knowledge gap regarding the context, content, and user patterns of speech and sketch interaction between humans and LLMs. This study focuses on the following research questions:

- **RQ1:** In which contexts do users prefer to use sketch and speech for interacting with LLMs?
- **RQ2:** What features and patterns emerge during multimodal interactions between users and LLMs?
- **RQ3:** How do users prefer to initiate the interactions, and how should the system respond?

3.1 Study Design & Procedure

We employed a Wizard-of-Oz design [77] to investigate how users would naturally interact with a system-level LLM on a tablet, envisioned as an advanced agent capable of understanding and executing diverse intentions for daily tasks beyond current capabilities. Through sketch and speech, participants interacted freely across 22 common applications (Fig. 2 suppl.) as well as system functions, without predefined constraints. This setup encouraged them to express their intentions openly, capturing authentic, diverse behaviors aligned with real-world scenarios.

The study was conducted in a private room with only the participant and experimenter present, while two authors jointly acted as a single wizard outside. The wizard monitored the tablet[1] and listened to the voice to interpret intents, then remotely controlled the device to execute commands and provide instant feedback through on-screen dialogs. To handle unexpected or complex requests, the wizard studied the apps and leveraged GPT-4o and web search to ensure responsive and realistic interaction.

Before the study, participants were briefed on its background and goals, emphasizing exploration of diverse interaction intents rather than evaluation of LLM performance. After practicing and receiving procedural guidance, they were encouraged to fully engage with 5–8 different apps, with the experimenter intervening only during app transitions. The study concluded with a semi-structured interview to gather insights on interaction scenarios and methods, lasting approximately 1 to 1.5 hours.

3.2 Apparatus & Participants

We used a Surface Book 2 (15-inch) with a Surface Pen, set in View Mode[2] on a table, allowing participants to adjust its position and angle. A floating button toggled intention expression, activated by touch or stylus. Participants could simultaneously sketch on the interface overlay and speak, while interaction with underlying apps was disabled. The interface was implemented using the Windows Presentation Foundation (WPF) framework.

We recruited 10 university students (aged 20-25 years, M=21.9; 6 female and 4 male) from diverse majors, referred to as FP1–FP10 in the remainder of the paper. All participants regularly used touch devices and a stylus, and had prior LLM experience, with some involved in related research. They provided informed consent and received $10/hour compensation. Detailed participant backgrounds are provided in Sec. 2.2 suppl.

[1] AnyDesk: remote control software, https://anydesk.com/en

[2] The Surface Book can function as a powerful tablet, with View Mode being an official configuration for tablet use; see https://tinyurl.com/yck9v4b.

3.3 Analysis

We adopted a mixed-method approach to understand the contexts, patterns, and preferences for multimodal interaction, combining thematic analysis of recorded study videos with study and interview notes. The analysis contained two coding phases. **(1) Exploratory phase**: two authors independently conducted open coding on each modality until saturation, followed by discussions among three authors to refine codes via axial coding and thematic analysis. **(2) Formal phase**: three authors first jointly coded 20% data, resolved discrepancies through discussion, and then two authors coded an additional 10% to achieve inter-coder agreement. One author subsequently coded the remaining data using the finalized codebook.

In total, we collected 215 intentions across 251 interaction iterations, amounting to 332 minutes of interaction time. Although omissions may exist, subjectivity was carefully minimized through iterative discussion and multi-stage review.

3.4 Dimensions of Intention Expression

From the preceding analysis, we identified core dimensions that summarize fundamental modes of intention expression, as shown in Figure 2. For each code, we provide a definition, occurrence frequency, and illustrative examples. Note that each intention expression may be composite and encompass multiple codes. More case examples are provided on the project webpage: https://zaynehuang.github.io/SketchGPT/.

3.4.1 Intention Scope and Context. User intentions range from global settings adjustments (e.g., display configuration, window management) to fine-grained, in-application operations. These intentions unfold across on-screen elements, background applications, and the broader conversational environment. In practice, speech naturally bridges between these contexts, while sketches tend to anchor directly to the current interface, providing immediate visual linkage to on-screen targets (see Figure 2(a)–(b)).

3.4.2 Sketch and Speech Content. Sketches and speech exhibit distinct content characteristics. Sketches enable rapid visual externalization of ideas that are difficult to articulate verbally, including references to on-screen elements, spatial arrangements, logical structures, and creative annotations. Speech, in contrast, offers efficiency and linguistic precision for specifying detailed requests, clarifying steps, and acquiring information, and was often considered "*faster than writing*" (FP10). Together, these modalities reduced the need for manual operations, supported continuous cognitive flow, and complemented each other to facilitate rich and flexible intention expression (see Figure 2(c)–(d)).

3.4.3 Modality Temporal and Semantic Relationship. The temporal and semantic interplay between sketches and speech is central to effective multimodal interaction. As shown in Figure 2(e), these modalities were most often used concurrently, enabling users to express intentions fluidly and maintain cognitive flow; in other cases, they appeared sequentially in a flexible order depending on context. Semantically, our analysis was informed by the CARE model [20], as shown in Figure 2(f). We found that sketches and speech frequently worked together to produce a more complete and nuanced expression of intentions, each leveraging its distinct strengths. At times, one modality fully conveyed the intent while the other served as optional or supportive, reflecting flexible redundancy patterns. In our case, we did not observe pure assignment, as any incomplete modality input was always complemented by the other. This combination "*reduced manual operations*" (FP5, FP9), supported "*creative expression*" (FP1, FP3, FP10), and allowed intentions to be conveyed "*more naturally and completely*" (FP6).

3.5 Dynamics of Multimodal Interaction

Beyond intention expression, we focus on the dynamic process of multimodal interaction.

3.5.1 Intent Granularity. Participants found multimodal input to be a "*natural and intuitive*" way to express intentions (FP3, FP8, FP10). Assuming highly capable LLMs, they conveyed intentions at all levels—from concrete commands to abstract tasks—across diverse applications. Both sketches and speech expressed desired outcomes and process instructions, trusting LLMs to decompose, plan, and execute accordingly.

3.5.2 Ambiguity Resolution. Open-ended intentions "*fully leveraged each modality and contextual cue to enhance clarity*" (FP5). Single modalities often led to ambiguity, while integrating multiple modalities semantically and temporally enabled participants to express intentions more clearly and completely. When a single modality did not fully convey their intentions, participants often clarified or supplemented with another modality or a combination.

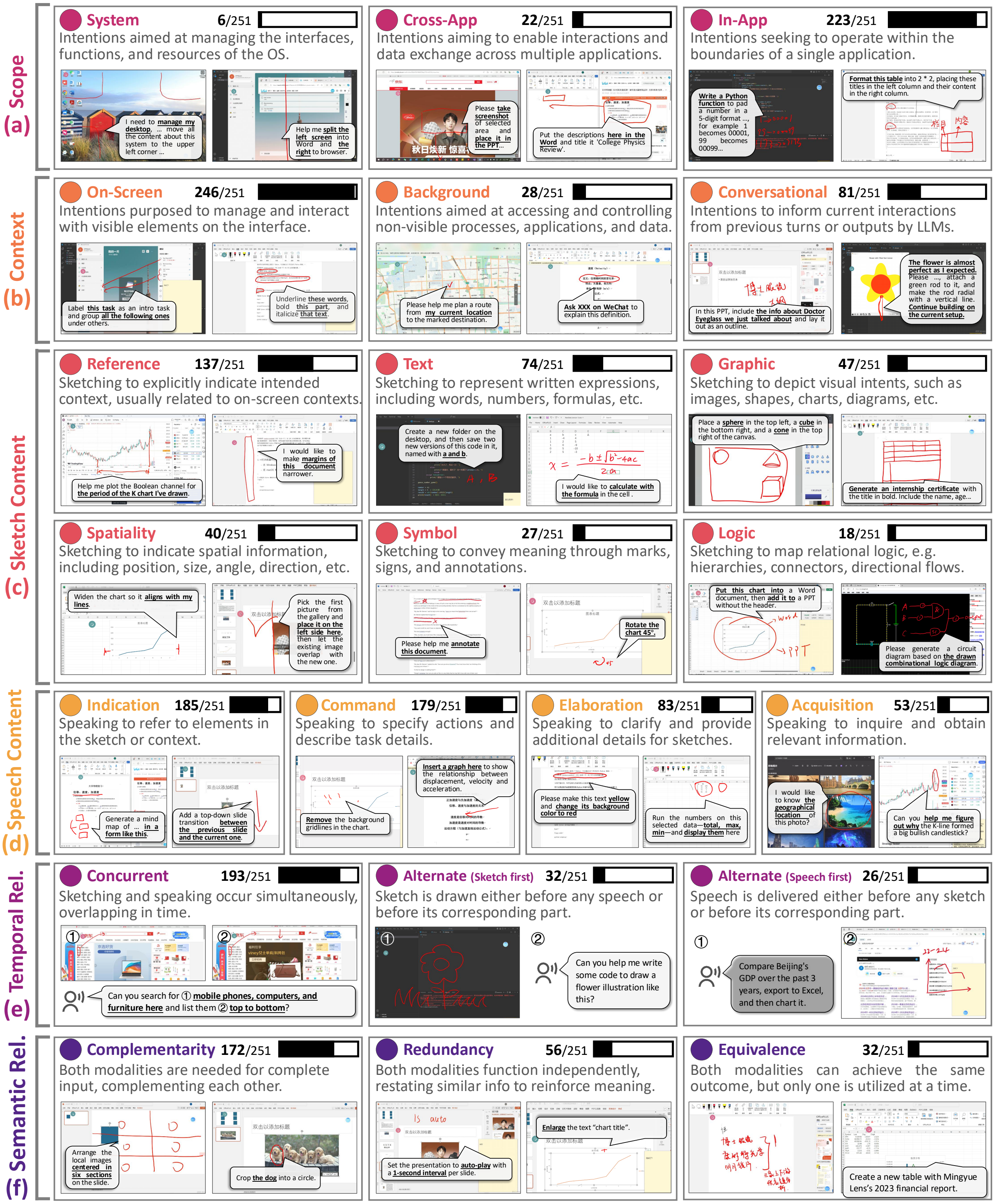
3.5.3 Feedback & Iteration. Participants found "*dialogue-style feedback of LLMs unsuitable*" (FP4), preferring embedded, contextual responses. They also wanted transparency and control, including "*visibility into LLMs' actions and progress*" (FP1, FP2, FP10), but favored an iterative workflow to "*selectively observe and revise outcomes*" (FP3, FP5, FP9, FP10). Iteration was common: participants preferred undoing and re-expressing intentions over editing prior sketches, considering modification "*tedious*" (FP2, FP3, FP5, FP10), assuming reversible actions for lightweight trial and error.

3.5.4 Interaction Initiation. Participants desired a "*more fluid and seamless*" (FP1, FP3) way to initiate interactions. Since intentions were often global, they suggested global triggers such as floating buttons, pull-down menus, multi-touch gestures, or wake words for a more "*intuitive and lively*" experience (FP1, FP3, FP9, FP10). The stylus was also seen as a natural trigger, with some proposing "*a physical button for direct activation*" (FP1, FP2, FP10).

3.6 Example Usage Scenario

We summarize the following usage scenarios from the use cases in this study. Words with underline represent sketch input, while words in italic represent speech input. The icons represent speech and sketch usage across the coding dimensions in Figure 2.

Tom is analyzing data in Excel and visualizing it through charts, expressing in-application A3 intents. First, he selects three columns, writes "max" below each, and says E2, "*I want to calculate the maximum value of each of these columns here...*" Then, he writes "max/min" and adds E1, "*and then put the range ratio of these values here.*" (Figure 3(a)) He then circles the data, sketches the chart and says E1, "*I want to use these data... to create a chart here, without gridlines.*" He marks the title position, stating, E1 "*The title is 'Data*



System 6/251
Intentions aimed at managing the interfaces, functions, and resources of the OS.

I need to **manage my desktop**, ... move all the content about this system to the upper left corner ...

Help me **split the left screen** into Word and **the right** to browser.

Cross-App 22/251
Intentions aiming to enable interactions and data exchange across multiple applications.

Please **take screenshot** of selected area and **place it in the PPT**...

Put the descriptions **here in the Word** and title it 'College Physics Review'.

In-App 223/251
Intentions seeking to operate within the boundaries of a single application.

Write a Python function to pad a number in a 5-digit format ..., for example 1 becomes 00001, 99 becomes 00099...

Format this table into 2 * 2, placing these titles in the left column and their content in the right column.

On-Screen 246/251
Intentions purposed to manage and interact with visible elements on the interface.

Label **this task** as an intro task and group **all the following ones** under others.

Underline **these words**, bold **this part**, and italicize **that text**.

Background 28/251
Intentions aimed at accessing and controlling non-visible processes, applications, and data.

Please help me plan a route from **my current location** to the marked destination.

Ask XXX on WeChat to explain this definition.

Conversational 81/251
Intentions to inform current interactions from previous turns or outputs by LLMs.

In this PPT, include **the info about Doctor Eyeglass we just talked about** and lay it out as an outline.

The flower is almost perfect as I expected. Please ..., attach a green rod to it, and make the rod radial with a vertical line. **Continue building on the current setup.**

Reference 137/251
Sketching to explicitly indicate intended context, usually related to on-screen contexts.

Help me plot the Boolean channel for **the period of the K chart I've drawn**.

I would like to make **margins of this document** narrower.

Text 74/251
Sketching to represent written expressions, including words, numbers, formulas, etc.

Create a new folder on the desktop, and then save two new versions of this code in it, named with **a and b**.

I would like to **calculate with the formula** in the cell .

Graphic 47/251
Sketching to depict visual intents, such as images, shapes, charts, diagrams, etc.

Place a **sphere** in the top left, a **cube** in the bottom right, and a **cone** in the top right of the canvas.

Generate an internship certificate with the title in bold. Include the name, age...

Spatiality 40/251
Sketching to indicate spatial information, including position, size, angle, direction, etc.

Widen the chart so it **aligns with my lines**.

Pick the first picture from the gallery and **place it on the left side here**, then let the existing image overlap with the new one.

Symbol 27/251
Sketching to convey meaning through marks, signs, and annotations.

Please help me **annotate this document**.

Rotate the chart 45°.

Logic 18/251
Sketching to map relational logic, e.g. hierarchies, connectors, directional flows.

Put this chart into a Word document, then **add it to** a PPT without the header.

Please generate a circuit diagram based on **the drawn combinational logic diagram**.

Indication 185/251
Speaking to refer to elements in the sketch or context.

Generate a mind map of ... **in a form like this**.

Add a top-down slide transition **between the previous slide and the current one**.

Command 179/251
Speaking to specify actions and describe task details.

Remove the background gridlines in the chart.

Insert a graph here to show the relationship between displacement, velocity and acceleration.

Elaboration 83/251
Speaking to clarify and provide additional details for sketches.

Please make this text **yellow** and **change its background color to red**

Run the numbers on this selected data—**total**, **max**, **min**—and **display them** here

Acquisition 53/251
Speaking to inquire and obtain relevant information.

I would like to know **the geographical location** of this photo?

Can you **help me figure out why** the K-line formed a big bullish candlestick?

Concurrent 193/251
Sketching and speaking occur simultaneously, overlapping in time.

Can you search for ① **mobile phones, computers, and furniture here** and list them ② **top to bottom**?

Alternate (Sketch first) 32/251
Sketch is drawn either before any speech or before its corresponding part.

Can you help me write some code to draw a flower illustration like this?

Alternate (Speech first) 26/251
Speech is delivered either before any sketch or before its corresponding part.

Compare Beijing's GDP over the past 3 years, export to Excel, and then chart it.

Complementarity 172/251
Both modalities are needed for complete input, complementing each other.

Arrange the local images **centered in six sections** on the slide.

Crop **the dog** into a circle.

Redundancy 56/251
Both modalities function independently, restating similar info to reinforce meaning.

Set the presentation to **auto-play** with a **1-second interval** per slide.

Enlarge the text "chart title".

Equivalence 32/251
Both modalities can achieve the same outcome, but only one is utilized at a time.

Create a new table with Mingyue Lens's 2023 financial report.

Figure 2: Dimensions of intention expression: (a) intention scope, (b) intention context, (c) sketch content (d) speech content, (e) modality temporal relationship, (f) modality semantic relationship. Examples are provided where red strokes indicate sketches, and the message box contains speech transcription with bold and underlined text highlighting the related expression.

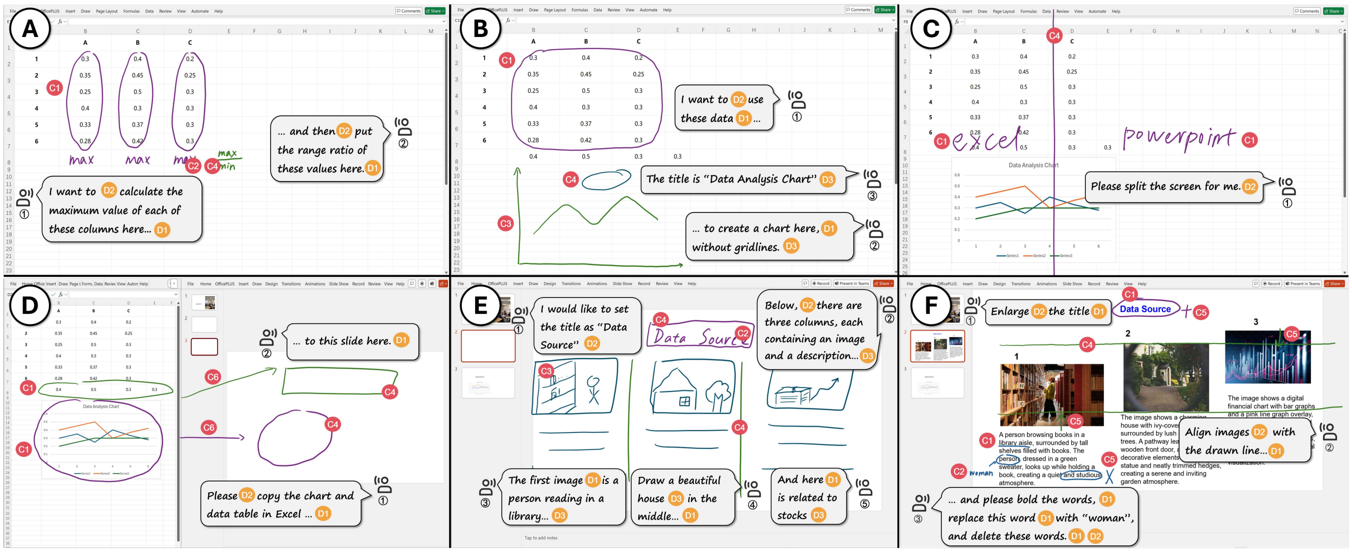

A
I want to D2 calculate the maximum value of each of these columns here... D1
... and then D2 put the range ratio of these values here. D1
max
max
max
max
min
B
I want to D2 use these data D1 ...
The title is "Data Analysis Chart" D3
... to create a chart here, D1 without gridlines. D3
C
excel
powerpoint
Please split the screen for me. D2
D
... to this slide here. D1
Please D2 copy the chart and data table in Excel ... D1
E
I would like to set the title as "Data Source" D2
Data Source
Below, D2 there are three columns, each containing an image and a description... D3
The first image D1 is a person reading in a library... D3
Draw a beautiful house D3 in the middle... D1
And here D1 is related to stocks D3
F
Enlarge D2 the title D1
Data Source
Align images D2 with the drawn line... D1
woman
... and please bold the words, D1 replace this word D1 with "woman", and delete these words. D1 D2

Figure 3: We present an illustrative usage scenario, with the text inside the dialogue box representing user's speech input: (A) User circles data, inputs formulas, while specifying result placement via speech. (B) User sketches a chart and refines contents with speech. (C) Texts and symbols are used to convey split-screen operation and position. (D) Circles and arrows indicate cross-app transfer. (E) User sketches the layout and graphics with speech providing details. (F) Symbolic sketches are used to indicate layout adjustments and content edits.

Analysis Chart'." (Figure 3(b)) In order to view Excel and PowerPoint at the same time, Tom demonstrates a system-level A1 intent. Tom draws a line, labeling "Excel" on the left and "PowerPoint" on the right, and says E2, "*Please split the screen for me.*" (Figure 3(c))

Tom wanted to move data and charts from Excel to PowerPoint, thus expressing a cross-application A2 intent. He first says,"*Please copy the chart and data table in Excel to this slide here.*". After saying that E3, he circles the chart and results in Excel, marks corresponding positions in PowerPoint, and uses arrows (Figure 3(d)) After copying the charts and data from Excel, Tom continues working on his PowerPoint presentation, thus expressing in-application A3 intents. At the top-center of the blank page, he sketches a title box labeled "Data Source" and says E1, "*I would like to set the title as 'Data Source'.*" Then, he sketches the basic layout of the slide and explains E1, "*Below, there are three columns, each containing an image and a description.*" From left to right, he sketches the image details and explains E1, "*The first image is a person reading in a library... Draw a beautiful house in the middle... And here is related to stocks.*" (Figure 3(e)) Later, he revises details. He circles the title and draws a plus sign, saying E1, "*Enlarge the title.*" Noticing misaligned images, he first draws two horizontal lines on the slide and says E2, "*Align images with the drawn line.*" Tom also wants changes in the text sections, sketching lines, circles, crosses and writing word. He says E1, "*Bold the words, replace this word with 'woman,' and delete these words.*" (Figure 3(f)) In panels E and F, the user again expressed in-application A3 intents.

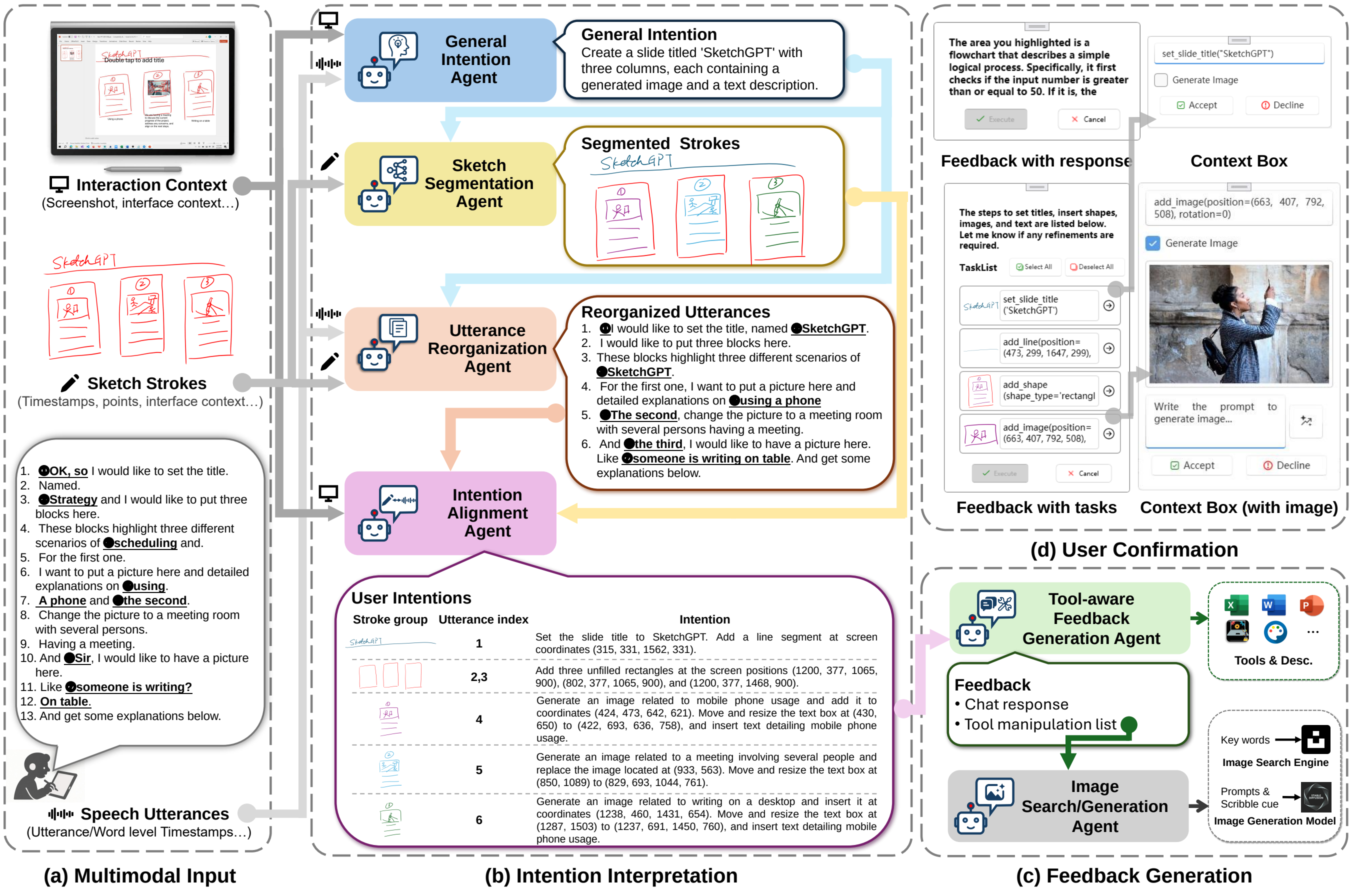
When analyzing data in Excel and preparing his PowerPoint slides, Tom primarily operates on on-screen B1 elements and expresses complementary F1 sketch and speech intents. In the split-screen scenario, bringing up Powerpoint represents a background B2 intent.

4 SketchGPT

Informed by insights from the formative study, we designed and developed a prototype system named SketchGPT, as the first attempt to interpret and execute open, application-agnostic intentions expressed via sketch and speech. This new task requires no additional training data and transcends application-specific contexts, enabling more flexible interactions. SketchGPT uses multiple CoT agents to interpret intentions, generate tool manipulation lists, and produce chat responses. Users then enter a confirmation phase to preview and selectively execute tasks. The framework is illustrated in Figure 4 with the prompts details provided in Sec. 1 suppl.

4.1 Interpreting Multimodal User Intentions

To understand user intent from closely linked modalities, We developed a multi-agent workflow with CoT agents to decompose unimodal intentions, align them temporally and semantically with context, and form a complete understanding of the user's intent. CoT prompting is a reasoning framework designed to break down complex problems into intermediate steps, enabling agents to iteratively analyze and combine multimodal data [100]. This approach ensures that multimodal dependencies are systematically considered, leading to more accurate intent interpretation [114]. The process of intention interpretation is shown in Figure 4(b).



1. ●**OK, so** I would like to set the title.
2. Named.
3. ●**Strategy** and I would like to put three blocks here.
4. These blocks highlight three different scenarios of ●**scheduling** and.
5. For the first one.
6. I want to put a picture here and detailed explanations on ●**using**.
7. **A phone** and ●**the second**.
8. Change the picture to a meeting room with several persons.
9. Having a meeting.
10. And ●**Sir**, I would like to have a picture here.
11. Like ●**someone is writing?**
12. **On table**.
13. And get some explanations below.

General Intention
Create a slide titled 'SketchGPT' with three columns, each containing a generated image and a text description.

Reorganized Utterances

1. ●I would like to set the title, named ●**SketchGPT**.
2. I would like to put three blocks here.
3. These blocks highlight three different scenarios of ●**SketchGPT**.
4. For the first one, I want to put a picture here and detailed explanations on ●**using a phone**
5. ●**The second**, change the picture to a meeting room with several persons having a meeting.
6. And ●**the third**, I would like to have a picture here. Like ●**someone is writing on table**. And get some explanations below.

User Intentions

Stroke group	Utterance index	Intention
SketchGPT	1	Set the slide title to SketchGPT. Add a line segment at screen coordinates (315, 331, 1562, 331).
	2,3	Add three unfilled rectangles at the screen positions (1200, 377, 1065, 900), (802, 377, 1065, 900), and (1200, 377, 1468, 900).
	4	Generate an image related to mobile phone usage and add it to coordinates (424, 473, 642, 621). Move and resize the text box at (430, 650) to (422, 693, 636, 758), and insert text detailing mobile phone usage.
	5	Generate an image related to a meeting involving several people and replace the image located at (933, 563). Move and resize the text box at (850, 1089) to (829, 693, 1044, 761).
	6	Generate an image related to writing on a desktop and insert it at coordinates (1238, 460, 1431, 654). Move and resize the text box at (1287, 1503) to (1237, 691, 1450, 760), and insert text detailing mobile phone usage.

The area you highlighted is a flowchart that describes a simple logical process. Specifically, it first checks if the input number is greater than or equal to 50. If it is, the

✓ Execute ✕ Cancel

set_slide_title("SketchGPT")

□ Generate Image

Accept Decline

The steps to set titles, insert shapes, images, and text are listed below. Let me know if any refinements are required.

TaskList Select All Deselect All

- set_slide_title ('SketchGPT')
- add_line(position= (473, 299, 1647, 299),
- add_shape (shape_type='rectangl
- add_image(position= (663, 407, 792, 508),

✓ Execute ✕ Cancel

add_image(position=(663, 407, 792, 508), rotation=0)

☑ Generate Image

Write the prompt to generate image...

Accept Decline

Feedback
- Chat response
- Tool manipulation list

Figure 4: SketchGPT framework. (a) Multimodal Input: The user's speech and sketch inputs, along with interaction contexts, are transformed into Interaction Contexts, Sketch Strokes, and Speech Utterances, serving as inputs to SketchGPT. (b) Intention Interpretation: The General Intention Agent extracts the core intention from these inputs, guiding the Sketch Segmentation Agent and Utterance Reorganization Agent. Subsequently, the Intention Alignment Agent integrates the outputs of the Sketch Segmentation Agent and the Utterance Reorganization Agent to create the User Intention List. (c) Feedback Generation: Based on user intentions and available tools, the Tool-aware Feedback Generation Agent converts the intentions into a tool manipulation list, along with a chat response. (d) User Confirmation: Finally, the feedback is visualized through the interface, allowing the user to preview and confirm each manipulation.

4.1.1 *General Intention Understanding.*

In interactions, the user's speech and sketches can be semantically redundant or complementary. Analyzing either modality in isolation may lead to misinterpretation or bias. Therefore, we first combine data from both speech and sketch modalities to infer a general intent, which guides the subsequent intent analysis, helping to avoid unimodal bias. Upon completion of the interaction by the user, the Interaction Context, Sketch Strokes, and Speech Utterances are fed into the General Intention Agent, as illustrated in Figure 4(a). Specifically, the Interaction Context comprises screenshots and interface information; the Sketch Strokes are downsampled timestamped point sets; and the Speech Utterances are transcriptions with word-level timestamps. The General Intention Agent summarizes the user's core intention in concise natural language. This general intent serves as a high-level guide, which is then passed on to subsequent stages of multimodal intent analysis, ensuring that the detailed processing remains aligned with the user's overarching goals.

4.1.2 *Sketch Segmentation.*

A single user sketch may consist of multiple semantic segments. For instance, one stroke may indicate a reference, while another may convey textual content. To accurately infer the detailed execution steps, we group strokes based on their semantics. We use the DBSCAN [23] clustering algorithm as the initial step because it is well-suited for identifying clusters of arbitrary shapes and does not require the number of clusters to be specified in advance. DBSCAN groups strokes by identifying dense regions in a feature space, making it effective for spatial and temporal stroke data. To handle variability in sketches, we determine the DBSCAN algorithm's parameters by analyzing k-distance plots. However, it is limited by its reliance solely on spatial and temporal distance, which may lead to misclassification as it cannot capture the semantic meaning of strokes. To address these limitations, we utilize the Sketch Segmentation Agent to refine the clustering. The result from DBSCAN serves as a preliminary reference, and the Sketch Segmentation Agent further optimizes the segmentation to better align with the intended semantics. This agent incorporates not only the general intent but also global visual semantics, ensuring a more accurate segmentation.

4.1.3 Utterances Reorganization. People often experience pauses, repetitions, and inaccuracies in their verbalizations. Speech recognition systems capture words spoken by the user, but they rely on inter-word spacing to define utterances, which may not align with the actual semantic structure. Consequently, the transcribed text often contains noise (Figure 4 ①) and segmentation errors (Figure 4 ④⑤⑦) that hinder accurate intent understanding. To address this, we introduce the Utterance Reorganization Agent to clean and reorganize the transcribed text. This agent not only processes the original utterances based on temporal separation and the general intent but also refines the utterances by eliminating noise and correcting segmentation errors, producing optimized transcriptions aligned with natural speech patterns. Moreover, the Utterance Reorganization Agent leverages visual information from the sketch to further improve accuracy. It compensates for the inherent limitations of speech recognition systems. These systems often struggle with out-of-vocabulary words, which are common in scenarios like meetings or creative design discussions [44]. For example, as shown in Figure 4 ②③, a term like "SketchGPT," frequently used in such contexts, may be misrecognized as "Strategy". By utilizing visual cues from the sketch, the agent can correct such errors.

4.1.4 Intention Alignment. At this stage, we have refined both the utterances and sketch segmentation results. To achieve accurate semantic alignment between stroke groups and their corresponding utterances, we utilize the Intention Alignment Agent. This agent integrates the outputs from the Sketch Segmentation Agent and the Utterance Reorganization Agent to ensure that the alignment is semantically coherent. Prior research [2, 44] suggests a general temporal correspondence between written and spoken content. However, discrepancies can arise, such as when users repeat previously written elements. To address these issues, the Intention Alignment Agent corrects initial temporal mappings based on semantic coherence. It generates a user intent list, where each entry is a triplet consisting of a stroke group, a transcribed text segment, and a user intention. Throughout the user intention interpretation process, we repeatedly employed a strategy where the agent optimizes the initial rule-based results by leveraging contextual semantic information. This approach aims to deeply parse multimodal intentions while trying to minimize ambiguity.

4.2 Generating Feedback from User Intentions

When the user's multimodal intent is parsed into a list of intent units, the next task for SketchGPT is to organize these intent units into a Tool Manipulation List, which includes tool names and parameters for direct execution by the Tool Agent, and a Chat Response, which provides a natural language description of manipulations for user understanding. The detailed process is illustrated in Figure 4(c).

4.2.1 Tool-aware Manipulation Steps Generation. To prevent misuse, the Tool-aware Feedback Generation Agent only invokes a tool when the prompt explicitly requests tool usage. To enhance the stability of tool invocation, the agent converts the user's intent into a format of tool name and corresponding parameters, which are then passed to the Tool Agent for execution. Specifically, tool functions and their associated annotations are incorporated into the prompt.

4.2.2 Chat Response Generation. Text containing tool functions and parameters is often not suitable for non-specialists to understand. To help users quickly grasp the execution details, a natural language explanation of the operations should be generated. Additionally, this explanation supports the system's question-and-answer functionality by addressing user acquisitions. The Tool-aware Feedback Generation Agent is responsible for producing a natural language summary that succinctly captures the user's intent and the task execution plan, and for responding to user queries made during the interaction.

4.2.3 Image Search and Generation. Standard LLMs do not support image generation or retrieval. To meet users' frequent image needs during operations, we integrate both image search and image generation API tools. The Image Search/Generation Agent is tasked with inferring relevant keywords that describe the desired image by combining the user's verbal expressions and the shape of the sketch. For image search, these keywords are used as search terms, and a randomly selected image from the relevant search hits is returned (supported by the API provider). For image generation, the keywords and sketch strokes are used as the prompt for generating the image.

4.3 User Confirmation

According to the recommendations of [85], artificial intelligence technologies should allow users to exert appropriate control while providing high levels of automation, in order to enhance the system's reliability, safety, and trustworthiness. To support this, after SketchGPT completes the feedback generation phase, the system transitions to a user confirmation phase. In this phase, task details are clearly presented through a Task List and Context Boxes (as shown in Figure 4(d)), allowing users to confirm tasks. This approach integrates human input seamlessly into the task execution loop, ensuring a more precise, safe, and user-guided process.

4.3.1 Task Overview. The Task List displays individual tasks generated by SketchGPT, each linked to a contextual action. Users can review tasks individually and select multiple tasks for batch processing. For each task, users have the option to execute, discard, or adjust parameters. This interface supports both individual and bulk task management, streamlining the workflow.

4.3.2 Context Box. Each task in the Task List is linked to a Context Box, accessible via double-click or a single-click on the button with a right-arrow, which offers a detailed view of the task parameters. In this interface, users can review task parameters, choose to execute the task, or discard it. In addition, for tasks involving image generation, users have the capability to preview the generated images. If the resulting image does not meet the expected criteria, users can input new image generation prompts and regenerate the images accordingly.

4.4 Executing Operations in Typical Scenarios

To enhance the interaction loop, SketchGPT is equipped with common tools that allow the agent to perform keyboard and mouse operations. However, current LLMs may encounter challenges in executing complex interface tasks accurately and consistently [63]. To address these challenges, we have developed specialized toolkits for typical scenarios involving text, canvas, and data tables.

4.4.1 Common Interface Operations. SketchGPT operates by providing the LLM with access to a custom-developed toolkit of interface operations. Similar to external tools or APIs of LLMs [1], our system exposes a set of specialized interface manipulation tools that the model can select and invoke. When receiving an instruction, the LLM analyzes the task requirements and determines which tools to call from this toolkit to accomplish the task. These tools are implemented using PyAutoGUI [89], which enables the system to programmatically control keyboard and mouse operations. While platform-specific automation APIs (e.g., macOS Automation[3]) could provide deeper system integration, we chose PyAutoGUI for its cross-platform compatibility and ability to work with any application without requiring specific API support. To enhance the interaction loop, SketchGPT is equipped with common tools that allow the agent to perform keyboard and mouse operations. However, current LLMs may encounter challenges in executing complex interface tasks accurately and consistently [63]. To address these challenges, we have developed specialized toolkits for typical scenarios involving text, canvas, and data tables.

4.4.2 Document Editing and Annotation. SketchGPT combines speech and sketch modalities, enabling users to express their editing and annotation intentions to LLMs. To effectively implement these intentions within documents, we have developed a dedicated document toolbox. Document editing typically involves the addition and removal of text, images, tables, and other content. Document annotation often requires format adjustments, including highlighting, bolding, italicizing, strikethroughs, and color changes. To enable SketchGPT to effectively understand and modify document formatting, the Tool Agent supports operations within the scope of both Markdown syntax and Microsoft Word, facilitating editing and tagging in both formats.

4.4.3 Canvas Design and Operations. Sketching plays a crucial role in facilitating creative activities [96]. By using SketchGPT for canvas design, users can alleviate the burden of complex interface operations and concentrate on their creative expression. The system provides functionalities for adding, deleting, and updating text boxes, images, shapes, and more for canvas design tasks, encompassing a wide range of features that are relevant to the actual design process.

4.4.4 Data Table Management and Analysis. The table data toolbox encompasses common table operations, including writing, deleting, merging cells, inserting charts, and inputting formulas.

4.5 Implementation Details

The user interface of SketchGPT was developed using the WPF framework, incorporating the WPF UI component library [50] for modern design elements. In inactive mode, the program is represented by a green button icon ● on the desktop, which users can reposition to avoid obstructing on-screen information. Upon clicking the button icon or pressing the stylus side button, SketchGPT is activated, and the button icon changes to pink ●. The user can then engage in multimodal interactions with SketchGPT until the button icon is clicked or the stylus side button is pressed again.

The system uses Azure Speech-to-Text service [60] for speech input recognition, with backend communication managed through a Redis database for efficient data exchange. The backend constructs reasoning chains with LangGraph [38], leveraging the multimodal capabilities of OpenAI's GPT-4o model [64] for inference. LangGraph's built-in human-in-the-loop mechanism ensures seamless integration of automated reasoning with user confirmation.

For image-related functionalities, the system employs an image search module that uses keywords derived from the user's speech and hand-drawn sketches to retrieve relevant thematic images via the Unsplash API [94]. If no suitable images are found, the system transitions to an image generation module. This module leverages a Stable Diffusion v1.5 model [78], conditioned on scribble images through ControlNet [112], and is integrated with FastAPI [75] to enable image generation based on sketches and prompts.

5 Evaluation Study

Despite prior comparisons of sketch, speech, and multimodal inputs [5, 6, 25, 44], impacts of modalities in open and application-agnostic intention input for LLMs remains unexplored. Our user study contains two sessions to address the following research questions:

- **RQ1:** How do different interaction modalities affect user performance and experience?
- **RQ2:** Can SketchGPT effectively support real-world tasks and gain user acceptance?

5.1 Study Design

5.1.1 Session 1: Comparison between Input Modalities. This session focuses on how interfaces with different input modalities affect efficiency, user experience, and LLM interpretation of user intent. To assess input experience and reasoning accuracy, participants perform and evaluate a single complete input-output process.

Conditions. This session includes three INTERFACE conditions: SketchGPT and two unimodal baselines, SketchOnly and SpeechOnly. The interfaces were kept identical, with the only difference being the exclusion of the missing modality in each case.

Tasks. Given that manipulating text and images covers most use cases in formative studies, we selected image-rich documents as our experimental setting. We designed tasks to evaluate four common user INTENTIONS: adding, removing, modifying, and moving content. Each INTENTION was tested through three tasks: expressing the intention on text at a single location, text across two locations, and image content. For example, "remove text across two locations," or "modify a specified object in the image," etc. For detailed descriptions of all tasks, please refer to Sec 3.3 suppl. Participants

[3] Automator User Guide: https://support.apple.com/guide/automator/welcome/mac

were provided with Markdown documents containing both text and images to maintain content alignment.

Design. The study followed a within-subject design to ensure that each participant was exposed to all conditions and tasks in a balanced manner. INTERFACE and INTENTION appeared an equal number of times in each position. Participants were assigned to various INTERFACE-version-INTENTION combinations according to the Graeco-Latin Square, ensuring that each participant encountered all INTERFACE and versions in a counterbalanced order. We alternate the order of INTENTION and tasks for each participant to balance the order and sequence effects.

We gathered the interaction data and subjective ratings for 3 INTERFACE × 4 INTENTION × 3 task × 12 PARTICIPANT = 432 trials.

5.1.2 Session 2: Exploration within Typical Scenarios. Document, spreadsheet, and presentation creation/editing were the most common scenarios in our formative study. We used these scenarios to simulate everyday SketchGPT usage and evaluate its user experience. Participants used SketchGPT with Typora[4], Excel, and PowerPoint, iterating until satisfied. Tasks were balanced using a pre-generated order, and subjective ratings were collected via a 7-point Likert scale.

5.2 Procedure and Apparatus

Participants were briefed on the study, provided consent, and learned each interface via video tutorials. They completed trials as per task instructions, rated intention closeness after LLM processing, and provided task load and subjective evaluations upon completing all trials for each interface. In Session 2, participants were introduced to the tasks and full usage of SketchGPT through tutorials, completed multiple iterations per task, and finally provided overall ratings and participated in semi-structured interviews.

The study was conducted in a dedicated room, instrumented with video and audio recording equipment. In Session 1, interfaces were adapted from SketchGPT, ensuring differences arose only from input variations. Participants viewed task content and provided ratings on a separate device. Inferred intentions were shown as SketchGPT chat responses. In Session 2, participants freely used all SketchGPT features to complete tasks.

5.3 Participants

Both studies involved 12 participants (aged 21-27, M = 23.5; 6 female, 6 male) recruited from our institution and other universities, denoted as EP1-EP12. None of the participants had taken part in the previous study. All participants were experienced with touchscreen devices, including smartphones and tablets, and many had also used large displays, touch-screen laptops, and interactive pen displays, with varying frequencies of stylus use. Additionally, they had prior experience with LLMs for conversational Q&A, programming, text processing, image generation, and personal assistance, with some involved in LLM-related R&D. All participants provided informed consent and were compensated $10/hour for their time.

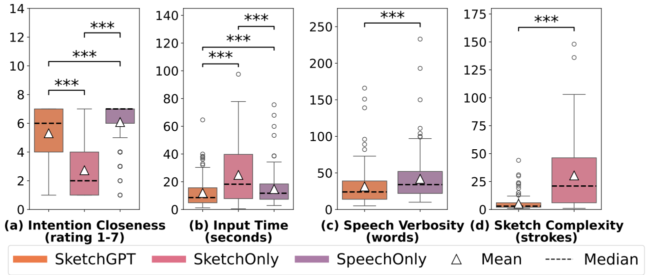


Figure 5: Interaction performance measures from Evaluation Study, Session 1. The x-axis includes three input interfaces: SketchGPT and its baselines, SketchOnly and SpeechOnly. (* indicates $p < .05$, ** indicates $p < .01$, *** indicates $p < .001$)

5.4 Quantitative Results

For the statistical analysis, we used Wilcoxon signed-rank tests to analyze the data. Statistical significance was established at $p < .05$ for all analyzes.

5.4.1 Intention Closeness. The system’s accuracy in interpreting user intentions was rated on a 7-point Likert scale, from “distant” to “close”. As shown in Figure 5(a), SpeechOnly scored highest, surpassing SketchOnly ($p < .001$) and SketchGPT ($p < .001$), while SketchGPT outperformed SketchOnly ($p < .001$). SpeechOnly ($M = 6.08$) slightly exceeded SketchGPT ($M = 5.31$) but significantly outperformed SketchOnly ($M = 2.73$). By INTENTION, SketchGPT and SpeechOnly showed no significant difference in “modify” and “move” tasks but differed in “add” and “remove” tasks, where SpeechOnly performed better. LLMs process text better than visuals, limiting SketchOnly’s effectiveness. SketchGPT helped by integrating sketches but remained less precise than speech. Users in SpeechOnly provided clear descriptions, while sketching introduced variability, reducing inference accuracy.

5.4.2 Input Time. Input time was measured from the first stroke or spoken word to the last, assessing temporal efficiency. As shown in Figure 5(b), SketchGPT had the lowest input time, outperforming SketchOnly ($p < .001$) and SpeechOnly ($p < .001$), while SpeechOnly was faster than SketchOnly ($p < .001$). With SketchOnly, users often wrote additional text for clarity, increasing input time. SketchGPT allowed simultaneous sketching and speaking, improving efficiency—users could sketch for references and speak for complex details. By intention type, SketchGPT and SpeechOnly showed no significant difference for add and modify. For remove, SketchGPT and SketchOnly were similar, both faster than SpeechOnly ($p = .005$). For move, SketchOnly and SpeechOnly showed no difference. In all other cases, SketchGPT was significantly faster, highlighting the efficiency of multimodal input.

5.4.3 Speech Verbosity. Speech verbosity, measured by word count, was significantly lower in SketchGPT than SpeechOnly ($p < .001$), as shown in Figure 5(c). This reduction was consistent across all INTENTION, indicating that sketches effectively replace verbose descriptions. Lower Mental Demand in SketchGPT further supports this efficiency.

[4] Typora: a commonly used Markdown document editor, https://typora.io/

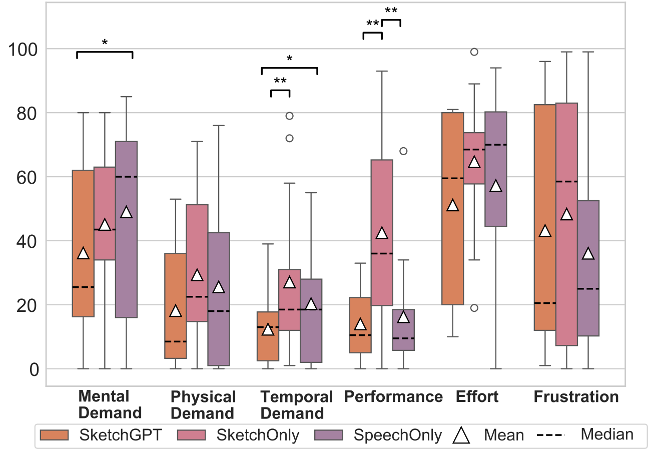


Figure 6: Task load results in NASA-TLX from Evaluation Study, Session 1 (the lower the better). The x-axis displays the six NASA-TLX subscales. The y-axis shows the corresponding scores. (* indicates $p < .05$, ** indicates $p < .01$, * indicates $p < .001$)**

5.4.4 Sketch Complexity. Sketch complexity, measured by stroke count, was significantly lower in SketchGPT than SketchOnly ($p < .001$), as shown in Figure 5(d). This trend held across all INTENTION, indicating that speech integration simplifies sketches by reducing the need for symbols and text.

5.4.5 Task Load. NASA-TLX results (Figure 6) show that SketchGPT had lower Mental ($p = .038$) and Temporal Demand ($p < .05$) than SpeechOnly and SketchOnly. SpeechOnly and SketchGPT outperformed SketchOnly in Performance ($p < .01$). No significant differences were found in other aspects. While SketchGPT reduced effort and enhanced efficiency, occasional misinterpretations led to moderate Frustration. The natural integration of sketching and speech facilitated intent expression, improving immersion and reducing Temporal Demand.

5.4.6 Modality Input Experience. A questionnaire in 7-point Likert scale was employed to assess users' experiences with different modalities of input, as shown in Figure 7. Results revealed that SketchGPT outperformed SketchOnly on all questions, with statistically significant differences ($p < .05$), except for Freedom of Expression (Q4, $p = .057$). Compared to SpeechOnly, SketchGPT received higher ratings in Ease of Use (Q1, $p = .024$) and Comfort and Naturalness (Q7, $p = .047$), highlighting its more user-friendly and natural multimodal interface. However, SpeechOnly slightly edged out SketchGPT in Correct Understanding of Intentions (Q4), likely due to its more detailed linguistic descriptions.

5.4.7 System Exploration Experience. We utilized a 7-point Likert scale questionnaire to evaluate participants' subjective experiences with SketchGPT while completing everyday tasks, with the results shown in Figure 8. Across all questions, average scores exceeded 5, indicating high satisfaction. Participants reported strong willingness to use SketchGPT (Q1, Q10) and found it easy to use (Q2, Q7). They expressed confidence in conveying their intentions (Q4, Q9) and appreciated the multimodal interaction (Q3). While understanding and execution (Q5, Q6, Q11) received slightly lower ratings, the human-in-the-loop optimization (Q8) was positively received. When compared to traditional methods, SketchGPT was considered more novel (Q14), convenient (Q12), and natural (Q15), though there was still room for improvement in outcome satisfaction (Q13).

5.5 Qualitative Results

We distill several key findings from the two study sessions and interviews.

Combination of sketch and speech enhances expression and understanding. Participants universally opted for multimodal interaction in Session 2, particularly for "*tasks that required shape or position description, or were more complex*" (EP10). While unimodal inputs were considered "*acceptable*" for simpler tasks (EP1, EP10), they were often insufficient for conveying a wide range of intentions. Specifically, speech alone was described as "*laborious for graphical contents or elements that were hard to refer to*" (EP5). Conversely, the primary challenge of the SketchOnly interface was the inherent ambiguity of drawings, as "*sketches could be easily misunderstood due to a lack of universal translation*" without verbal clarification (EP1). Both observations corroborate the findings from Session 1 (Figure 9). Notably, the qualitative preference for multimodal interaction did not always translate into statistically significant performance gains. For instance, no significant difference in intention closeness was observed between the SketchGPT and SpeechOnly interfaces for "modifying" and "move" INTENTIONS concerning "image content" and "single location" tasks. Similarly, for the "remove" INTENTION, input time did not differ significantly between the SketchGPT and SketchOnly interfaces. Despite these specific cases, there was a strong consensus among participants that "*combining sketches with speech is highly necessary*" (EP5, EP7).

Confirmation mechanism acts as a helpful tool to prevent errors. Typical failure cases in Session 2 primarily stemmed from the LLM's limitations in spatial reasoning, which was most evident in PowerPoint tasks. These limitations led to errors such as incorrect element positioning, sizing, and occlusion (Figure 10 (a)). For such errors, participants were often unwilling to correct the coordinates in the confirmation step, preferring instead to execute the command and perform manual adjustments afterward. In contrast, for Markdown tasks, the linear nature of the documents meant that precise LLM positioning was less critical, and most generated outputs met user expectations. However, Excel tasks again highlighted the challenges of LLM positioning. A typical error involved the system misinterpreting cell locations, for instance, operating on the wrong column when asked to perform a calculation (Figure 10 (b)). While this was correctable during confirmation, some participants chose to abandon the command and restart. Furthermore, participants who rated the system poorly (Q8, Figure 8) cited the overly technical descriptions in the confirmation step and the absence of an undo function post-execution as significant drawbacks. Despite these issues, participants generally valued the confirmation mechanism, particularly for preventing errors that would be difficult to rectify later. Features like image preview and regeneration were

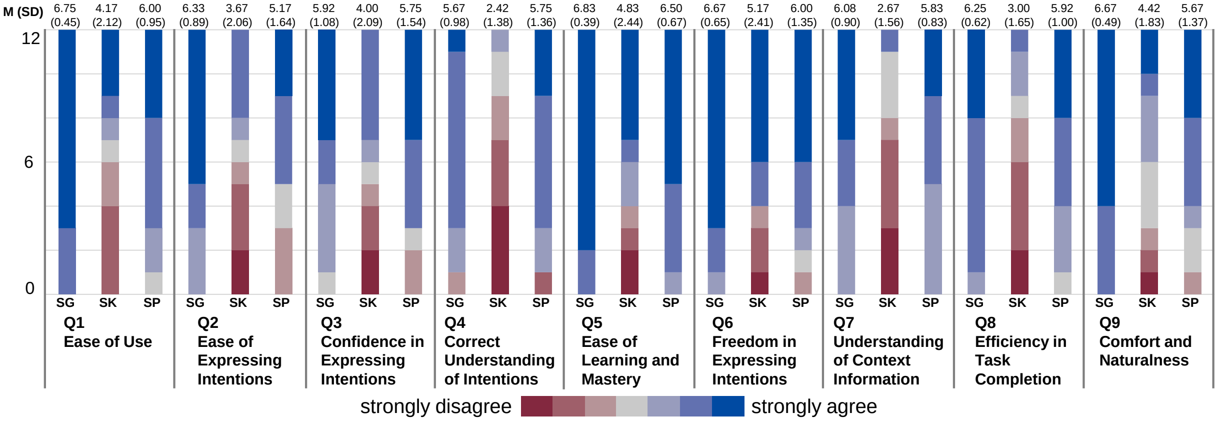


Figure 7: Questionnaire Results on the 7-Point Likert Scale from "Strongly Disagree" to "Strongly Agree" in Evaluation Study, Session 1. The x-axis represents different questions, covering various aspects such as system usability evaluation, intention expression, context understanding, and interaction experience. The y-axis shows the Likert scale ratings provided by participants for different systems across the various questions. SG, SK and SP respectively represent SketchGPT, SketchOnly and SpeechOnly.

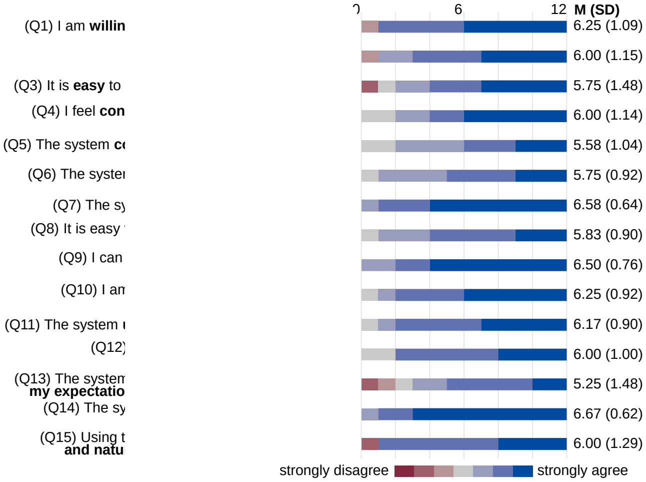
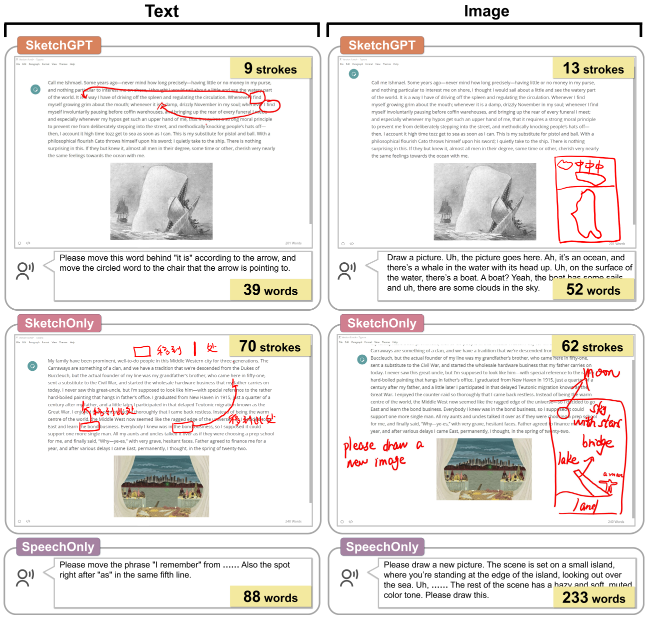


Figure 8: Questionnaire results on the 7-point Likert Scale from Evaluation Study, Session 2. Ratings are collected on a scale from "strongly disagree" to "strongly agree". The horizontal bar graph represents the distributions of these ratings.



Please move this word behind "it is" according to the arrow, and move the circled word to the chair that the arrow is pointing to.

Draw a picture. Uh, the picture goes here. Ah, it's an ocean, and there's a whale in the water with its head up. Uh, on the surface of the water, there's a boat. A boat? Yeah, the boat has some sails and uh, there are some clouds in the sky.

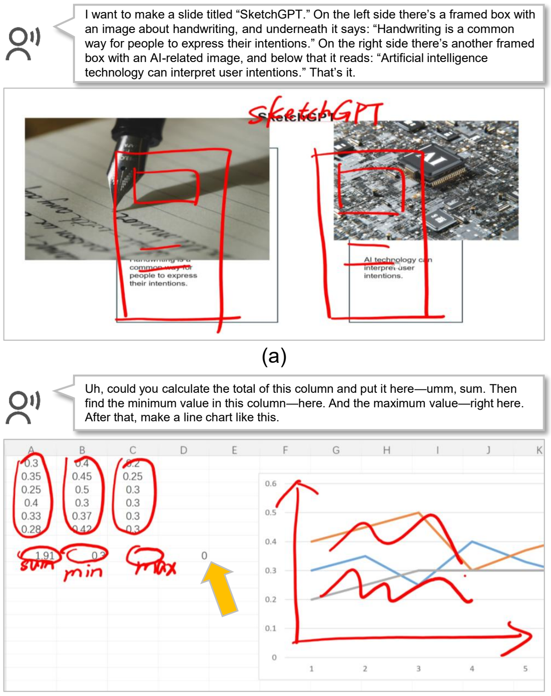
Please move the phrase "I remember" from ...... Also the spot right after "as" in the same fifth line.

Please draw a new picture. The scene is set on a small island, where you're standing at the edge of the island, looking out over the sea. Uh, ... The rest of the scene has a hazy and soft, muted color tone. Please draw this.

Figure 9: Examples of Text and Image-related tasks completed in Evaluation Study, Session 1, using the SketchGPT, SketchOnly, and SpeechOnly interfaces, along with corresponding stroke counts and speech-to-text word counts.

praised for their effectiveness. As EP8 noted, "*I tend to use the confirmation mechanism to preview generated images because correcting an incorrect image is difficult. For other items, I'm more willing to accept them all at once—since for other kinds of errors, like positional deviations or text adjustments, I might fix them manually or simply go through another iteration with SketchGPT.*"

Input modalities and response times present usability challenges. Regarding input modalities, speech was considered acceptable in private settings but "*inconvenient in workplace scenarios*" (EP7). Conversely, the necessity of a stylus was not perceived as a limitation, as participants found that "*touch input on touchscreens was also sufficient for sketching purposes*" (EP4). System latency emerged as another key usability concern, particularly for simple tasks. As EP2 noted, users weigh the time saved against the system's processing delay: "*For simple operations, I consider whether I should complete the task manually. But for complex and multi-step tasks, the wait is worthwhile*". This suggests that while latency is tolerated for complex commands, it can diminish the user experience for simple operations that could be performed faster manually.



I want to make a slide titled "SketchGPT." On the left side there's a framed box with an image about handwriting, and underneath it says: "Handwriting is a common way for people to express their intentions." On the right side there's another framed box with an AI-related image, and below that it reads: "Artificial intelligence technology can interpret user intentions." That's it.

SketchGPT
SketchGPT
Handwriting is a common way for people to express their intentions.
AI technology can interpret user intentions.

(a)

Uh, could you calculate the total of this column and put it here—umm, sum. Then find the minimum value in this column—here. And the maximum value—right here. After that, make a line chart like this.

A	B	C	D
0.3	0.4	0.2	
0.35	0.45	0.25	
0.25	0.5	0.3	
0.4	0.3	0.3	
0.33	0.37	0.3	
0.28	0.42	0.3	
1.91	0.3		0

(b)

Figure 10: Failure cases in Session 2 of the evaluation study. (a) Improper position and size of the image and box. (b) Mistaken table column.

6 Discussion

6.1 A Sketch Worth a Thousand Words

Sketches have long been used to enhance intent expression, such as conveying visualization goals [91] or supporting story creation [17]. Building on related work, SketchGPT enables freehand sketches combined with voice to offer a more open, natural, and efficient way to interact with LLMs. It supports conveying complex, high-level semantics that are difficult to express through text or predefined pen gestures.

In our studies, users under the SpeechOnly interface had to construct constrained contexts (e.g., specifying phrases, counting lines, or referencing occurrences) to disambiguate targets, increasing mental load and verbosity. In tasks like "add image," participants first described overall layouts before detailing objects, a process that was inherently demanding. With SketchGPT, participants naturally sketched targets while using referential language, reducing mental, temporal, and physical demands despite the multimodal input. For image description tasks, they primarily used quick, abstract sketches with verbal cues, focusing on meaning rather than precise detail, enabling more fluid and efficient intent expression.

6.2 Several Words Simplify a Detailed Sketch

In the SketchOnly interface, participants easily identified targets but struggled to specify operations. While textual annotations were effective, they were slower and increased complexity; symbolic representations reduced effort but introduced ambiguity due to inconsistent human interpretations. Without prior grounding [95], LLMs often misinterpret user-specific symbols, reducing accuracy and increasing cognitive load. In image tasks, abstract sketches made object categories difficult to convey, and adding text labels further increased temporal demand.

Speech input addresses these challenges by efficiently clarifying sketch ambiguities, reducing written input and cognitive burden. In SketchGPT, participants rarely wrote full sentences, relying instead on brief speech cues. This complementary use of speech and sketch improved both efficiency and accuracy, enabling natural and seamless intent expression.

6.3 Sketch + Speech Unleash Greater Potentials

Participants praised SketchGPT's potential across diverse tasks and devices. They found it "*especially helpful for creating presentations and documents, quickly turning ideas into drafts*" (EP1). Despite some challenges, such as element misplacement or incomplete data selection, they expressed interest in applying SketchGPT to coding, navigation, mind mapping, and 3D design. The integration of speech and sketch input was considered "*intuitive and easy to learn, even for elderly users and children*" (EP3).

The potential of SketchGPT's multimodal interaction extends across devices. On desktops and laptops without pen support, a mouse suffices for basic sketching that doesn't require precision. On tablets and smartphones, users can sketch via touch and speak through built-in microphones. Speech input was generally preferred over soft keyboards for its efficiency and lower effort, especially when expressing complex intentions.

Overall, the combination of sketch and speech promises to enhance user experience and efficiency when interacting with LLMs, supporting a wide range of user needs and applications.

6.4 Limitation

SketchGPT relies heavily on large language models. SketchGPT relies on MLLMs, so its reasoning depends on LLMs performance, which may lead to variable or inaccurate outcomes despite careful prompt design. These variations can affect user experience and evaluation consistency; however, participants generally expressed approval of the results. SketchGPT may involve higher latency and cost, but unlike potential lightweight methods requiring task-specific data, it works without such data and serves as a practical starting point for real-world scenarios and iterative optimization.

The study lacks more realistic and long-term observations. Our evaluations of SketchGPT were conducted in controlled settings and, while covering diverse scenarios (e.g., text, images, tables, canvases, and code), cannot capture all real-world cases. Although the formative study helped generalize interaction patterns, long-term real-world use remains to be explored. We envision SketchGPT as an open-ended system for interpreting and executing user intentions, but real-world contexts are highly complex and user habits vary widely, posing significant challenges for robust deployment. Observations of experts from diverse backgrounds are also valuable, as they may reveal different usage patterns and scenarios.

Modal Conflict Awareness and Feedback. In our study, we observed redundancy between the two modalities, with their information generally corroborating each other to enhance system stability. However, conflicts can arise due to user errors or LLM misinterpretation. The current system ignores such conflicts and defaults to one modality. Ideally, it should detect conflicts, provide feedback, and allow users to clarify their true intentions.

6.5 Future Work

Investigating other benefits like learnability and expressiveness. Traditional pen gesture systems rely on predefined vocabularies that require user learning. In contrast, SketchGPT supports free-form intentions, enhancing learnability. Future work could explore how user experience levels affect performance and whether new tasks or functions emerge as users gain expertise.

Exploring new possibilities for interactive iterations using sketches. The current confirmation phase relies on technical task descriptions that don't align well with users' workflow needs. Some participants (EP4, EP8, EP10) preferred directly editing and iterating on sketches rather than one-way inference, and requested reversible actions plus previews of execution outcomes. Future work could focus on developing more intuitive, flexible, and user-friendly methods for task confirmation and iteration.

Building a more direct interaction and response mode. Participants suggested that real-time understanding and immediate responses would "*enhance scenarios like document editing by increasing certainty*" (FP1). They desired "*clearer, more tangible feedback akin to direct manipulation*" to improve interaction experience (EP8). Combining direct manipulation on the interface with sketch input as an overlay could offer a promising hybrid interaction.

Establishing a benchmark for sketch intention inference evaluation. Our current evaluation relies on participant scoring designed to be as objective as possible, but fully objective assessment needs standardized benchmarks and clear metrics. Future work could establish these benchmarks to enable continuous technique iteration and explore multi-agent setups using LLMs as evaluators [7, 84], serving as a reflection mechanism.

Developing advanced approaches for understanding multimodal sketch interactions. SketchGPT relies heavily on LLMs' intention parsing, highlighting the need for more accurate and efficient methods. Prior work has addressed sketch semantic understanding (e.g., semantic classification [35, 105], handwriting recognition [19], scene segmentation [27]). Given that visual aids can enhance LLM reasoning [36], integrating these expert models could further improve domain-specific inference performance. Spatial understanding remains a common challenge for LLMs, making it crucial to improve spatial intent interpretation in future solutions.

7 Conclusion

This paper presents SketchGPT, a multimodal interaction paradigm that integrates sketches and speech to interact with LLMs directly on the system interface. Sketches offer richer semantic expression, particularly for spatial and graphical intentions that are difficult to convey through text alone. SketchGPT supports open-ended application scenarios within system contexts and effectively resolves multimodal ambiguities. By combining the complementary strengths of sketch and speech, the framework enhances interaction efficiency and fosters a more natural and fluid user experience.

For further investigation on the framework, we conducted a formative study using a Wizard-of-Oz design, bypassing current LLM limitations to observe potential user interaction patterns and preferences. Insights from this study informed the development of the SketchGPT prototype, which integrates a multimodal intention inference chain powered by MLLMs and employs a human-in-the-loop approach, enabling users to confirm task execution. Our evaluation study shows that comparing to single-modality interactions (sketch or speech), SketchGPT demonstrates superior interaction efficiency, reduced task load, and improved user experience.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62272447, 62332019, 62461160309, and 62473356), the Basic Research Program of Institute of Software Chinese Academy of Sciences (Grant No. ISCAS-JCMS-202404), CAS Major Project (Grant No. RCJJ-145-24-14), Science and Technology Innovation Key R&D Program of Chongqing (Grant No. CSTB2023TIAD-STX0027), Beijing Natural Science Foundation Haidian Original Innovation Joint Fund (Grant No. L242060, L232028, and L222008). We express our gratitude to the reviewers for their valuable comments and insights. Our sincere appreciation goes to all the participants who actively contributed to the user studies.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] A. Adler and R. Davis. 2007. Speech and sketching: an empirical study of multimodal interaction. In *Proceedings of the 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling* (Riverside, California) *(SBIM '07)*. Association for Computing Machinery, New York, NY, USA, 83–90. doi:10.1145/1384429.1384449

[3] Henry Alps. 2024. OpenManus: An open-source initiative to replicate the capabilities of the Manus AI agent. https://github.com/henryalps/OpenManus. Accessed: 2025-04-01.

[4] Christine Alvarado and Randall Davis. 2007. Resolving ambiguities to create a natural computer-based sketching environment. In *ACM SIGGRAPH 2007 Courses* (San Diego, California) *(SIGGRAPH '07)*. Association for Computing Machinery, New York, NY, USA, 16–es. doi:10.1145/1281500.1281527

[5] Richard Anderson, Crystal Hoyer, Craig Prince, Jonathan Su, Fred Videon, and Steve Wolfman. 2004. Speech, ink, and slides: the interaction of content channels. In *Proceedings of the 12th Annual ACM International Conference on Multimedia* (New York, NY, USA) *(MULTIMEDIA '04)*. Association for Computing Machinery, New York, NY, USA, 796–803. doi:10.1145/1027527.1027713

[6] Richard J. Anderson, Crystal Hoyer, Steven A. Wolfman, and Ruth Anderson. 2004. A study of digital ink in lecture presentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 567–574. doi:10.1145/985692.985764

[7] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=hSyW5go0v8

[8] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17682–17690.

[9] Richard A. Bolt. 1980. "Put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and*

Interactive Techniques (Seattle, Washington, USA) *(SIGGRAPH '80)*. Association for Computing Machinery, New York, NY, USA, 262–270. doi:10.1145/800250.807503

[10] Holly Branigan and Jamie Pearson. 2006. Alignment in human-computer interaction. *How people talk to computers, robots, and other artificial communication partners* (2006), 140–156.

[11] Runze Cai, Nuwan Janaka, Yang Chen, Lucia Wang, Shengdong Zhao, and Can Liu. 2024. PANDALens: Towards AI-Assisted In-Context Writing on OHMD During Travels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–24. doi:10.1145/3613904.3642320

[12] M Belén Calavia, Teresa Blanco, Ana Serrano, Anna Biedermann, and Roberto Casas. 2022. Think-Sketch-Create: Improving Creative Expression Through Sketching. In *International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing*. Springer, 1585–1597.

[13] Jorge D. Camba, Pedro Company, and Ferran Naya. 2022. Sketch-Based Modeling in Mechanical Engineering Design: Current Status and Opportunities. *Computer-Aided Design* 150 (Sept. 2022), 103283. doi:10.1016/j.cad.2022.103283

[14] Ozan Cetinaslan and Verónica Orvalho. 2018. Direct Manipulation of Blendshapes Using a Sketch-Based Interface. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology*. ACM, Poznań Poland, 1–10. doi:10.1145/3208806.3208811

[15] Ishan Chatterjee, Robert Xiao, and Chris Harrison. [n. d.]. Gaze+Gesture: Expressive, Precise and Targeted Free-Space Interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (Seattle Washington USA, 2015-11-09). ACM, 131–138. doi:10.1145/2818346.2820752

[16] Weihao Chen, Chun Yu, Huadong Wang, Zheng Wang, Lichen Yang, Yukun Wang, Weinan Shi, and Yuanchun Shi. 2023. From Gap to Synergy: Enhancing Contextual Understanding through Human-Machine Collaboration in Personalized Systems. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–15.

[17] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching Stories with Generative Pretrained Language Models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 209, 19 pages. doi:10.1145/3491102.3501819

[18] Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. 1997. QuickSet: multimodal interaction for distributed applications. In *Proceedings of the Fifth ACM International Conference on Multimedia* (Seattle, Washington, USA) *(MULTIMEDIA '97)*. Association for Computing Machinery, New York, NY, USA, 31–40. doi:10.1145/266180.266328

[19] Denis Coquenet, Clément Chatelain, and Thierry Paquet. 2023. End-to-End Handwritten Paragraph Text Recognition Using a Vertical Attention Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2023), 508–524. doi:10.1109/TPAMI.2022.3144899

[20] Joëlle Coutaz, Laurence Nigay, Daniel Salber, Ann Blandford, Jon May, and Richard M Young. 1995. Four easy pieces for assessing the usability of multimodal interaction: the CARE properties. In *Human—Computer Interaction: Interact'95*. Springer, 115–120.

[21] Hai Dang, Sven Goller, Florian Lehmann, and Daniel Buschek. 2023. Choice over control: How users write with large language models using diegetic and non-diegetic prompting. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.

[22] Lars Engeln, Nhat Long Le, Matthew McGinity, and Rainer Groh. 2021. Similarity Analysis of Visual Sketch-based Search for Sounds. In *Audio Mostly 2021*. ACM, virtual/Trento Italy, 101–108. doi:10.1145/3478384.3478423

[23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, Vol. 96. 226–231.

[24] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. [n. d.]. Orbits: Gaze Interaction for Smart Watches Using Smooth Pursuit Eye Movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte NC USA, 2015-11-05). ACM, 457–466. doi:10.1145/2807442.2807499

[25] Kenneth D Forbus, Ronald W Ferguson, and Jeffery M Usher. 2001. Towards a computational model of sketching. In *Proceedings of the 6th international conference on Intelligent user interfaces*. 77–83.

[26] Weiwei Gao, Kexin Du, Yujia Luo, Weinan Shi, Chun Yu, and Yuanchun Shi. 2024. EasyAsk: An In-App Contextual Tutorial Search Assistant for Older Adults with Voice and Touch Inputs. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 102 (Sept. 2024), 27 pages. doi:10.1145/3678516

[27] Ce Ge, Haifeng Sun, Yi-Zhe Song, Zhanyu Ma, and Jianxin Liao. 2022. Exploring Local Detail Perception for Scene Sketch Semantic Segmentation. *IEEE Transactions on Image Processing* 31 (2022), 1447–1461. doi:10.1109/TIP.2022.3142511

[28] Daniele Giunchi, Alejandro Sztrajman, Stuart James, and Anthony Steed. 2021. Mixing Modalities of 3D Sketching and Speech for Interactive Model Retrieval in Virtual Reality. In *Proceedings of the 2021 ACM International Conference on Interactive Media Experiences* (Virtual Event, USA) *(IMX '21)*. Association for Computing Machinery, New York, NY, USA, 144–155. doi:10.1145/3452918.3458806

[29] Google. 2024. Circle (or highlight or scribble) to Search. [Online]. https://blog.google/products/search/google-circle-to-search-android/ Accessed on April 14, 2024.

[30] Ziwei Gu, Ian Arawjo, Kenneth Li, Jonathan K. Kummerfeld, and Elena L. Glassman. 2024. An AI-Resilient Text Rendering Technique for Reading and Skimming Documents. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–22. doi:10.1145/3613904.3642699

[31] Benjamin Hagedorn and Jürgen Döllner. 2008. Sketch-Based Navigation in 3D Virtual Environments. In *Smart Graphics*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Andreas Butz, Brian Fisher, Antonio Krüger, Patrick Olivier, and Marc Christie (Eds.). Vol. 5166. Springer Berlin Heidelberg, Berlin, Heidelberg, 239–246. doi:10.1007/978-3-540-85412-8_23

[32] Jiyeon Han, Jimin Park, Jinyoung Huh, Uran Oh, Jaeyoung Do, and Daehee Kim. 2024. AscleAI: A LLM-based Clinical Note Management System for Enhancing Clinician Productivity. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–7. doi:10.1145/3613905.3650784

[33] Violet Yinuo Han, Tianyi Wang, Hyunsung Cho, Kashyap Todi, Ajoy Savio Fernandes, Andre Levi, Zheng Zhang, Tovi Grossman, Alexandra Ion, and Tanya R. Jonker. 2025. A Dynamic Bayesian Network Based Framework for Multimodal Context-Aware Interactions. In *Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI '25)*. Association for Computing Machinery, New York, NY, USA, 54–69. doi:10.1145/3708359.3712070

[34] A. G. Hauptmann. 1989. Speech and gestures for graphic image manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '89)*. Association for Computing Machinery, New York, NY, USA, 241–245. doi:10.1145/67449.67496

[35] Haoxiang Hu, Cangjun Gao, Yaokun Li, Xiaoming Deng, YuKun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. 2024. SpaceGTN: A Time-Agnostic Graph Transformer Network for Handwritten Diagram Recognition and Segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 3 (Mar. 2024), 2211–2219. doi:10.1609/aaai.v38i3.27994

[36] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models. *arXiv preprint arXiv:2406.09403* (2024).

[37] Junshi Huang, Si Liu, Junliang Xing, Tao Mei, and Shuicheng Yan. 2014. Circle & Search: Attribute-Aware Shoe Retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 1, Article 3 (sep 2014), 21 pages. doi:10.1145/2632165

[38] LangChain Inc. 2024. Langgraph. [Online]. https://langchain-ai.github.io/langgraph/ Accessed on March 20, 2024.

[39] Yuka Iwanaga, Masayoshi Tsuchinaga, Kosei Tanada, Yuji Nakamura, Takemitsu Mori, and Takashi Yamamoto. 2025. Sketch Interface for Teleoperation of Mobile Manipulator to Enable Intuitive and Intended Operation: A Proof of Concept. In *Proceedings of the 2025 ACM/IEEE International Conference on Human-Robot Interaction* (Melbourne, Australia) *(HRI '25)*. IEEE Press, 193–202.

[40] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. PromptMaker: Prompt-based Prototyping with Large Language Models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. ACM, New Orleans LA USA, 1–8. doi:10.1145/3491101.3503564

[41] Peiling Jiang, Jude Rayan, Steven P. Dow, and Haijun Xia. 2023. Graphologue: Exploring Large Language Model Responses with Interactive Diagrams. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. ACM, San Francisco CA USA, 1–20. doi:10.1145/3586183.3606737

[42] Michael Johnston, Philip R. Cohen, David McGee, Sharon L. Oviatt, James A. Pittman, and Ira Smith. 1997. Unification-based Multimodal Integration. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Madrid, Spain, 281–288. doi:10.3115/976909.979653

[43] Edward C Kaiser. 2007. Cross-domain matching for automatic tag extraction across redundant handwriting and speech events. In *Proceedings of the 2007 workshop on Tagging, mining and retrieval of human related activity information*. 55–62.

[44] Edward C Kaiser, Paulo Barthelmess, Candice Erdmann, and Phil Cohen. 2007. Multimodal redundancy across handwriting and speech during computer mediated human-human interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1009–1018.

[45] Soheil Kianzad, Yinan Li, and Hasti Seifi. 2025. Feel the Connection: Haptic Enhanced Interaction with an AI Agent. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 275, 8 pages.

doi:10.1145/3706599.3720173

[46] Hark-Joon Kim, Hayoung Kim, Seungho Chae, Jonghoon Seo, and Tack-Don Han. [n. d.]. AR Pen and Hand Gestures: A New Tool for Pen Drawings. *Augmented Reality* ([n. d.]).

[47] Taewan Kim, Donghoon Shin, Young-Ho Kim, and Hwajung Hong. 2024. DiaryMate: Understanding User Perceptions and Experience in Human-AI Collaboration for Personal Journaling. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–15. doi:10.1145/3613904.3642693

[48] Andy Kong, Karan Ahuja, Mayank Goel, and Chris Harrison. [n. d.]. EyeMU Interactions: Gaze + IMU Gestures on Mobile Devices. In *Proceedings of the 2021 International Conference on Multimodal Interaction* (Montréal QC Canada, 2021-10-18). ACM, 577–585. doi:10.1145/3462244.3479938

[49] Jaewook Lee, Jun Wang, Elizabeth Brown, Liam Chu, Sebastian S. Rodriguez, and Jon E. Froehlich. 2024. GazePointAR: A Context-Aware Multimodal Voice Assistant for Pronoun Disambiguation in Wearable Augmented Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–20. doi:10.1145/3613904.3642230

[50] lepo.co. 2024. WPF UI. [Online]. https://wpfui.lepo.co/ Accessed on July 20, 2024.

[51] Chunyuan Liao, François Guimbretière, and Ken Hinckley. 2005. PapierCraft: a command system for interactive paper. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA) *(UIST '05)*. Association for Computing Machinery, New York, NY, USA, 241–244. doi:10.1145/1095034.1095074

[52] Chunyuan Liao, François Guimbretière, Ken Hinckley, and Jim Hollan. 2008. Papiercraft: A gesture-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact.* 14, 4, Article 18 (Jan. 2008), 27 pages. doi:10.1145/1314683.1314686

[53] Haichuan Lin, Yilin Ye, Jiazhi Xia, and Wei Zeng. 2025. SketchFlex: Facilitating Spatial-Semantic Coherence in Text-to-Image Generation with Region-Based Sketches. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 546, 19 pages. doi:10.1145/3706598.3713801

[54] Susan Lin, Jeremy Warner, J.D. Zamfirescu-Pereira, Matthew G Lee, Sauhard Jain, Shanqing Cai, Piyawat Lertvittayakumjorn, Michael Xuelin Huang, Shumin Zhai, Bjoern Hartmann, and Can Liu. 2024. Rambler: Supporting Writing With Speech via LLM-Assisted Gist Manipulation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–19. doi:10.1145/3613904.3642217

[55] Michael Xieyang Liu, Advait Sarkar, Carina Negreanu, Benjamin Zorn, Jack Williams, Neil Toronto, and Andrew D. Gordon. 2023. "What It Wants Me To Say": Bridging the Abstraction Gap Between End-User Programmers and Code-Generating Large Language Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 598, 31 pages. doi:10.1145/3544548.3580817

[56] Shiyang Lu, Tao Mei, Jingdong Wang, Jian Zhang, Zhiyong Wang, and Shipeng Li. 2015. Exploratory Product Image Search With Circle-to-Search Interaction. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 7 (2015), 1190–1202. doi:10.1109/TCSVT.2014.2372272

[57] Cui-Xia Ma, Yong-Jin Liu, Hong-An Wang, Dong-Xing Teng, and Guo-Zhong Dai. 2012. Sketch-based annotation and visualization in video authoring. *IEEE Transactions on Multimedia* 14, 4 (2012), 1153–1165.

[58] Manus Team. 2024. Manus: a general AI agent that bridges minds and actions. https://manus.im/. Accessed: 2025-04-08.

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

[60] Microsoft. 2024. Microsoft Azure Speech to Text Service. [Online]. https://learn.microsoft.com/en-us/azure/ai-services/speech-service/speech-to-text Accessed on March 14, 2024.

[61] Darius Miniotas, Oleg Špakov, Ivan Tugoy, and I. Scott MacKenzie. [n. d.]. Speech-Augmented Eye Gaze Interaction with Small Closely Spaced Targets. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications - ETRA '06* (San Diego, California, 2006). ACM Press, 67. doi:10.1145/1117309.1117345

[62] Capucine Nghiem, Adrien Bousseau, Mark Sypesteyn, Jan Willem Hoftijzer, Maneesh Agrawala, and Theophanis Tsandilas. 2024. STIVi: Turning Perspective Sketching Videos into Interactive Tutorials. In *Proceedings of the 50th Graphics Interface Conference* (Halifax, NS, Canada) *(GI '24)*. Association for Computing Machinery, New York, NY, USA, Article 16, 13 pages. doi:10.1145/3670947.3670969

[63] Runliang Niu, Jindong Li, Shiqi Wang, Yali Fu, Xiyu Hu, Xueyuan Leng, He Kong, Yi Chang, and Qi Wang. 2024. ScreenAgent: A Vision Language Model-driven Computer Control Agent. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 6433–6441. doi:10.24963/ijcai.2024/711 Main Track.

[64] OpenAI. 2024. OpenAI Platform. [Online]. https://platform.openai.com/docs/overview Accessed on August 20, 2024.

[65] Sharon Oviatt. 1999. Mutual disambiguation of recognition errors in a multimodel architecture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 576–583. doi:10.1145/302979.303163

[66] Sharon Oviatt. 2000. Taming recognition errors with a multimodal interface. *Commun. ACM* 43, 9 (Sept. 2000), 45–51. doi:10.1145/348941.348979

[67] Sharon Oviatt and Philip Cohen. 2000. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Commun. ACM* 43, 3 (March 2000), 45–53. doi:10.1145/330534.330538

[68] Kaiyue Pang, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. 2020. Solving Mixed-Modal Jigsaw Puzzle for Fine-Grained Sketch-Based Image Retrieval. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 10344–10352. doi:10.1109/CVPR42600.2020.01036

[69] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. [n. d.]. Gaze-Touch: Combining Gaze with Multi-Touch for Interaction on the Same Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu Hawaii USA, 2014-10-05). ACM, 509–518. doi:10.1145/2642918.2647397

[70] B. Plimmer, J. Grundy, J. Hosking, and R. Priest. 2006. Inking in the IDE: Experiences with Pen-based Design and Annotatio. In *Visual Languages and Human-Centric Computing (VL/HCC'06)*. 111–115. doi:10.1109/VLHCC.2006.28

[71] Aditya kumar Purohit, Aditya Upadhyaya, and Adrian Holzer. 2023. Chatgpt in healthcare: Exploring ai chatbot for spontaneous word retrieval in aphasia. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*. 1–5.

[72] Fei-wei Qin, Shu-ming Gao, Xiao-ling Yang, Jing Bai, and Qu-hong Zhao. 2017. A Sketch-Based Semantic Retrieval Approach for 3D CAD Models. *Applied Mathematics-A Journal of Chinese Universities* 32, 1 (March 2017), 27–52. doi:10.1007/s11766-017-3450-3

[73] Shwetha Rajaram, Hemant Bhaskar Surale, Codie McConkey, Carine Rognon, Hrim Mehta, Michael Glueck, and Christopher Collins. 2025. Gesture and Audio-Haptic Guidance Techniques to Direct Conversations with Intelligent Voice Interfaces. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1133, 20 pages. doi:10.1145/3706598.3714310

[74] Niroop Channa Rajashekar, Yeo Eun Shin, Yuan Pu, Sunny Chung, Kisung You, Mauro Giuffre, Colleen E Chan, Theo Saarinen, Allen Hsiao, Jasjeet Sekhon, Ambrose H Wong, Leigh V Evans, Rene F. Kizilcec, Loren Laine, Terika Mccall, and Dennis Shung. 2024. Human-Algorithmic Interaction Using a Large Language Model-Augmented Artificial Intelligence Clinical Decision Support System. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–20. doi:10.1145/3613904.3642024

[75] Sebastián Ramírez. 2024. FastAPI. [Online]. https://fastapi.tiangolo.com/ Accessed on March 12, 2024.

[76] Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 314, 7 pages. doi:10.1145/3411763.3451760

[77] Laurel D. Riek. 2012. Wizard of Oz studies in HRI: a systematic review and new reporting guidelines. *J. Hum.-Robot Interact.* 1, 1 (jul 2012), 119–136. doi:10.5898/JHRI.1.1.Riek

[78] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[79] Karl Toby Rosenberg, Rubaiat Habib Kazi, Li-Yi Wei, Haijun Xia, and Ken Perlin. 2024. DrawTalking: Towards Building Interactive Worlds by Sketching and Speaking. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 113, 8 pages. doi:10.1145/3613905.3651089

[80] Sigurdur Gauti Samuelsson and Matthias Book. 2020. Eliciting Sketched Expressions of Command Intentions in an IDE. *Proc. ACM Hum.-Comput. Interact.* 4, ISS, Article 200 (Nov. 2020), 25 pages. doi:10.1145/3427328

[81] Patsorn Sangkloy, Wittawat Jitkrittum, Diyi Yang, and James Hays. 2022. A sketch is worth a thousand words: Image retrieval with text and sketch. In *European Conference on Computer Vision*. Springer, 251–267.

[82] Vishnu Sarukkai, Lu Yuan, Mia Tang, Maneesh Agrawala, and Kayvon Fatahalian. 2024. Block and Detail: Scaffolding Sketch-to-Image Generation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 33, 13 pages. doi:10.1145/3654777.3676444

[83] Chuhan Shi, Yicheng Hu, Shenan Wang, Shuai Ma, Chengbo Zheng, Xiaojuan Ma, and Qiong Luo. 2023. RetroLens: A Human-AI Collaborative System for Multi-step Retrosynthetic Route Planning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.

[84] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 377, 19 pages.

[85] Ben Shneiderman. 2020. Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy. *International Journal of Human–Computer Interaction* 36, 6 (2020), 495–504. doi:10.1080/10447318.2020.1741118 arXiv:https://doi.org/10.1080/10447318.2020.1741118

[86] Katie A Siek, Yvonne Rogers, and Kay H Connelly. 2005. Fat finger worries: how older and younger users physically interact with PDAs. In *Human-Computer Interaction-INTERACT 2005: IFIP TC13 International Conference, Rome, Italy, September 12-16, 2005. Proceedings 10*. Springer, 267–280.

[87] Sruti Srinivasa Ragavan, Zhitao Hou, Yun Wang, Andrew D Gordon, Haidong Zhang, and Dongmei Zhang. 2022. Gridbook: Natural language formulas for the spreadsheet grid. In *27th international conference on intelligent user interfaces*. 345–368.

[88] Arjun Srinivasan and Vidya Setlur. 2021. Snowy: Recommending utterances for conversational visual analysis. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 864–880.

[89] Al Sweigart. 2024. PyAutoGUI: Cross-platform GUI Automation for Human Beings. [Online]. https://github.com/asweigart/pyautogui/tree/master Accessed on September 1, 2024.

[90] CAMEL-AI Team. 2024. Owl: Optimized Workforce Learning for General Multi-Agent Assistance in Real-World Task Automation. https://github.com/camel-ai/owl. Accessed: 2025-04-01.

[91] Zhongwei Teng, Quchen Fu, Jules White, and Douglas C. Schmidt. 2021. Sketch2Vis: Generating Data Visualizations from Hand-drawn Sketches with Deep Learning. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 853–858. doi:10.1109/ICMLA52953.2021.00141

[92] Anh Truong, Peggy Chi, David Salesin, Irfan Essa, and Maneesh Agrawala. 2021. Automatic Generation of Two-Level Hierarchical Tutorials from Instructional Makeup Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 108, 16 pages. doi:10.1145/3411764.3445721

[93] Emmanuel Turquin, Jamie Wither, Laurence Boissieux, Marie-paule Cani, and John Hughes. 2007. A Sketch-Based Interface for Clothing Virtual Characters. *IEEE Computer Graphics and Applications* 27, 1 (Jan. 2007), 72–81. doi:10.1109/MCG.2007.1

[94] Unsplash. 2024. Beautiful Free Images & Pictures | Unsplash. [Online]. https://unsplash.com Accessed on September 1, 2024.

[95] Priyan Vaithilingam, Ian Arawjo, and Elena L. Glassman. 2024. Imagining a Future of Designing with AI: Dynamic Grounding, Constructive Negotiation, and Sustainable Motivation. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 289–300. doi:10.1145/3643834.3661525

[96] Ilse M Verstijnen, Cees van Leeuwen, Gabriela Goldschmidt, Ronald Hamel, and JM Hennessey. 1998. Sketching and creative discovery. *Design studies* 19, 4 (1998), 519–546.

[97] Fengjie Wang, Yanna Lin, Leni Yang, Haotian Li, Mingyang Gu, Min Zhu, and Huamin Qu. 2024. OutlineSpark: Igniting AI-powered Presentation Slides Creation from Computational Notebooks through Outlines. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–16. doi:10.1145/3613904.3642865

[98] Yunlong Wang, Shuyuan Shen, and Brian Y Lim. 2023. RePrompt: Automatic Prompt Editing to Refine AI-Generative Art Towards Precise Expressions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–29. doi:10.1145/3544548.3581402

[99] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. 2024. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–21.

[100] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[101] Nadir Weibel, Adriana Ispas, Beat Signer, and Moira C. Norrie. 2008. Paperproof: a paper-digital proof-editing system. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (Florence, Italy) *(CHI EA '08)*. Association for Computing Machinery, New York, NY, USA, 2349–2354. doi:10.1145/1358628.1358682

[102] Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. 2022. PromptChainer: Chaining Large Language Model Prompts through Visual Programming. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. ACM, New Orleans LA USA, 1–10. doi:10.1145/3491101.3519729

[103] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–22. doi:10.1145/3491102.3517582

[104] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences* 68, 2 (2025), 121101.

[105] Yu-Ting Yang, Yan-Ming Zhang, Xiao-Long Yun, Fei Yin, and Cheng-Lin Liu. 2022. CASIA-onDo: A New Database for Online Handwritten Document Analysis. In *Pattern Recognition*, Christian Wallraven, Qingshan Liu, and Hajime Nagahara (Eds.). Springer International Publishing, Cham, 174–188.

[106] Ryan Yen, Jian Zhao, and Daniel Vogel. 2024. Code Shaping: Iterative Code Editing with Free-form Sketching. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST Adjunct '24)*. Association for Computing Machinery, New York, NY, USA, Article 101, 3 pages. doi:10.1145/3672539.3686324

[107] Ryan Yen, Jian Zhao, and Daniel Vogel. 2025. Code Shaping: Iterative Code Editing with Free-form AI-Interpreted Sketching. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 872, 17 pages. doi:10.1145/3706598.3713822

[108] Dongwook Yoon, Nicholas Chen, François Guimbretière, and Abigail Sellen. 2014. RichReview: blending ink, speech, and gesture to support collaborative document review. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 481–490.

[109] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 437, 21 pages. doi:10.1145/3544548.3581388

[110] Shane W Zamora and Eyrún A Eyjólfsdóttir. [n. d.]. CircuitBoard: Sketch-Based Circuit Design and Analysis. ([n. d.]).

[111] Xin Zeng, Xiaoyu Wang, Tengxiang Zhang, Chun Yu, Shengdong Zhao, and Yiqiang Chen. 2024. GestureGPT: Toward Zero-Shot Free-Form Hand Gesture Understanding with Large Language Model Agents. *Proc. ACM Hum.-Comput. Interact.* 8, ISS, Article 545 (Oct. 2024), 38 pages. doi:10.1145/3698145

[112] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 3813–3824. doi:10.1109/ICCV51070.2023.00355

[113] Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems* 37 (2024), 132208–132237.

[114] Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. 2024. Multimodal Chain-of-Thought Reasoning in Language Models. *Transactions on Machine Learning Research* (2024). https://openreview.net/forum?id=y1pPWFVfvR

[115] Zhaohui Zhang, Haichao Zhu, and Qian Zhang. 2020. ARSketch: Sketch-Based User Interface for Augmented Reality Glasses. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, Seattle WA USA, 825–833. doi:10.1145/3394171.3413633

[116] Weiqin Zu, Wenbin Song, Ruiqing Chen, Ze Guo, Fanglei Sun, Zheng Tian, Wei Pan, and Jun Wang. 2024. Language and Sketching: An LLM-driven Interactive Multimodal Multitask Robot Navigation Framework. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 1019–1025. doi:10.1109/ICRA57147.2024.10611462

[117] Ran Zuo, Haoxiang Hu, Xiaoming Deng, Cangjun Gao, Zhengming Zhang, Yu-Kun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. 2024. SceneDiff: Generative Scene-Level Image Retrieval with Text and Sketch Using Diffusion Models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 1825–1833. doi:10.24963/ijcai.2024/202 Main Track.

[118] Gustavo Zurita, Nelson Baloian, and Felipe Baytelman. 2008. A collaborative face-to-face design support system based on sketching and gesturing. *Advanced Engineering Informatics* 22, 3 (2008), 340–349.