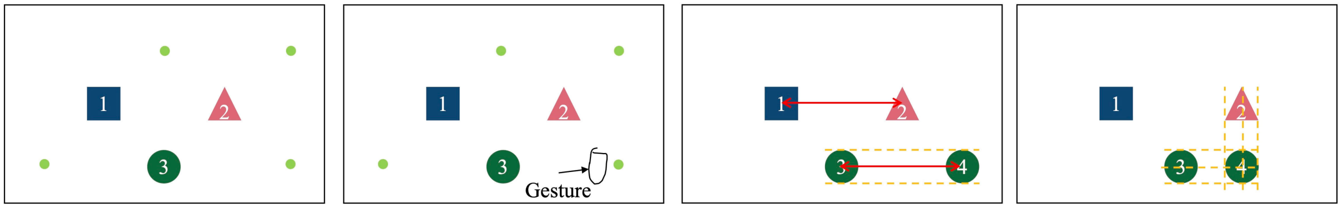
A Human-Computer Collaborative Editing Tool for Conceptual Diagrams

Lihang Pan
plh18@mails.tsinghua.edu.cn
Department of Computer science and Technology,
Tsinghua University
Beijing, China

Chun Yu*
chunyu@mail.tsinghua.edu.cn
Department of Computer science and Technology,
Tsinghua University
Beijing, China

Zhe He
hez19@mails.tsinghua.edu.cn
Department of Computer science and Technology,
Tsinghua University
Beijing, China

Yuanchun Shi
shiyc@tsinghua.edu.cn
Department of Computer science and Technology,
Tsinghua University
Beijing, China
Qinghai University
Xining, China



(a)

(b) “Create an circle here”

(c)

(d)

Figure 1: How the user edits the diagram in a multimodal collaborative manner. (a) There are three elements on the canvas before the command. The green dots indicate the predicted positions for the next element. (b) The user command contains a casual gesture and a vague instruction. Note that the user does not specify a precise position, color, or size for the new element. (c) The best solution calculated by the system. (d) The user can switch to another solution manually.

ABSTRACT

Editing (e.g., editing conceptual diagrams) is a typical office task that requires numerous tedious GUI operations, resulting in poor interaction efficiency and user experience, especially on mobile devices. In this paper, we present a new type of human-computer collaborative editing tool (CET) that enables accurate and efficient editing with little interaction effort. CET divides the task into two parts, and the human and the computer focus on their respective specialties: the human describes high-level editing goals with multimodal commands, while the computer calculates, recommends, and performs detailed operations. We conducted a formative study (N = 16) to determine the concrete task division and implemented the tool on Android devices for the specific tasks of editing concept diagrams. The user study (N = 24 + 20) showed that it increased diagram editing speed by 32.75% compared with existing state-of-the-art commercial tools and led to better editing results and user experience.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; **Ubiquitous and mobile computing systems and tools**.

KEYWORDS

conceptual diagram, natural content editing, multi-modal interaction, human-computer collaboration

ACM Reference Format:
Lihang Pan, Chun Yu, Zhe He, and Yuanchun Shi. 2023. A Human-Computer Collaborative Editing Tool for Conceptual Diagrams. In *CHI '23: The ACM CHI Conference on Human Factors in Computing Systems, April 23–28 2023, Hamburg, Germany*. ACM, New York, NY, USA, 29 pages. https://doi.org/10.1145/3544548.3580676

1 INTRODUCTION

Editing (e.g., editing conceptual diagrams [53, 77]) is a typical office task. Whatever the editing device is (e.g., PC, tablet, or big screen), it is currently accomplished with GUI applications (e.g., Microsoft PowerPoint), where users waste tremendous time and effort on

*indicates the corresponding author.

CHI '23, April 23–28, 2023, Hamburg, Germany
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3580676

complex and tedious GUI operations. For example, when editing diagrams, users must search for the desired function on a complex interface and constantly drag and drop an element to control its position precisely. Smartphones enable users to meet job obligations at flexible times and locations. Users can fully use the fragmented time to boost productivity and leisure time [90, 98] while minimizing the impact on family duties [5, 26, 104, 105] when work is inevitable. The aforementioned problem is even more prominent when users edit on their mobile devices [19, 65, 91, 96, 98].

The primary reason for this problem is that after determining the editing goals, users have to convert them into complicated GUI operations and manually conduct the procedures. Artificial intelligence (AI) may help or even replace users in these operations, collaborating with them to accomplish tasks with greater efficiency and quality. This mechanism has been applied in design and creation activities [61, 62, 113] but hardly to office tasks.

To reduce the interaction overhead while ensuring the accuracy of content editing, we propose a novel type of human-computer collaborative editing tool (CET). The most significant feature of CET is that it supports task division and collaboration between the human and the computer. As shown in Figure 1-b, the user describes high-level editing effects via multi-modal commands (inaccurate gestures and vague verbal instructions) instead of the precise and tedious operations required by traditional GUI applications. The tool collaborates with the user to guarantee accurate editing results: CET automatically calculates candidate solutions that satisfy user commands and recommends them to the user in order (Figure 1-c&d). Besides, it predicts and displays the user's possible subsequent behaviors (e.g., the green dots in Figure 1-a).

We selected conceptual diagram editing as a typical example of various content editing tasks and conducted two user studies. The first was a formative study where we determined the task division between the human and the computer: what information the human is willing to provide and how the computer supplements the remaining. The experimental results indicated that users specified only the important features of the editing content with vague inputs, and the assistants complemented operation details based on prior knowledge and content-related posterior knowledge. We accordingly proposed the functionality design, system design, and collaborative interaction design of CET and completed implementation, named SGDiag. Fourteen Figures in this paper (Figure 4 & 6 - 18) are created with SGDiag. The second study evaluated the functionality of SGDiag. The experiment results showed that our tool saved 32.75% of the time and scored 21.89% higher in editing results than the existing state-of-the-art commercial system (Microsoft PowerPoint for Android). In addition, users showed significant preferences for our tool.

Our contributions are two folds:

(1) We present a formative study strategy for determining the task division between humans and computers by simultaneously observing the behaviors of end users and human assistants.
(2) We propose a new class of human-computer collaborative editing tools that introduce AI into existing editing tasks. We demonstrated how such tools assisted end users in completing mobile diagram editing activities with great experience, efficiency, and quality.

2 RELATED WORK

2.1 Human-computer Collaborative Editing Systems

Traditional human-computer collaborative systems provide optional references to help the editing process. A common application domain is patient notes, where collaborative tools automatically display patient information [112, 120] and disease encyclopedia [110] based on physicians' inputs. Besides, such systems are widely used for other tasks, such as drawing [24], brainstorming [111], game editing [38], video creation [54, 55], and storytelling [101, 117].

With the rapid development of neural networks, human-computer collaborative systems have changed their roles from assistants to creators. A typical example is AI-assisted painting, where the user only draws part of the image, and the deep learning algorithms automatically generate a complete picture [28, 34, 39, 51, 74, 84]. Similarly, Ryan Louie proposed a music editing system [75]: the author manually writes some notes and sets some parameters, and the AI automatically composes the remaining parts. In addition, such methods are fully exploited in creative writing [20, 21, 35].

The multimodal collaborative tool proposed in this paper has two significant differences from the existing collaborative systems. First, our tool supports multimodal interactions such as casual gestures and vague utterances, whereas existing systems require rigid GUI interactions. Second, our tool is a compromise of the two existing categories. Our tool directly participates in editing rather than merely providing optional hints. Besides, it significantly differs from modern collaborative creators that complete the generation process independently from end to end. Instead, it follows user instructions step by step and collaborates with the user to resolve the vagueness of the multimodal commands.

2.2 Multimodal Interactions

Multimodal interactions reduce interaction burden, improve interaction efficiency and user satisfaction [1, 88, 121], and are a hot research topic in human-computer interaction. Researchers proposed many multimodal interaction techniques [33, 60, 80, 118, 119] and applied them in different interaction tasks such as system control [2, 11, 45, 122], visualization [69, 100], mobile interaction [29, 30, 48, 115], information retrieval [4] , and interface styling [57]. Combining speech and gesture is a common multimodal scheme [1, 11, 50, 86, 87, 113] that excels in semantic information expression and spatial location specification [22, 121]. SGDiag employs this scheme due to its advantages.

A typical practice of multimodal editing tools is adding voice commands as optional shortcuts to existing functions [52, 122]. VoiceCuts [58] adds a speech modality to Adobe Photoshop for creative experts. However, most interactions are still in traditional GUIs. Another example is PixelTone [67], which enables multimodal image editing on small and portable devices. However, users must specify every detail in the commands because the system cannot infer absent parameters from the editing contents. Besides, machine learning researchers have proposed many multimodal systems for generating (instead of editing) different contents, such as texts [14], images [16, 44], music [18], and videos [73]. Nevertheless, they mainly focus on deep learning models instead of the interaction processes. Users cannot control how the models generate the contents. In addition, multimodal editing tools play an essential role in accessibility, supporting the disabled in drawing [41, 43], cursor control [10, 23, 40], text input [116], and web page design [92]. However, for ordinary people, these systems impose an additional interaction burden.

Dissolving the vagueness of user commands is an important topic in interaction research. The fat finger problem leads to ambiguity in gesture interaction, to which researchers have proposed many solutions [42, 99, 108]. Speech interaction faces a similar situation, which has not been fully explored [7, 89]. In this paper, we investigate the vagueness of multimodal commands in natural editing and its solutions through a bidirectional user behavior observation experiment (Study 1) and design a collaborative system according to the findings.

2.3 Conceptual Diagram Editing Tools

A conceptual Diagram *"provides a graphical overview of conceptual models—the relationship between concrete and abstract entities"* [77]. It has various applications in different fields, such as scientific writing [70], software development [17], and education [3, 109].

Existing diagram editing tools can be divided into two categories: those based on direct manipulation and those based on programming languages [77]. General drawing tools (e.g., Microsoft PowerPoint and Noyon [97]) support "WYSIWYG" (What You See Is What You Get), but users need to constantly specify different attributes of elements (e.g., positions and colors) with numerous tedious operations [37]. Programming language-based diagram editing tools [13, 94] support rapid code-based prototyping and automatic optimization of the diagram layouts [46, 47, 66]. However, these tools have a steep learning curve [78] and are, therefore, rarely used by non-experts.

The diagram editing tool in this paper combines the features of both types of editing tools. We still adopt a WYSIWYG design while replacing the traditional GUI interface with a multimodal natural interaction interface, reducing the learning cost and extending applicable scenarios. The users only describe high-level editing effects, and the editing tools calculate and optimize the diagrams.

2.4 Editing on Mobile Devices

Editing on mobile devices is prevalent nowadays. Compared to traditional office desktop devices, mobile devices break spatial and temporal limits and support better work-life balance when work is inevitable [26, 68, 85, 90, 98]. For example, work-related smartphone usage "makes an otherwise impossible trip possible" [104]. In addition to work-related usage, users edit photos and videos on their mobile devices and upload them directly to social applications [59, 63] to share their lives anytime and anywhere. In addition, mobile devices are cheap to afford and easy to use, which makes them serve a wide range of people. For example, people in developing countries (e.g., villages in Africa [9] and India [32]) and children [79] prefer mobile devices to create stories.

Previous work explored mobile editing of presentations [49, 56], images [31, 67, 76], videos [15, 106], stories [93], and games [8]. Editing diagrams is rarely explored in the academic community. Many commercial general-purpose drawing applications that support editing diagrams have mobile phone versions, such as Microsoft PowerPoint, Keynote, WPS office, Zoho Show, and Adobe Illustrator Draw. However, these applications follow the interaction design of their desktop versions and are not optimized for mobile devices.

3 STUDY 1: OBSERVING THE USERS' AND THE ASSISTANTS' TASK DIVISION IN EDITING DIAGRAMS

The goal of Study 1 is to determine the task division in the diagram editing process: what information the human is willing to provide and how the computer supplements the remaining. We observe the process of the multimodal collaborative editing of conceptual diagrams. Our expected results mainly focus on two aspects:

(1) The natural expression of editing intentions and the gaps between the expression and traditional GUI interactions. The natural expression indicates the human's responsibility in the collaborative editing task, while bridging the gaps is the responsibility of the computer;
(2) How the assistant bridges the gaps. This helps us determine how the computer accomplishes its functionalities.

3.1 Study Design: A formative Study

Table 1: Tasks in Study 1. Please refer to the Appendix (Section A.1 and Figure 23) for details of the tasks.

Task Id	Discipline	Topics
1	Ecology	An overview of the carbon cycle
2	Chemistry	DNA and the base pairs
3	Computer	The framework of Android system

Two participants take part in the study simultaneously, one as the end user and the other as the human assistant. The end user finishes three tasks (as shown in Table 1) from a blank canvas in random order. Before each task, we provide detailed natural language descriptions of the target diagram (Section A.1 in the Appendix). The end user gives the human assistant multimodal (speech & gesture) instructions. We do not impose any restrictions on user commands and require users to express their intentions naturally, i.e., "to express their ideas in the same way they think about them" [82] The human assistant edits the diagram in Microsoft PowerPoint according to the user's instructions and is required to improve the presentation of the diagram. In contrast to Wizard-of-Oz (WoZ) studies, we observe the actions of human assistants and end users.

Figure 2 shows the details of the study apparatus. Table 2 illustrates the communication between the participants and the experimenter. We ask the human assistant to think aloud about why they edit in that way. To avoid human-human elicitation, the experimenter does not talk to the end user or the human assistant. The human assistant does not provide any verbal feedback to the end user. We use QuickTime Player to simultaneously record the laptop

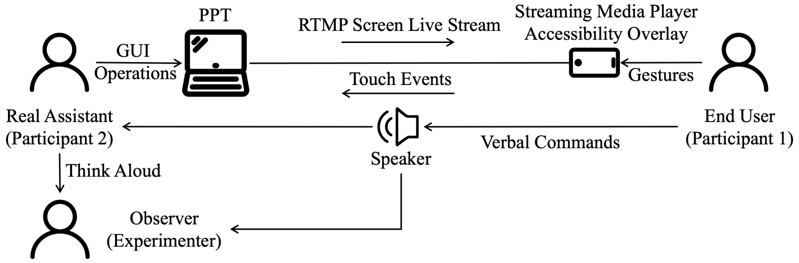


Figure 2: An overview of the apparatus. The human assistant edits diagrams with Microsoft PowerPoint on the laptop. The screen is captured and forwarded to the end user's smartphone via RTMP (Real-Time Messaging Protocol) live stream so that the end user can see the editing results. The end user gives verbal commands and performs gestures on the smartphone. Voice commands are played through a speaker to the human assistant. The touch events are sent to the laptop and rendered on a transparent overlay.

screen (including the actions of the human assistant and the user gestures) and the conversations among the three subjects as a video file.

3.2 Participants

We recruited 16 participants (10 males and 6 females, aged 21-35). All of them had experience of editing diagrams with Microsoft PowerPoint before the experiment. The participants were randomly paired up to complete the tasks. The study lasted 60 minutes.

3.3 Results

3.3.1 Behaviors of the End Users and the Gap to GUI Operations. We collected 349 commands. Users referred to existing diagram elements or configured attributes with speeches or gestures (Figure 3). 45.85% (160) of the commands contain multimodal interactions, as shown in Table 3.

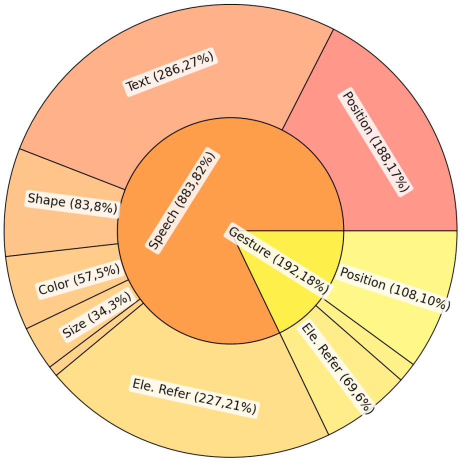


Figure 3: The proportion of different modalities and their functions in the commands. Note that one command may contain multimodal data and several attributes. We do not mark out the sectors less than 10 degrees (Speech/Font & Gesture/Shape).

All commands contain one or more attributes, such as position, size, shape, color, and text. Users neglected many unimportant attributes in natural behaviors, as opposed to explicitly specifying all attribute values while using traditional GUI tools. Furthermore, they did not always assign specific values to the unignored attributes. We summarize the three phenomena listed below:

Position was the most important attribute, and unimportant attributes were neglected. Users merely cared about important features rather than specifying every detail as in GUI interactions. As shown in Figure 3, the position attributes (i.e., the x and y coordinates) were of most interest to the users. Some unimportant attributes (e.g., fonts and colors) accounted for only a small percentage, meaning they were unimportant and always ignored. By explicitly specifying the position attributes, users controlled the macroscopic representation of the diagram (also called lateral transformations [36, 78]) without wasting time or effort on other details (vertical transformations [36, 78]).

Positions were indefinite with vague ranges. The end users never directly specified precise values for element positions as they do in GUI interactions. Instead, they described vague ranges with gestures or by comparing with positions of other elements. The expected position values were indefinite. We summarized three kinds of commands related to element positions, the details and examples of which are shown in Table 4:

(1) Describing the comparisons among position attributes (45.89%). The expected value was indefinite as the user restricted the element within a range instead of a point.
(2) Setting position attributes with gestures (46.75%). The expected position values were unclear due to the fat finger problem.
(3) Describing the equivalence among position attributes (7.36%). The human assistant could calculate the value accurately in this rare case.

End users commented that the relationships among the position attributes were important. They never calculated the exact values of element positions (the x and y coordinates) in their minds; instead, they imagined the diagram layout and expressed the relationships via natural language and gestures. In contrast, they specified precise values for non-position attributes such as color

Table 2: The communication between participants and the experimenter.

		Speaker		
		Experimenter	Human Assistant	End User
Listener	Experimenter	-	Think Aloud	No Communication
	Human Assistant	No Communication	-	Multimodal Instructions
	End User	No Communication	No Communication	-

Table 3: Different kinds of multimodal commands. The texts in quotation marks are the speeches and the italics are the gestures.

Type	Role of Speeches	Role of Gestures	Count	Example
1+1	Configure attributes	Refer to elements	36	"create a new circle to the right of this" *draw a line under an existing element*
	Configure attributes	Configure attributes	77	"create a blue square here" *draw a square on the canvas*
1+2	Configure attributes Refer to elements	Configure attributes	17	"copy the circle here and change its text to Animal" *tap on the canvas*
	Configure attributes Refer to elements	Refer to elements	19	"move it to the right of the black square" *tap on an existing element*
Others	-	-	11	-

Table 4: How users specified positions for elements. The texts in quotation marks are the speeches and the italics are the gestures.

Type		Count	Example
Comparison	Direction (Left/Above/...)	61	"create a circle to the right of the square"
	Distance (Close to/Apart from)	14	"move it closer to the black square" *tap on an element*
	Between	31	"draw a circle between the red and the blue"
Gesture	-	108	"move it here" *tap on the canvas*
Equivalence	Equal	14	"create C so that the distance between A and B equals that between B and C"
	Middle	3	"create C in the middle of A and B"

and text. Non-position attributes were rarely associated with each other, and it was easier to figure out their values because they were intuitive and discrete.

Conflict Commands for positions. The position relationships in the current commands might conflict with those in history commands (74 in 349, 21.20%). However, the end users did not designate how to solve the conflict. Figure 4 is a typical example. The user’s new command "move B here" conflicts with the history command "create C so that the distance between A and B equals that between B and C." There are at least three possible solutions:

(1) Move B and C together while keeping A unmoved. This solution satisfies both commands but overturns the command that determines the position of B.
(2) Move A, B, and C as a whole. This solution satisfies both commands but overturns the commands that determine the positions of A and B.
(3) Keep A and C unmoved and move B only. This solution only satisfies the current command.

3.3.2 How Human Assistants Bridged the Gap. We analyzed the human assistants’ think-aloud and their GUI operations to summarize how they bridged the gaps between the natural instructions and the GUI interactions.

Absent/Indefinite attribute complementary with prior knowledge. The prior knowledge reflected the human assistants’ perceptual preferences and complemented half of the missing or unclear attributes, as shown in Figure 5. A typical example is fine-tuning the color. The human assistant did not set the color to **#0000FF** when the user changed an element to blue. Instead, the assistant set it to **#24B6DB** and said, "Sky blue is a kind of blue that looks good." The assistant used prior knowledge mainly in the early stages of editing diagrams when there were few existing elements as references. The average index ratio of prior knowledge[1] is 38.46%.

Absent/Indefinite attribute complementary with posterior knowledge. The posterior knowledge reflected the local features related with diagram contents, which could be generalized to other elements. Posterior knowledge accounted for half of the absent

[1] $\frac{index\ of\ the\ command\ complemented\ with\ prior\ knowledge}{length\ of\ the\ command\ list}$

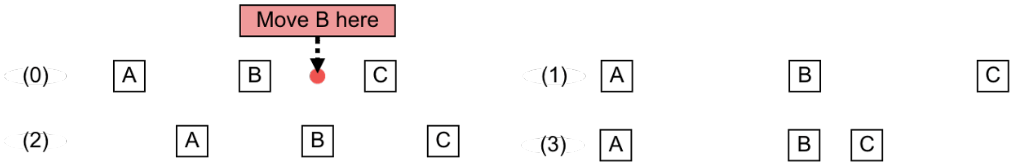


Figure 4: An example of conflict command (0) and its possible solutions (1 - 3).

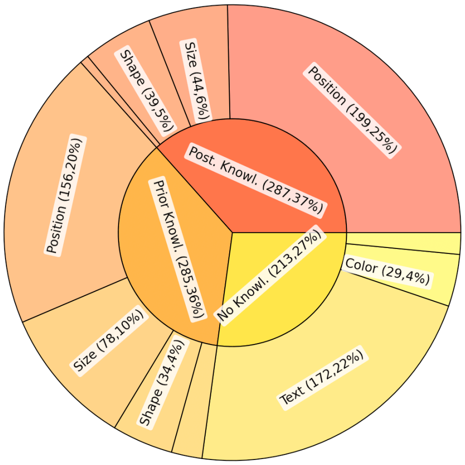


Figure 5: The proportion of different ways of the human assistants calculating element attribute values. We do not mark out the sectors less than 10 degrees.

or unclear attributes (Figure 5), indicating that it is necessary to build a content-aware editing tool. For example, the human assistant complemented absent attributes of a new element by copying from similar elements (the color of B and C in Figure 6). Besides, they generalized relationships from other elements to calculate the position attributes, as shown in Figure 6. The human assistant determined the position of element C by reusing the relationship between A and B, i.e., copying the distance (both horizontally and vertically) between A and B. The average index ratio of posterior knowledge[2] is 55.81%.

Existing relationships maintenance: associated modification of attributes. After users gave conflicting commands, human assistants spontaneously modified element attributes to maintain existing relationships both specified by the users and inferred by the assistants. Many elements that did not appear in the commands were also modified due to their indirect association.

There might be different ways to maintain the relations, as shown in Figure 4. In most cases, the assistants determined their solutions by intuition instead of logical reasoning. They also mentioned several reasons listed below:

(1) The assistants chose the solution that required minimal GUI operations to reduce their interaction burden ("lazy assistants") and hoped that the users would further adjust the diagram if not satisfied with the results.
(2) The assistants chose to reserve the most important relations. There were various standards to measure the importance, such as the elements in the relations and the creation time of the relations.
(3) The assistants chose the best solution to convey the concepts based on the diagram semantics. However, they acknowledged that it was difficult to infer the semantics from only parts of the diagram. It would help if they knew the study tasks.

4 THE COLLABORATIVE NATURAL EDITING TOOL AND ITS INSTANCE FOR DIAGRAMS

To reduce the interaction overhead while guaranteeing the accuracy of content editing, we propose a new kind of editing tool: the collaborative editing tool (CET). It combines the advantages of existing editing tools and human-computer collaborative assistants and extends them to multimodal interactions. The editing tool divides the task into two parts, with the human and the computer focusing on their specialties: the end user describes high-level editing effects, while the editing tool complements, recommends, and conducts detailed editing operations.

We implement CET for editing conceptual diagrams and call the implementation result SGDiag (**Diag**ramming tool with **S**peeches and **G**estures). In this section, we will introduce the design of CET with details of SGDiag as examples. The design of CET is composed of the following three aspects:

(1) Functional design, i.e. the part of the task division that the computer is responsible for.
(2) System design. We propose a system architecture to implement the functional design.
(3) Collaborative interaction design. We design a collaborative interaction process to support proper system execution and accurate user editing.

4.1 Functional Design

According to Study 1, the end user vaguely provides incomplete information throughout the human-computer collaborative editing process. CET should incorporate the following three functionalities to "autocomplete" the user's intentions:

Complementing absent/indefinite contents. To lessen the interaction burden, users tend to shorten their commands. Unimportant or apparent content is frequently overlooked. Furthermore, the commands are typically non-deterministic. On the one hand, the user's behaviors are not always accurate. Natural language has a limited ability to describe spatial locations, and the "fat finger problem" makes it difficult for users to refer to positions on the canvas precisely via gestures. On the other hand, a major portion of

[2] $\frac{\textit{index of the command complemented with posterior knowledge}}{\textit{length of the command list}}$

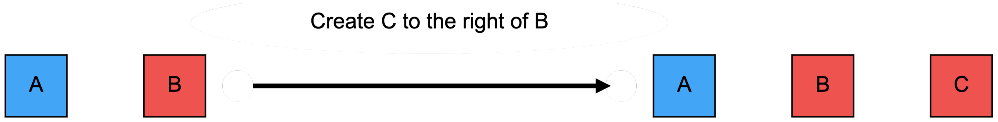


Figure 6: The assistant copies the color and shape from B to C. The horizontal distance between B and C is the same as that between A and B. The vertical coordinate of C is the same as that of B.

the user's consideration is centered on the relationships rather than the exact values. After describing high-level relationships, the user does not always care about the execution details. CET should complement the absent and indefinite contents in the user commands, transforming the natural behaviors into executable operations.

Dissolving conflicts between commands. The editing contents are unavoidably modified by the user, resulting in contradictions between current and historical instructions. Because different portions of the contents are tightly connected, the user may need to perform many associated modifications to retain certain crucial relationships. Describing these modifications is repetitive and time-consuming. The user delivers the most critical and fundamental modification commands in human-computer collaboration, and the editing system completes the associated modifications. CET must be able to resolve command conflicts and select the best editing option intelligently.

Providing multiple candidate solutions. The inherent ambiguity of user commands leads to the existence of multiple solutions. Even a human assistant cannot always determine the user's intentions. CET needs to recommend multiple reasonable options based on confidence to ensure interaction accuracy.

4.2 System Design

To implement the aforementioned functionalities, CET models the editing contents and goes through the following four stages: command parsing, solution generation, solution recommendation, and intention prediction, as shown in Figure 7. Solution generation implements the first and the second functionalities, while solution recommendation implements the third.

4.2.1 Content Modeling. CET stores the content and the relationships among different parts. It utilizes content modeling as the content-related posterior knowledge and maintains its correctness. In SGDiag, we calculate a matrix to model the positional relationships among diagram elements, i.e., the geometric topology, as discussed in Section 5.1.

4.2.2 Command Parsing. Command parsing consists of verbal instruction parsing and cross-modal instruction alignment, intending to transform user input into a form that a computer can process. Understanding the literal meaning of the commands is a hot topic in natural language processing and is beyond the scope of this paper. SGDiag applies a traditional context-free grammar method with hand-written rules. Researchers can collect and annotate interaction data to train a more sophisticated model.

4.2.3 Solution Generation. CET calculates several candidate solutions based on the literal meaning of the commands. Solution generation can be divided into the following two steps:

(1) **Complementing absent/indefinite attributes** (Functionality 1). CET complements the absent or unclear content in the instructions based on predefined prior knowledge and content-related posterior knowledge.
(2) **Resolving conflict commands** (Functionality 2). CET searches possible strategies to resolve conflicts, and adjusts contents accordingly.

4.2.4 Solution Recommendation. CET recommends solutions in order according to their scores (Functionality 3). The score calculation follows two principles:

(1) The use of prior and posterior knowledge should be reasonable. The definition of reasonableness may vary in different editing tasks.
(2) The solution tries best to preserve history editing behaviors and results, avoiding extensive modifications.

In addition, CET reveals the insights and logic behind the solutions to assist users in selecting from different options. In SGDiag, the topological relationships of elements are difficult to determine visually; therefore, we visualize them with auxiliary lines, as shown in Figure 11-left.

4.2.5 Intention Prediction. CET infers subsequent intentions from the interaction history and the existing editing contents. The intuitions are as follows:

(1) There exist bidirectional dependencies of user behaviors in the interaction sequence (Figure 9-right).
(2) The subsequent interactions tend to be coherent with the posterior knowledge derived from the current contents (Figure 9-left).

Note that CET is an **editing** rather than a **generative** tool; the editing task is not fed into the system as prior knowledge. Therefore, CET cannot accurately predict subsequent intentions. Prompts for the predicted intentions should be innocuous so that the inaccuracy will not cause any side-effect.

4.3 Collaborative Interaction Design

The interaction of CET contains three phases: pre-command, intra-command, and post-command, as shown in Figure 8.

Pre-command Interactions. CET predicts user intention and gives prompts for subsequent instructions to improve the interaction efficiency. As shown in Figure 9-left, SGDiag infers positions for new elements based on the current diagram and marks them on the canvas as references for the user's gestures. In addition, it supports batched element adjustment (Figure 9-right) based on the user action history.

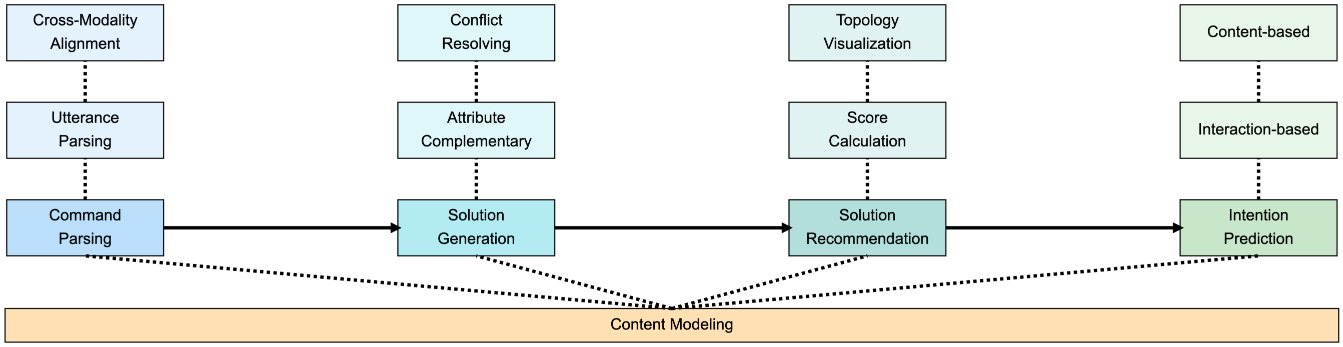


Figure 7: An overview of the system architecture.

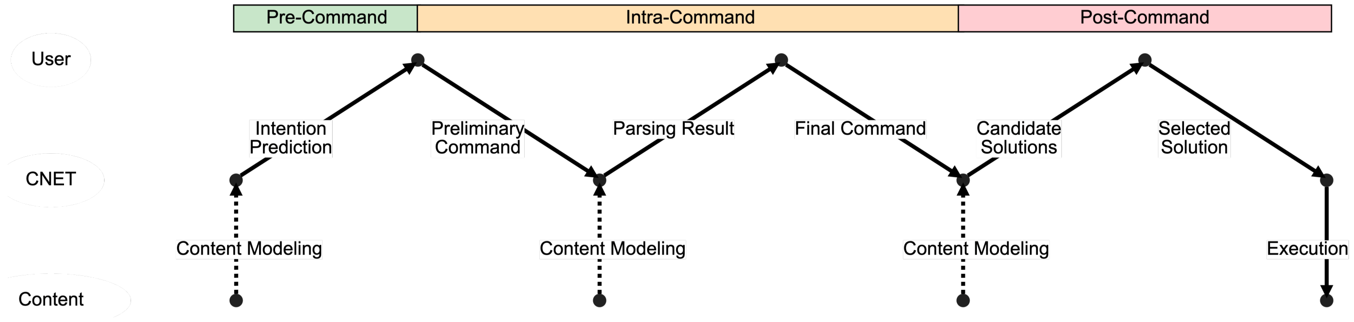
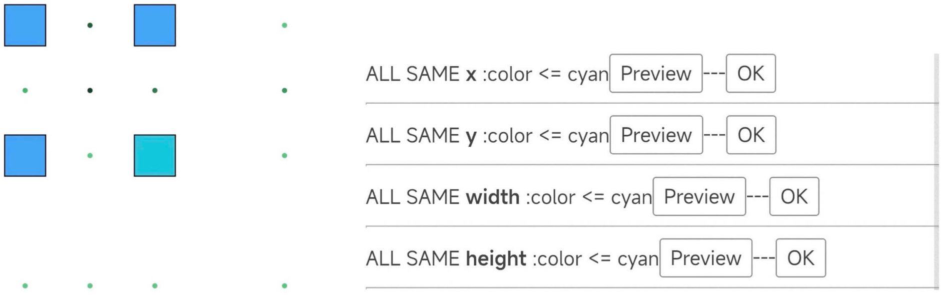


Figure 8: An overview of the interaction design. CET collaborates with users based on content modeling. Users may switch to another solution after selecting one from the candidates, which is not included in the figure.


ALL SAME **x** :color <= cyan

ALL SAME **y** :color <= cyan

ALL SAME **width** :color <= cyan

ALL SAME **height** :color <= cyan

Figure 9: Pre-command prompts in SGDiag. (left) Predicted positions (green dots) for the next element. (right) Predicted subsequent interactions. The last operation is to set the rectangle in the lower right corner to cyan.

Intra-command Interactions. CET displays user utterances and the parsing results while receiving the user's multimodal instructions. Users can correct them manually to ensure that their ideas are conveyed correctly. As shown in Figure 10, SGDiag visualizes touch trajectories, automatic speech recognition (ASR) results, and the referred element.

Post-command Interactions. CET proposes several possible solutions according to its functionalities. Meanwhile, it shows the internal logic of each solution to help users choose from the candidates or inspire the next instructions. SGDiag provides an overview for each possible solution and displays topological relations (such as equidistance and alignment) among elements, as shown in Figure 11.

5 DETAILS OF SGDIAG

5.1 Content Modelling

5.1.1 Attributes and Relations. SGDiag records both the attribute values and their relationships. Table 5 illustrates the supported attributes and examples of their values. In addition, SGDiag models the topology of the diagram as equations among position attributes (x & y in Table 5), which is the most significant feature of the diagram. For example, Equation 1 & 2 in Figure 12-b indicates that

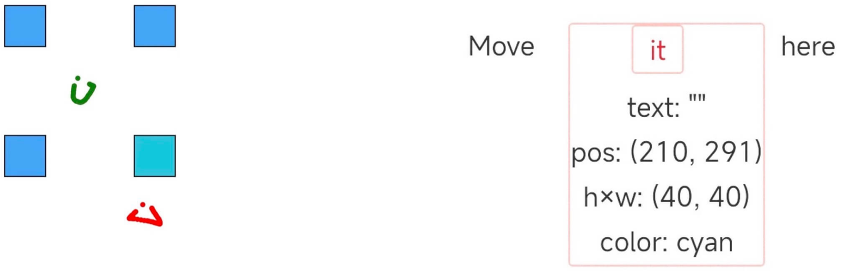


Figure 10: Visualization of the user input in SGDiag. (left) Touch trajectories. The red one is the first trace (corresponding to "it") and the green one is the second (corresponding to "here"). (right) ASR result & the element reference in the command

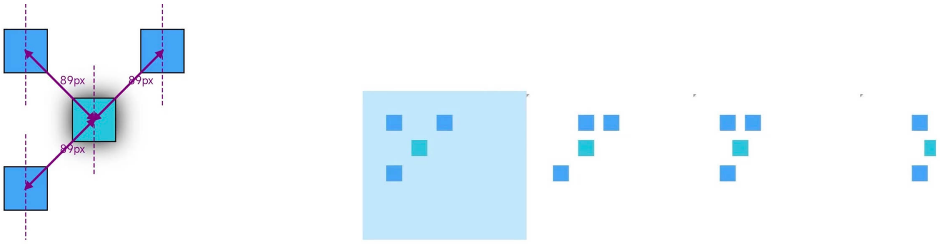


Figure 11: Visualization of candidates. (left) The visualization of the topology relationships. (right) Overviews for the candidate solutions.

Table 5: Attributes and example values for shapes (cells in white) and lines (cells in light blue). Please refer to Table 11 in the Appendix for the actual values of *red-0* and *blue-0*.

Element ID	Type	x	y	width	height	Fill Color	Text
1	Rect	100 px	100 px	20 px	30 px	red-0	"Output"
2	Circle	200 px	200 px	40 px	50 px	blue-0	"Input"
Element ID	**Type**	**Source ID**	**Target ID**	**Dashed**	**Source Arrow**	**Target Arrow**	**Text**
3	Line	2	1	False	False	True	"Process"

the user wants to set the coordinate to (10, 10) explicitly; Equation 5 indicates that the horizontal distance between A and C equals the vertical distance between them. We introduce *non-attribute variables* (c1 and c2 in Figure 12-c) to represent the constant terms[3] in the equations and extend the attribute coefficient matrix (colored blue) into an augmented matrix (blue + red). We formulate the diagram topology as

$$\begin{bmatrix} R_a & R_c \end{bmatrix} * \begin{bmatrix} \mathbf{a} \\ \mathbf{c} \end{bmatrix} = \mathbf{0} \qquad (1)$$

, where R_a, R_c, $\mathbf{a}$, and $\mathbf{c}$ denote the attribute coefficient matrix (the blue matrix in Figure 12-c), the non-attribute coefficient matrix (the red matrix), the attributes (the blue vector), and the non-attribute variables (the red vector), respectively.

5.1.2 Model Maintenance. The topology matrix is modified according to the user's commands. For example, SGDiag adds two columns to the matrix when the user creates a new element. The columns correspond to the horizontal and vertical coordinates, respectively. Besides, it will be updated automatically in the following three cases:

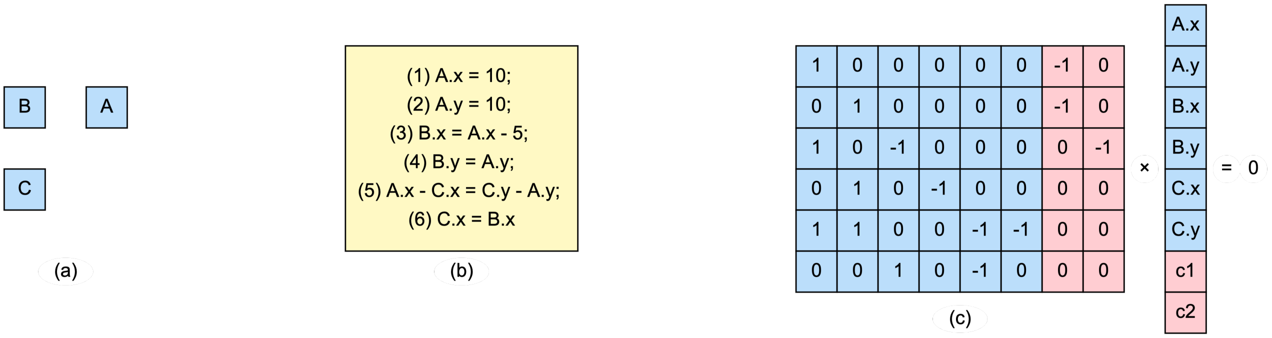
After the user deleting an element. SGDiag recalculates the diagram topology while keeping the position values unchanged when the user deletes an element from the diagram. The recalculation is one step in Gaussian elimination: reducing coefficients of the deleted element's attributes to zero with a sequence of elementary row operations, as shown in Figure 13.

Underdetermined. In this case, the rank of the attribute coefficient (R_a) matrix is not full, meaning attributes outnumber the equations, and some of the attributes are not well-constrained. We transform the attribute coefficient matrix into the reduced row echelon form via Gauss–Jordan elimination and classify the attributes into two categories: (1) constrained attributes that correspond to the first nonzero coefficients in the rows of the echelon form and (2) free attributes, i.e., the others.

SGDiag adds an equation for each free attribute. If the attribute value has already been determined[4], we will create a non-attribute

[3] Although we call these numbers **constant** terms, they may change during Solution Generation (5.3.2)

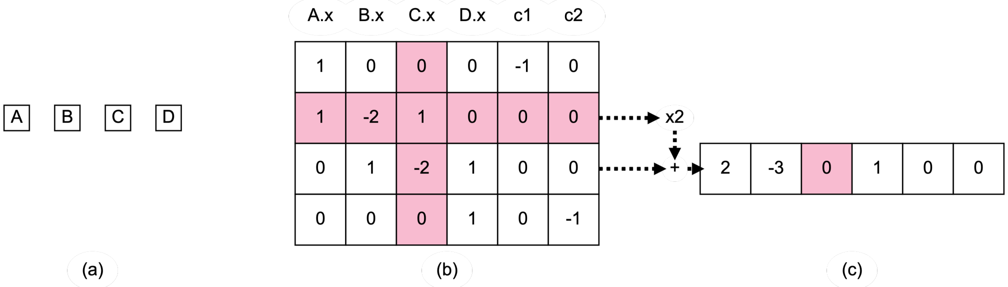
[4] The only possible case is that the value is calculated by optimizing Equation 2 in Section 5.3.2.



(1) A.x = 10;
(2) A.y = 10;
(3) B.x = A.x - 5;
(4) B.y = A.y;
(5) A.x - C.x = C.y - A.y;
(6) C.x = B.x

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} A.x \\ A.y \\ B.x \\ B.y \\ C.x \\ C.y \\ c1 \\ c2 \end{bmatrix} = 0$$

Figure 12: An overview of how SGDiag models the topology relationships. (a) Three elements on the canvas; (b) the equations that determine their values; (c) the topology matrix, each row of which corresponds to an equation in (b). Note that SGDiag stores the matrix instead of the equations in its implementation. We use equations in writing because they are easy to understand.



$$\begin{array}{cccccc} A.x & B.x & C.x & D.x & c1 & c2 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{array}$$

$$\begin{array}{cccccc} 2 & -3 & 0 & 1 & 0 & 0 \end{array}$$

Figure 13: An overview of how SGDiag recalculates the topology matrix after the user deletes the element C. (a) The original four elements on the canvas. (b) The original matrix. All elements have the same vertical coordinates, which are omitted in the figure. SGDiag will delete cells in red. (c) The coefficients of *C.x* are reduced to zero with row operations.

variable for this value, and the new equation is *attribute equals the new variable*. If the value is not determined, SGDiag will add an equation with prior or posterior knowledge, as discussed in Section 5.3.1.

Overdetermined. In this case, the rank of the attribute coefficient matrix (R_a) is less than that of the augmented matrix ($\begin{bmatrix} R_a & R_c \end{bmatrix}$), indicating that the equations conflict with each other. SGDiag will delete equations until the diagram is no longer overdetermined. We will discuss the detailed algorithms in Section 5.3.2.

5.2 Command Parsing

5.2.1 Supported Instructions. SGDiag supports three kinds of instructions: element creation, attribute modification, and element deletion. Users can refer to existing elements and canvas positions and specify attribute constraints in the commands.

Element reference. Users can refer to existing elements using gestural or verbal designators. The red trajectory in Figure 10-left is a typical gestural designator: the user marks the element on the canvas with a gesture. Verbal designators contain descriptions of the element attributes (type, color, and text). For example, the user says, "change the color of **the red circle** to blue."

Position reference. Users can refer to positions on the canvas with gestures. We calculate the centers of the trajectories as the referred positions. Besides, they can describe the position relationships between existing elements and the target position, such as "create a new red rectangle **to the left of** this element."

Position reference is always ambiguous due to the fat finger problem [108] and the limited user ability of verbally specifying spatial information. SGDiag infers several possible interpretations (Section 5.3.1) and uses the referred positions to filter away unreasonable ones (Section 5.4).

Attribute constraints. Users can specify constraints that the attribute values should satisfy:

(1) Equivalences among attributes, such as "set the color of the circle **to** red." Equivalences among position attributes will be added to the topology matrix.
(2) Comparisons among attributes, such as "make the horizontal distance between A and C **greater than** the vertical distance between C and D."
(3) Relationships between positions and attributes, which set elements to the referred positions. For example, the user can drag the element to the target position, which equals the command "move it here."

SGDiag calculates the attribute values according to the constraints of the first kind and adds the constraints to the topology matrix before generating solutions (Section 5.3) if they contain position attributes. SGDiag utilizes other kinds of constraints as the temporary prior knowledge (Section 5.3.1) and, more importantly, as the criteria of solution recommendation (Section 5.4).

Other control commands. SGDiag supports the following commands to help users control the editing process better:

(1) Undo/Redo. Users can revoke their commands when they are unsatisfied with the editing results.
(2) Explicitly forcing some elements unchanged. SGDiag will not modify the specified element when resolving conflicts (Section 5.3.2).
(3) Attribute increment and decrement. Users can modify the attributes based on their current values, such as "make its color darker" and "make its width smaller."
(4) Copy. Users can copy an existing element to a desired position.

5.2.2 Parsing Commands. We parse verbal commands with context-free grammar (CFG) [6]. To align gestures and verbal commands, SGDiag requires that users say *it* (element reference with gestures) or *here* (position reference with gestures) when performing gestures.

5.3 Solution Generation

SGDiag interprets the vagueness of user commands in different ways and generates several candidate solutions. Solution generation can be divided into two phases: complementing attributes and resolving conflicts.

5.3.1 Complementing Absent/Indefinite Attributes. The absent or indefinite attributes include (1) non-positional attributes that do not appear in the user command, (2) free position attributes discussed in Section 5.1.2, and (3) attributes to be modified without new equivalence constraints.

Using predefined prior knowledge. SGDiag sets attributes to default values at the beginning of the editing task when there is little posterior knowledge. Table 6 shows the default value for every attribute. We predefine a color scheme, as shown in Table 11 in the Appendix. If the user specifies a range for an attribute (attribute constraints Type 2 & 3), SGDiag treats the center of the range as temporary prior knowledge.

Using content-related posterior knowledge. For non-position attributes, SGDiag searches for the most similar elements in the diagram and copies the attribute values of the search results. For example, in Figure 14, SGDiag copies the shapes, sizes, and colors from existing elements (A & B) to the created elements (C & D).

For position attributes, SGDiag generates new equations by shuffling coefficients in rows of the attribute coefficient matrix. Figure 15 demonstrates an example. We apply the following heuristic rules to prune the generation process so that the number of generated equations is reasonable:

(1) Attributes other than the missing attribute in the new equation must appear in the original equation.
(2) Coefficients shuffling is restricted to attributes of the same coordinate.
(3) At least half of the coefficients do not change.

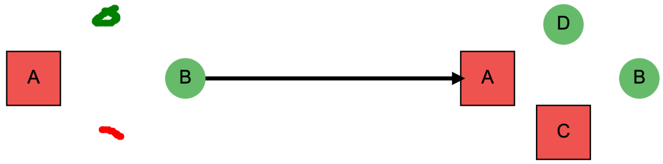


Figure 14: SGDiag infers the shapes, sizes, and colors for the new elements. The user commands are "create a new rectangle here (the red trace)" and "create a new green element here (the green trace)."

5.3.2 Resolving Conflicts. The basic idea is to override old commands with new commands. The overriding is quite simple for non-position attributes since we do not model relationships among them. However, position attributes connect with each other in the topology equations. Modification of one attribute can result in the modification of others. We propose three principles for fixing topological conflicts:

(1) Minimize the changes of current position values, i.e.,$\min_{\hat{\mathbf{a}},\hat{\mathbf{c}}}(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix} - \begin{bmatrix}\hat{\mathbf{a}}\\ \hat{\mathbf{c}}\end{bmatrix})$, where $\hat{\mathbf{a}}$ and $\hat{\mathbf{c}}$ are position attributes and non-position variables after fixing the conflicts, respectively. $\|\cdot\|$ indicates the Euclidean norm. The definitions of other notations are the same as those in Formula 1.
(2) Minimize the changes in current position relationships, i.e., $\min_{\hat{\mathbf{a}},\hat{\mathbf{c}}}(\begin{bmatrix}R_a & R_c\end{bmatrix} * \begin{bmatrix}\hat{\mathbf{a}}\\ \hat{\mathbf{c}}\end{bmatrix})$.
(3) Minimize the number of changed values and relationships, i.e., $\min_{\hat{\mathbf{a}},\hat{\mathbf{c}}}(\#_{nonzero}(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix} - \begin{bmatrix}\hat{\mathbf{a}}\\ \hat{\mathbf{c}}\end{bmatrix}) + \#_{nonzero}(\begin{bmatrix}R_a & R_c\end{bmatrix}\begin{bmatrix}\hat{\mathbf{a}}\\ \hat{\mathbf{c}}\end{bmatrix}))$.

The final optimization goal (adding Principle 1 & 2 together) is minimizing the following expression:

$$L(\hat{X}) \;:=\; A * \hat{X} - X \;:=\; \begin{bmatrix}R_a & R_c\\ I & \mathbf{0}\\ \mathbf{0} & I\end{bmatrix} * \begin{bmatrix}\hat{\mathbf{a}}\\ \hat{\mathbf{c}}\end{bmatrix} - \begin{bmatrix}\mathbf{0}\\ \mathbf{a}\\ \mathbf{c}\end{bmatrix} \tag{2}$$

, where I denotes the identity matrix and := denotes *define as.* To fulfill the third principle, we apply an A* search algorithm where we delete one row in each search step from A until $A * \hat{X} = X$ can be solved. We use the minimal value of Equation 2 as the heuristic function. The search finishes when we get 20 different solutions. The remaining rows in each solution indicate the remaining equations. Figure 16 demonstrates several search results.

5.4 Solution Recommendation

SGDiag filters out solutions according to the second & third types of attribute constraints. It sorts and recommends the remaining solutions to the users.

5.4.1 Solution Score Calculation. SGDiag calculates the scores as *score* = *Reasonableness* − *Error* and sorts the solutions accordingly. The top-1 solution is executed automatically.

Reasonableness measures whether the added attributes (Section 5.3.1) are reasonable. We calculate the reasonableness based on the following heuristic rules:

Table 6: Default values for shapes (cells in white) and lines (cells in light blue). Please refer to Table 11 for the actual value of *red-0*.

Type	x	y	width	height	Fill Color	Text
Rect	100 px	100 px	30 px	30 px	red-0	""

Type	Dashed	Source Arrow	Target Arrow	Text	Source ID	Target ID
Line	False	False	True	""	Not Omittable	Not Omittable

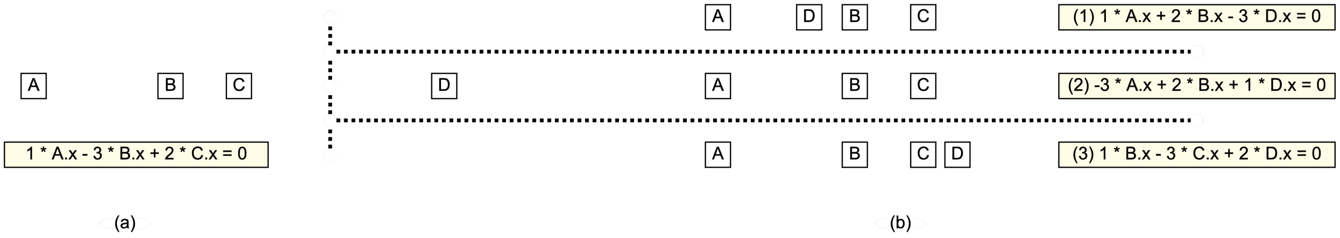

A
B
C
1 * A.x - 3 * B.x + 2 * C.x = 0
(a)
A
D
B
C
(1) 1 * A.x + 2 * B.x - 3 * D.x = 0
D
A
B
C
(2) -3 * A.x + 2 * B.x + 1 * D.x = 0
A
B
C
D
(3) 1 * B.x - 3 * C.x + 2 * D.x = 0
(b)

Figure 15: SGDiag calculates the position of the new element D with content-related posterior knowledge. Y coordinates are omitted in the figure. (a) Three elements and the equation among their position attributes; (b) three possible positions and equations for the next element. The coefficients (1, 2, and -3) are shuffled.

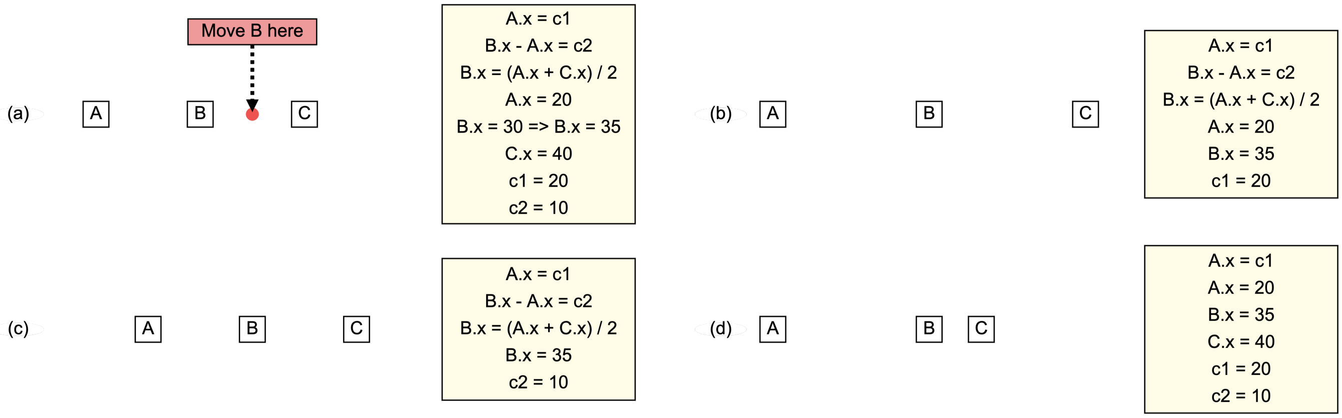


Figure 16: Possible solutions for the conflict command. (a) the original elements. SGDiag infers that the user want to change the x-coordinate of B from 30 to 35. (b) & (c) & (d) three different search results.

(1) The elements in the added equations should be similar. We measure the similarity with the number of the same attributes.
(2) If we utilize posterior knowledge, the original equations should be similar to the generated equations. We measure the similarity with cosine distances between the coefficient vectors.
(3) If the user specifies ranges of attribute values (attribute constraint Type 2 & 3), the values should be close to the center of the ranges.
(4) We prefer solutions without prior knowledge. The score is divided by one plus the number of attributes from prior knowledge as a penalty.

Error measures the side-effects of deleting conflict commands (Section 5.3.2). $Error = \lambda \#_{nonzero} L(\hat{X}) + \ L(\hat{X})$, where L is the optimization function in Equation 2. λ is a hyper-parameter that weighs the importance of the two items. We set $\lambda = 10$ in SGDiag.

5.4.2 Topology Visualization. We visualize the position relationships as the inner logic behind each solution to help the user select the appropriate candidate. By default, we display relationships of the last modified element only. Users can select other interesting elements manually.

We represent the attribute values by non-attribute variables as $\mathbf{a} = \{a_i\} = -R_a^{-1} * R_c * \mathbf{c}$ according to Formula 1, where a_i is the value of the i-th position attribute. Let R_i denote the i-th row in $-R_a^{-1} * R_c$, i.e., the coefficient vector of a_i. We compare the coefficients of non-attribute variables to discover implicit relationships behind the equations:

(1) Alignment, i.e., $R_i = R_j$. This relationship is presented with a dashed line connecting the center of the elements, as shown in Figure 17;
(2) Multiple distances, i.e., $R_i - R_j = N * (R_m - R_n)$, where N is an integer. This relationship is presented with arrows, as shown in Figure 17.

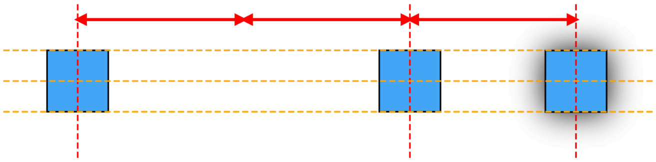


Figure 17: Visualization of the relationships. The yellow dashed lines indicate that the three squares have the same vertical coordinates. The red lines and arrows indicate that the horizontal distance between the first and the second squares is twice as long as that between the second and the third.

5.5 Intention Prediction

Behavior-related Prediction. SGDiag predicts subsequent commands by replacing the element designators in the executed commands with those of similar elements. As shown in Figure 9-right, when the user sets the color of an element to cyan, SGDiag recommends subsequent commands, such as *set the colors of elements with the same x coordinate to cyan.*

Content-related Prediction. SGDiag marks the possible positions of subsequent elements on the canvas, as shown in Figure 9-left. We predict subsequent elements with posterior knowledge, as discussed in Section 5.3.1.

5.6 GUI Design

Figure 18 shows the GUI of SGDiag. It is implemented with React and can run in any modern browser. We utilize Google Speech-to-Text as the automatic speech recognition service. The green dots on the canvas indicate the predicted positions of the next element. There are five tabs in the Function Panel:

(1) Candidates, where the user can select the appropriate solution (Figure 11-right);
(2) Element Settings, where the user can manually modify the attributes of the elements. Note that all the modifications are supported by voice command, and the use of the GUI is rare;
(3) Predicted Interactions, where the user can trigger the predicted intentions (Figure 9-right);
(4) Settings, where the user can clear the touch trajectories, disable the topology visualization and the next position prediction;
(5) Parsing Result, where the user can check and correct the parsing results (Figure 10-right).

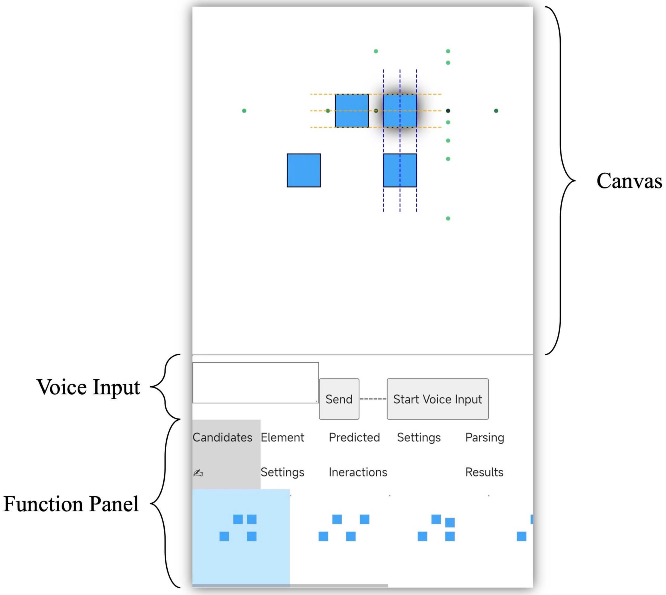


Figure 18: The GUI of SGDiag.

6 STUDY 2: EVALUATION OF SGDIAG

The goal of Study 2 is to evaluate whether SGDiag can provide correct functionalities and collaborate with the users so that they can accomplish diagram editing tasks with high efficiency, good results, and positive attitudes.

6.1 Tasks

Table 7 shows the details of the tasks. Given that some participants may lack necessary background knowledge, the experimenter provided hand-drawn sketches as references, as shown in Figure 24 in the Appendix. However, the participants did not need to follow the sketches strictly. The only requirement was the clear conveyance of the concepts.

6.2 Baseline

We compared SGDiag with PowerPoint for Android, a popular and state-of-the-art general-purpose tool that Microsoft specifically design for smartphones. It has over one billion downloads in Google play[5].

6.3 Procedure

We randomly divided the participants into two equal-sized groups. The study contained two phase for each participant. In the first phase, the participant used one of the editing tools (Group 1: SGDiag, Group 2: PowerPoint) on a 6.8-inch Android smartphone to finish the four tasks in random order. In the second phase, the participant used another tool (Group 1: PowerPoint, Group 2: SGDiag) on the same Android phone to finish the same four tasks in random order. By separating the two phases by more than three months, we prevented the user's different familiarity with the tasks from

[5] https://play.google.com/store/apps/details?id=com.microsoft.office.powerpoint

Table 7: An overview of tasks in Study 2.

Task ID	Discipline	Descriptions
1	Computer Science	The structure of a transformer encoder.
2	Chemistry	The cubic crystal structure of caesium chloride.
3	Linear Algebra	The pivot element of a matrix in row-echelon form.
4	Biology	Self-fertilization of the heterozygous (Law of segregation of genes).

influencing the outcomes. Participants finished 192 (24 users * 2 tools * 4 tasks) tasks in total.

In each phase, we briefly introduced the editing tool and demonstrated some basic interactions. When participants were using SGDiag, we provided a cheat sheet for supported functions and example utterances. After they finished the tasks, we distributed questionnaires to collect subjective feedback. For participants using SGDiag, we printed every command and the diagrams before and after the command. The participants were asked to annotate the following:

(1) whether the top-1 solution satisfied their intentions;
(2) if the top-1 solution was not satisfactory, how they improved the diagram.

After all of the participants finished their tasks, we invited judges to grade the editing results from 0 to 100. For every task, the judges sorted the editing results first and assigned scores to them according to the order. We didn't have any mandatory judging criteria, but we advised that they graded the diagrams on correctness, comprehensibility, and attractiveness. They did not know the editing tools for the result diagrams.

6.4 Participants

We recruited 24 users (12 males and 12 females, aged 20-32) for the editing tasks. None of them participated in the first study. We recruited 20 users online to grade the editing results. None of them participated in the first study or the editing tasks. All participants were casual users that had completed diagram editing tasks on a desktop device but had no experience in a mobile environment. Two of them would search and download online templates to boost their diagrams. The others did not have special setups.

6.5 Results

We evaluated if SGDiag could provide correct functionalities to assist users in accomplishing diagram editing tasks with high efficiency, quality, and experience. The results contain the following five aspects:

(1) Interaction behaviors. We report the collected user behaviors and summarize the most significant feature.
(2) Functional correctness. We assessed whether SGDiag implemented its functional design, i.e., producing candidate solutions (functionality 1 & 2) and recommending solutions in order (functionality 3).
(3) Interaction efficiency, i.e., if users could complete editing tasks quickly with SGDiag.
(4) Editing results, i.e., if users could achieve good editing results with SGDiag.
(5) Subjective feedback, i.e., if users had positive attitudes towards SGDiag.

6.5.1 Interaction Behaviors. All participants finished the tasks successfully. We collected 4658 editing commands. Most of the commands (60.65%) contain both speeches and gestures, indicating users' preferences for multimodal interactions.

A significant feature of the behaviors is the reuse of verbal commands. The users reused the same verbal commands with different gestures to execute the same operations to different elements. 71.85% of the commands fell into this category. This strategy not only enables batched operations of elements and improves interaction efficiency but also reduces the complexity of verbal commands, the mental burden, and the possibility of errors.

6.5.2 Functional Correctness.

16.36% (762) of user commands were accurate. SGDiag generated candidate solutions for the remaining instructions by complementing attributes (3464, 74.37%, functionality 1), resolving conflicts (422, 9.06%, functionality 2), or performing the two functionalities together (10, 0.21%). The top-1 accuracy of the proposed solutions is 94.84% (201 errors in 3896 commands, functionality 3).

All of the inaccurate solutions are related to position attributes. Users applied the following three strategies to correct the solutions:

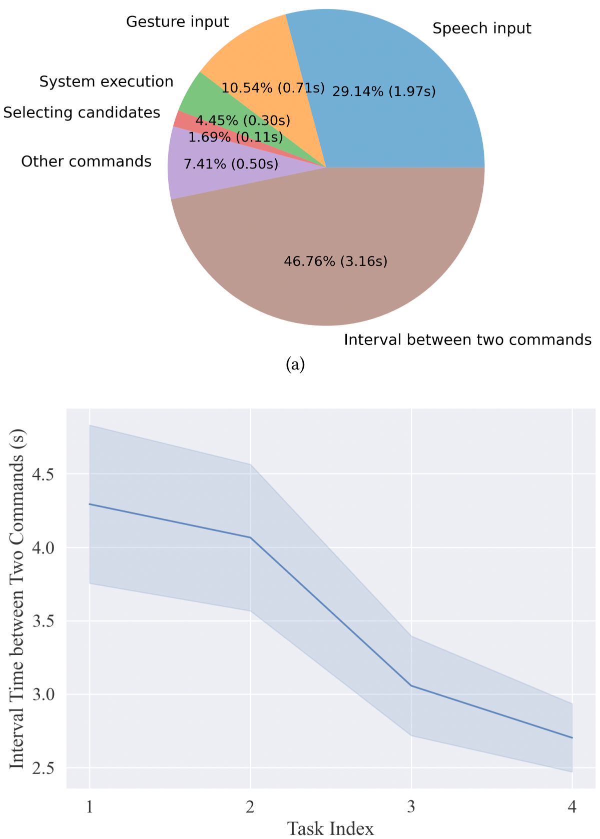
(1) Selecting another solution from the list (117 / 201, 58.21%). The average rank of the selected solutions was 2.92 (min = 2, median = 2, max = 13, sd = 1.81). SGDiag proposed a satisfactory solution for 97.84% (84 errors in 3896) of the commands.
(2) Giving modification commands based on unsatisfactory results (38 / 201, 18.91%).
(3) Revoking the commands (46 / 201, 22.89%).

As discussed in Section 4.2.5, predicting the subsequent intentions is difficult. 45.43% of the elements fit one of the predicted positions. The users triggered the recommended modification 19 times. Users complained that they had to switch to another tab (*Predicted Interactions* in Figure 18) to access the recommendations, which was inconvenient and forgettable.

6.5.3 Interaction Efficiency. Figure 19-a shows the time for different stages of SGDiag. The intervals between two commands occupied almost half of the time. During the interval, users reflected on the appropriate multimodal commands within the capability of SGDiag and referred to the cheatsheet if necessary. The interval time decreases when users get familiar with SGDiag. Figure 19-b shows the relationship between the intervals and task indexes[6]. Repeated measures ANOVA (RM-ANOVA) shows significant user

[6] Task ID and task index are different. For example, if the user's first task is Task 4, the index of Task 4 is 1, but its id is still 4.

learning effects ($F_{3,69} = 13.85, p < 0.0001$). A post hoc pairwise comparison with Sidak adjustment revealed a significant difference between Index 3 and 4, indicating that users' interaction time will decrease further if they continue to use SGDiag.



(a)

(b)

Figure 19: (a) Time distribution in different stages. If speeches and gestures are simultaneous, we attribute the time to the speech input. (b) The learning curve for the interval time. The shadow indicates one standard error.

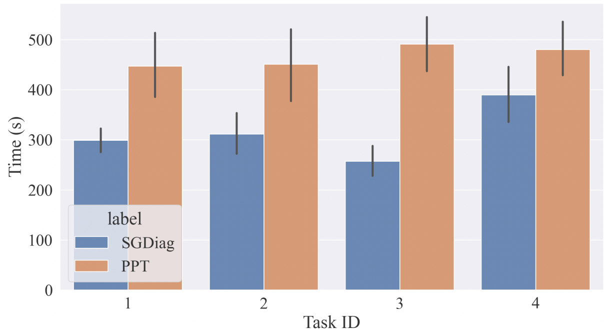


Figure 20: Interaction time for the two systems in different tasks. Error bar indicates one standard error.

Figure 20 compares the editing time of SGDiag and PowerPoint. In all of the four tasks, SGDiag was significantly more efficient than PowerPoint ($F_{1,23} = 19.09, p < 0.001$ for Task 1, $F_{1,23} = 10.44, p < 0.005$ for Task 2, $F_{1,23} = 50.47, p < 0.001$ for Task 3, $F_{1,23} = 4.74, p = 0.035 < 0.05$ for Task 4) and saved 32.75% of the interaction time on average. Table 8 compares different groups with different tools, indicating a significant difference between the two tools but no significant difference between the two groups. The only exception is that the speed of the first group significantly outperformed that of the second group when they were using PowerPoint. A possible reason is that the first group used PowerPoint in the second phase and was more familiar with the tasks. The reasons for the low efficiency of the baseline are threefold:

1. Unintentional interactions. Existing commercial applications rely solely on GUI to complete the interaction process. The smartphone screen size is limited, and different elements are located relatively close to each other, which is prone to unintentional interactions.
2. Seeking functions. Due to the limited size of the screen, many functions are hidden in multi-level menus. Users need to remember and find where the functions are located.
3. Low gesture accuracy. The baseline system did not correct users' inaccurate gestures. Users waste their time on subsequent adjustments.

6.5.4 Editing Results. Figure 21 compares the scores of the editing results from different systems. In all four tasks, the scores of diagrams edited with SGDiag are significantly higher than those with the baseline (p < 0.0001 in all the tasks). The average score of our system is 86.43 (sd = 15.31), 21.89% higher than the baseline (70.90, sd = 17.87). Table 9 compares different groups with different tools, indicating a significant difference between the two tools but no significant difference between the two groups. Figures (25 - 28) in the Appendix demonstrate the editing results of the two systems in different tasks.

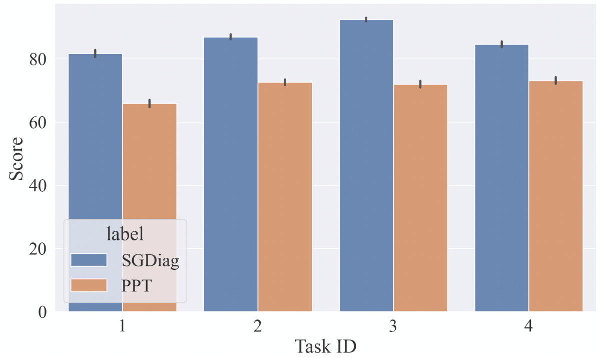


Figure 21: The scores of the editing results from different systems in different tasks. Error bar indicates one standard error.

Figure 22 shows the editing time and scores of the diagrams, illustrating that users got better results in less time with SGDiag. Considering that the subjective scores were noisy and strongly influenced by judges' personal preferences, we do not regress the data directly. Instead, we fit the upper and lower bounds of the scores. The two bounds of PowerPoint grow slowly as the editing

Table 8: P-values of the interaction time (the average time a user spent on one task) from the different groups and tools.

	Group 1 & SGDiag	Group 1 & PPT	Group 2 & SGDiag	Group 2 & PPT
Group 1 & SGDiag (mean=333.61, sd=51.35)	-	<0.0001	0.074	<0.05
Group 1 & PPT (mean=411.68, sd=87.28)	<0.0001	-	<0.0001	<0.05
Group 2 & SGDiag (mean=295.25, sd=44.19)	0.074	<0.0001	-	<0.001
Group 2 & PPT (mean=523.38, sd=108.79)	<0.05	<0.05	<0.001	-

Table 9: P-values of the scores (the average score a user got on the 4 tasks) from the different groups and tools.

	Group 1 & SGDiag	Group 1 & PPT	Group 2 & SGDiag	Group 2 & PPT
Group 1 & SGDiag (mean=86.30, sd=1.17)	-	<0.0001	0.7729	<0.0001
Group 1 & PPT (mean=70.91, sd=2.19)	<0.0001	-	<0.0001	0.9942
Group 2 & SGDiag (mean=86.56, sd=2.69)	0.7729	<0.0001	-	<0.0001
Group 2 & PPT (mean=70.90, sd=2.17)	<0.0001	0.9942	<0.0001	-

time increases (1.20 points per minute and 1.18 points per minute, respectively). Our system has a high upper bound even when the editing time is limited, and its lower bound grows very rapidly (2.02 points per minute).

6.5.5 Subjective Feedback. The results of the 7-point Likert scale are displayed in Table 10. Wilcoxon tests show that SGDiag significantly ($p < 0.05$) outperforms the baseline for all the aspects. Users showed a strong preference for SGDiag. They were satisfied with its ability to automatically correct vague instructions, *"Building a diagram requires only a few taps, and I do not need to control it precisely"* (P2, P4). They were surprised by its intelligence, especially when modifying the diagram. For example, in Task 1, P7 set the elements too far apart in the vertical direction, and there was not enough space on the canvas for the remaining elements. He dragged down the element "Input" to reduce the vertical distance. He shouted, "How smart it is!" when he found SGDiag automatically shortened the vertical distance between other elements.

7 DISCUSSION

7.1 Generalize to Other Tasks, Devices and Modalities

The core idea of CET is to divide the task into two parts, one of which is finished by an intelligent agent. This mechanism is not limited by specific tasks, modalities, or devices.

Researchers can conduct formative studies for other tasks, as discussed in Section 3, to reveal the appropriate task division between the human and the computer. The design of CET (discussed in Section 4) can be directly applied to other editing tasks. The editing tools need different content modeling and solution generation strategies for different tasks. For example, in abstract drawing tasks [24], users may be more concerned with the shapes and colors of the elements rather than the topological relationships.

CET can support more interaction modalities, such as mid-air pointing [83], gaze [107], and head gestures [60, 80], to support more devices and people and improve solution generation accuracy. For example, SGDiag requires users to touch the screen, which is not applicable to non-touchscreen devices and people with motor disabilities. Researchers can use eye and head movements to augment or replace touch interactions. The whole interaction process remains very natural: users unconsciously look at them when referring to elements or positions on the canvas.

CET can be migrated to more devices, such as interactive tabletops [12], virtual reality devices [122], and so on. For example, CET can support collaborative editing of multiple people on an interactive tabletop to avoid the embarrassment of all users gathering around the device in a conference room. In addition to being a new editing tool, CET can be integrated into existing editing tools as a plug-in.

7.2 The Role of CET: an Editing Tool rather than a Creative Tool

Hwang [44] splits the creative process into four stages: the Q&A stage, the wandering stage, the hands-on stage, and the camera-ready stage. CET benefits users in the last stage, where they *"execute ideas into presentable"* [44]. This explains why we provided hand-drawn sketches to the users in Study 2, as we assumed they had already determined the final ideas when they used SGDiag. Helping users optimize and finalize their ideas is the responsibility of the creative tools and is beyond the scope of CET.

However, users can benefit from the core idea of CET in other editing stages: focusing on the idea itself rather than the operations

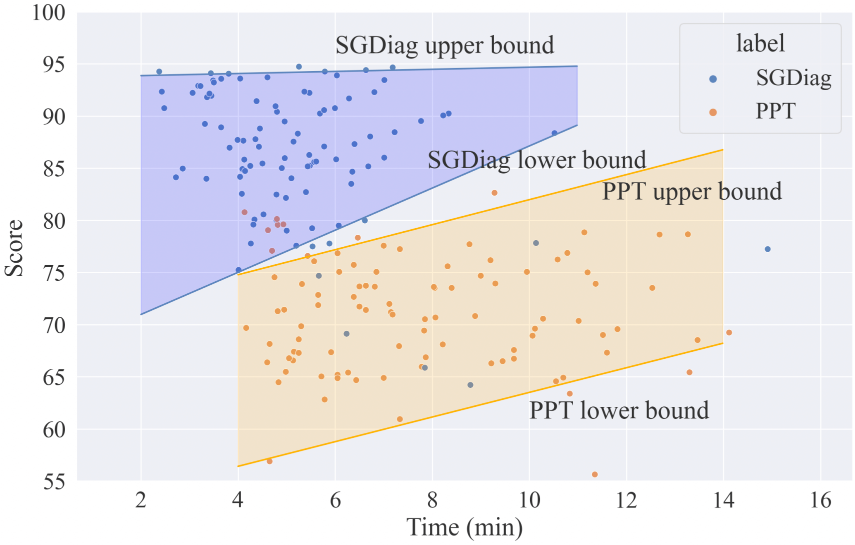


Figure 22: The editing time and score for each diagram. The lines indicate the upper and the lower bounds for the two systems.

Table 10: Subjective feedback.

Statements	SGDiag	Baseline	p-value
The interaction is natural and easy to learn.	5.58 (sd=0.90)	4.67 (sd=1.78)	**0.016 (<0.05)**
The collabration is intelligent.	5.46 (sd=0.90)	N/A	N/A
The dialogues help finish the tasks quickly.	5.17 (sd=1.44)	N/A	N/A
The GUI helps finish the tasks quickly.	5.50 (sd=1.23)	4.17 (sd=1.00)	**0.00020 (<0.0005)**
You are willing to use the system.	4.79 (sd=1.50)	2.46 (sd=1.61)	**0.00024 (<0.0005)**
You are willing to use the system in mobile scenarios.	5.96 (sd=0.72)	3.83 (sd=1.56)	**0.00015 (<0.0005)**

of tools. For example, finalizing an idea may involve numerous adjustments, which leads to significant overhead in traditional GUIs. Future creative tools can support associated modifications (Section 3.3.2) to reduce the interaction burden.

8 LIMITATION & FUTURE WORK

Using in real-world scenarios. Due to privacy [27] and social acceptance [95] considerations, users have mixed opinions regarding voice interaction in public scenarios. Many approaches, such as PrivateTalk [114], silent voice interaction [25, 64, 102], and soundproof masks [81], have been proposed by researchers to overcome this issue. In the future, SGDiag can combine these strategies to assist users in real-world settings.

Comparing with more baselines. Users, particularly experts, may have their own configuration for editing tasks. Some users, for example, utilize a stylus and a drawing tablet to complete creative digital drawings [103]. Future work might compare SGDiag to more professional setups, evaluating whether SGDiag can benefit a variety of people, including specialists.

Integrating existing features. SGDiag can integrate functions from existing tools in actual use. For example, the user can select a theme in PowerPoint. SGDiag can also support users to select a theme as prior knowledge, with which the system can better collaborate with the user.

Understanding the semantics of editing contents. SGDiag does not model the semantic relationships among the diagram elements. The diagram semantics are important and help generate reasonable solutions. For example, the element with the text *sky* tends to be above the element with the text *earth*. Future work can integrate the existing diagram interpretation and reasoning [53] methods to improve collaboration quality.

Modeling relationships among non-position attributes. SGDiag models the relationships among the position attributes. For non-position attributes, it only stores their values. It is promising for SGDiag to support more attributes in the relationships. For example, the user can draw a heat map with little effort in which the colors of the cells are calculated automatically according to their texts.

Modeling the nonlinearity. SGDiag models the diagram contents with a matrix and utilizes many existing linear algebra algorithms to maintain the relationships. Modeling the nonlinearity is helpful in some cases, for example, when the user wants to keep two lines perpendicular.

Parsing the commands with machine learning methods. SGDiag parses user commands with a CFG-based method. CFG is suitable for quick prototyping and widely used in the community of human-computer interaction [71, 72]. Researchers can collect interaction data and train a sophisticated model in the future.

9 CONCLUSION

In this paper, we present a new type of tool: the human-computer collaborative editing tool (CET). The core idea is to reduce the interaction burden by employing an intelligent agent to help with some of the work. We selected conceptual diagram editing as a typical example from various editing tasks and conducted the first study (N = 16) to observe the task division between users and assistants. We completed the design and implementation of the tool and conducted the second study (N = 24 + 20) to evaluate its performance. The experimental results showed that our tool had significant advantages over the state-of-the-art commercial application in terms of editing efficiency, editing effectiveness, and subjective feedback. We hope CET can inspire subsequent research and support more devices, modalities, and interaction tasks.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant no. 62132010, and by Beijing Key Lab of Networked Multimedia, the Institute for Guo Qiang, Tsinghua University, Institute for Artificial Intelligence, Tsinghua University (THUAI), and by 2025 Key Technological Innovation Program of Ningbo City under Grant No.2022Z080. We would like to thank Guangzhao Li and Yuxuan Chen for their great help in conducting the user studies and implementing the system. We would like to thank Luanqiao Yang for his great help in addressing presentation issues.

REFERENCES

[1] Ayodeji Opeyemi Abioye, Stephen D Prior, Peter Saddington, and Sarvapali D Ramchurn. 2022. The performance and cognitive workload analysis of a multimodal speech and visual gesture (mSVG) UAV control interface. *Robotics and Autonomous Systems* 147 (2022), 103915.

[2] Abdul Rafey Aftab, Michael Von Der Beeck, Steven Rohrhirsch, Benoit Diotte, and Michael Feld. 2021. Multimodal Fusion Using Deep Learning Applied to Driver's Referencing of Outside-Vehicle Objects. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1108–1115.

[3] Shaaron Ainsworth, Vaughan Prain, and Russell Tytler. 2011. Drawing to learn in science. *Science* 333, 6046 (2011), 1096–1097.

[4] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. 2020. Voxento: A Prototype Voice-Controlled Interactive Search Engine for Lifelogs. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge* (Dublin, Ireland) *(LSC '20)*. Association for Computing Machinery, New York, NY, USA, 77–81. https://doi.org/10.1145/3379172.3391728

[5] Tammy D Allen and Kristen Shockley. 2009. Flexible work arrangements: Help or hype. *Handbook of families and work: Interdisciplinary perspectives* (2009), 265–284.

[6] Amos Azaria, Jayant Krishnamurthy, and Tom M. Mitchell. 2016. Instructable Intelligent Personal Agent. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 2681–2689.

[7] Evelyn Balfe and Barry Smyth. 2004. Improving web search through collaborative query recommendation. In *ECAI*, Vol. 16. 268.

[8] Thomas Ball, Shannon Kao, Richard Knoll, and Daryl Zuniga. 2020. TileCode: Creation of Video Games on Gaming Handhelds. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1182–1193. https://doi.org/10.1145/3379337.3415839

[9] Nicola J. Bidwell, Thomas Reitmaier, Gary Marsden, and Susan Hansen. 2010. Designing with Mobile Digital Storytelling in Rural Africa. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1593–1602. https://doi.org/10.1145/1753326.1753564

[10] Jeff A. Bilmes, Xiao Li, Jonathan Malkin, Kelley Kilanski, Richard Wright, Katrin Kirchhoff, Amarnag Subramanya, Susumu Harada, James A. Landay, Patricia Dowden, and Howard Chizeck. 2005. The Vocal Joystick: A Voice-Based Human-Computer Interface for Individuals with Motor Impairments. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (Vancouver, British Columbia, Canada) *(HLT '05)*. Association for Computational Linguistics, USA, 995–1002. https://doi.org/10.3115/1220575.1220700

[11] Richard A. Bolt. 1980. "Put-That-There": Voice and Gesture at the Graphics Interface. *SIGGRAPH Comput. Graph.* 14, 3 (jul 1980), 262–270. https://doi.org/10.1145/965105.807503

[12] Christophe Bortolaso, Matthew Oskamp, Greg Phillips, Carl Gutwin, and T.C. Nicholas Graham. 2014. The Effect of View Techniques on Collaboration and Awareness in Tabletop Map-Based Tasks. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces* (Dresden, Germany) *(ITS '14)*. Association for Computing Machinery, New York, NY, USA, 79–88. https://doi.org/10.1145/2669485.2669504

[13] John C Bowman and Andy Hammerlindl. 2008. Asymptote: A vector graphics language. *TUGboat: The Communications of the TEX Users Group* 29, 2 (2008), 288–294.

[14] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* (2015).

[15] Pablo Cesar, Dick C. A. Bulterman, Jack Jansen, David Geerts, Hendrik Knoche, and William Seager. 2009. Fragment, Tag, Enrich, and Send: Enhancing Social Sharing of Video. *ACM Trans. Multimedia Comput. Commun. Appl.* 5, 3, Article 19 (aug 2009), 27 pages. https://doi.org/10.1145/1556134.1556136

[16] Alex J Champandard. 2016. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768* (2016).

[17] Mauro Cherubini, Gina Venolia, Rob DeLine, and Amy J. Ko. 2007. Let's Go to the Whiteboard: How and Why Software Developers Use Drawings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 557–566. https://doi.org/10.1145/1240624.1240714

[18] Keunwoo Choi, George Fazekas, and Mark Sandler. 2016. Text-based LSTM networks for automatic music composition. *arXiv preprint arXiv:1604.05358* (2016).

[19] Lewis L. Chuang, Stella F. Donker, Andrew L. Kun, and Christian P. Janssen. 2018. Workshop on The Mobile Office. In *Adjunct Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Toronto, ON, Canada) *(AutomotiveUI '18)*. Association for Computing Machinery, New York, NY, USA, 10–16. https://doi.org/10.1145/3239092.3239094

[20] 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 *CHI Conference on Human Factors in Computing Systems*. 1–19.

[21] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. Association for Computing Machinery, New York, NY, USA, 329–340. https://doi.org/10.1145/3172944.3172983

[22] P. R. Cohen, M. Dalrymple, D. B. Moran, F. C. Pereira, and J. W. Sullivan. 1989. Synergistic Use of Direct Manipulation and Natural Language. *SIGCHI Bull.* 20, SI (mar 1989), 227–233. https://doi.org/10.1145/67450.67494

[23] Liwei Dai, Rich Goldman, Andrew Sears, and Jeremy Lozier. 2003. Speech-Based Cursor Control: A Study of Grid-Based Solutions. *SIGACCESS Access. Comput.* 77–78 (sep 2003), 94–101. https://doi.org/10.1145/1029014.1028648

[24] Nicholas Davis, Chih-PIn Hsiao, Kunwar Yashraj Singh, Lisa Li, and Brian Magerko. 2016. Empirically Studying Participatory Sense-Making in Abstract Drawing with a Co-Creative Cognitive Agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (Sonoma, California, USA) *(IUI '16)*. Association for Computing Machinery, New York, NY, USA, 196–207. https://doi.org/10.1145/2856767.2856795

[25] Bruce Denby, Tanja Schultz, Kiyoshi Honda, Thomas Hueber, Jim M Gilbert, and Jonathan S Brumberg. 2010. Silent speech interfaces. *Speech Communication* 52, 4 (2010), 270–287.

[26] Daantje Derks, Arnold B Bakker, Pascale Peters, and Pauline van Wingerden. 2016. Work-related smartphone use, work–family conflict and family role performance: The role of segmentation preference. *Human relations* 69, 5 (2016), 1045–1068.

[27] Aarthi Easwara Moorthy and Kim-Phuong L Vu. 2015. Privacy concerns for use of voice activated personal assistant in the public space. *International Journal of Human-Computer Interaction* 31, 4 (2015), 307–335.

[28] Judith E. Fan, Monica Dinculescu, and David Ha. 2019. Collabdraw: An Environment for Collaborative Sketching with an Artificial Agent. In *Proceedings of the 2019 on Creativity and Cognition* (San Diego, CA, USA) *(C&C '19)*. Association for Computing Machinery, New York, NY, USA, 556–561. https://doi.org/10.1145/3325480.3326578

[29] Michael Fischer, Giovanni Campagna, Silei Xu, and Monica S. Lam. 2018. Brassau: Automatic Generation of Graphical User Interfaces for Virtual Assistants. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Barcelona, Spain) *(MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, Article 33, 12 pages. https://doi.org/10.1145/3229434.3229481

[30] Michael H. Fischer, Giovanni Campagna, Euirim Choi, and Monica S. Lam. 2021. DIY Assistant: A Multi-Modal End-User Programmable Virtual Assistant. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation* (Virtual, Canada) *(PLDI 2021)*. Association for Computing Machinery, New York, NY, USA, 312–327. https://doi.org/10.1145/3453483.3454046

[31] David Frohlich, Simon Robinson, Kristen Eglinton, Matt Jones, and Elina Vartiainen. 2012. Creative Cameraphone Use in Rural Developing Regions. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (San Francisco, California, USA) *(MobileHCI '12)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/2371574.2371603

[32] David M. Frohlich, Dorothy Rachovides, Kiriaki Riga, Ramnath Bhat, Maxine Frank, Eran Edirisinghe, Dhammike Wickramanayaka, Matt Jones, and Will Harwood. 2009. StoryBank: Mobile Digital Storytelling in a Development Context. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1761–1770. https://doi.org/10.1145/1518701.1518972

[33] Markus Funk, Vanessa Tobisch, and Adam Emfield. 2020. Non-Verbal Auditory Input for Controlling Binary, Discrete, and Continuous Input in Automotive User Interfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376816

[34] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3985–3993.

[35] Katy Ilonka Gero and Lydia B. Chilton. 2019. Metaphoria: An Algorithmic Companion for Metaphor Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300526

[36] Vinod Goel. 1995. *Sketches of thought*. MIT press.

[37] Thomas R. G. Green and Marian Petre. 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing* 7, 2 (1996), 131–174.

[38] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O. Riedl. 2019. Friend, Collaborator, Student, Manager: How Design of an AI-Driven Game Level Editor Affects Creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300854

[39] David Ha and Douglas Eck. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017).

[40] Susumu Harada, James A. Landay, Jonathan Malkin, Xiao Li, and Jeff A. Bilmes. 2006. The Vocal Joystick: Evaluation of Voice-Based Cursor Control Techniques. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility* (Portland, Oregon, USA) *(Assets '06)*. Association for Computing Machinery, New York, NY, USA, 197–204. https://doi.org/10.1145/1168987.1169021

[41] Susumu Harada, Jacob O. Wobbrock, and James A. Landay. 2007. Voicedraw: A Hands-Free Voice-Driven Drawing Application for People with Motor Impairments. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility* (Tempe, Arizona, USA) *(Assets '07)*. Association for Computing Machinery, New York, NY, USA, 27–34. https://doi.org/10.1145/1296843.1296850

[42] Christian Holz and Patrick Baudisch. 2010. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 581–590. https://doi.org/10.1145/1753326.1753413

[43] Anthony J. Hornof and Anna Cavender. 2005. EyeDraw: Enabling Children with Severe Motor Impairments to Draw with Their Eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 161–170. https://doi.org/10.1145/1054972.1054995

[44] Angel Hsing-Chi Hwang. 2022. Too Late to Be Creative? AI-Empowered Tools in Creative Processes. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 38, 9 pages. https://doi.org/10.1145/3491101.3503549

[45] Takeo Igarashi and John F. Hughes. 2001. Voice as Sound: Using Non-Verbal Voice Input for Interactive Control. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida) *(UIST '01)*. Association for Computing Machinery, New York, NY, USA, 155–156. https://doi.org/10.1145/502348.502372

[46] Yue Jiang, Ruofei Du, Christof Lutteroth, and Wolfgang Stuerzlinger. 2019. ORC Layout: Adaptive GUI Layout with OR-Constraints. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300643

[47] Yue Jiang, Wolfgang Stuerzlinger, Matthias Zwicker, and Christof Lutteroth. 2020. ORCSolver: An Efficient Solver for Adaptive GUI Layout with OR-Constraints. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376610

[48] Michael Johnston, John Chen, Patrick Ehlen, Hyuckchul Jung, Jay Lieske, Aarthi Reddy, Ethan Selfridge, Svetlana Stoyanchev, Brant Vasilieff, and Jay Wilpon. 2014. Mva: The multimodal virtual assistant. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 257–259.

[49] Tero Jokela, Jaakko T. Lehikoinen, and Hannu Korhonen. 2008. Mobile Multimedia Presentation Editor: Enabling Creation of Audio-Visual Stories on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 63–72. https://doi.org/10.1145/1357054.1357066

[50] Ed Kaiser, Alex Olwal, David McGee, Hrvoje Benko, Andrea Corradini, Xiaoguang Li, Phil Cohen, and Steven Feiner. 2003. Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality. In *Proceedings of the 5th International Conference on Multimodal Interfaces* (Vancouver, British Columbia, Canada) *(ICMI '03)*. Association for Computing Machinery, New York, NY, USA, 12–19. https://doi.org/10.1145/958432.958438

[51] Pegah Karimi, Mary Lou Maher, Nicholas Davis, and Kazjon Grace. 2019. Deep learning in a computational model for conceptual shifts in a co-creative design system. *arXiv preprint arXiv:1906.10188* (2019).

[52] L Karl, M Pettey, and B Shneiderman. 1993. Speech-activated versus mouse-activated commands for word processing applications. *International Journal of Man-Machine Studies* 39, 4 (1993), 667–687.

[53] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A Diagram Is Worth A Dozen Images. https://doi.org/10.48550/ARXIV.1603.07396

[54] Jeongyeon Kim, Yubin Choi, Minsuk Kahng, and Juho Kim. 2022. FitVid: Responsive and Flexible Video Content Adaptation. In *CHI Conference on Human Factors in Computing Systems*. 1–16.

[55] Joy Kim, Mira Dontcheva, Wilmot Li, Michael S. Bernstein, and Daniela Steinsapir. 2015. Motif: Supporting Novice Creativity through Expert Patterns. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1211–1220. https://doi.org/10.1145/2702123.2702507

[56] Meekyeong Kim, Chuleui Hong, Deakyung Kwon, and Sungchul Hong. 2012. Multimedia presentation authoring system for e-learning contents in mobile environment. *Appl. Math* 6, 2S (2012), 705S–711S.

[57] Tae Soo Kim, DaEun Choi, Yoonseo Choi, and Juho Kim. 2022. Stylette: Styling the Web with Natural Language. In *CHI Conference on Human Factors in Computing Systems*. 1–17.

[58] Yea-Seul Kim, Mira Dontcheva, Eytan Adar, and Jessica Hullman. 2019. Vocal Shortcuts for Creative Experts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300562

[59] David Kirk, Abigail Sellen, Carsten Rother, and Ken Wood. 2006. Understanding photowork. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 761–770.

[60] Rick Kjeldsen. 2006. Improvements in Vision-Based Pointer Control. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility* (Portland, Oregon, USA) *(Assets '06)*. Association for Computing Machinery, New York, NY, USA, 189–196. https://doi.org/10.1145/1168987.1169020

[61] Sandeep Kochhar and Mark Friedell. 1990. User control in cooperative computer-aided design. In *Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology*. 143–151.

[62] Sandeep Kochhar, Mark Friedell, Joe Marks, Steve Sistare, and Louis Weitzman. 1994. Interaction paradigms for human-computer cooperation in design. In *Conference companion on Human factors in computing systems*. 187–188.

[63] Ilpo Koskinen, Esko Kurvinen, and Turo-Kimmo Lehtonen. 2002. Mobile image. (2002).

[64] Yusuke Kunimi, Masa Ogata, Hirotaka Hiraki, Motoshi Itagaki, Shusuke Kanazawa, and Masaaki Mochimaru. 2022. E-MASK: A Mask-Shaped Interface for Silent Speech Interaction with Flexible Strain Sensors. In *Augmented Humans 2022*. 26–34.

[65] Elina Kuosmanen, Valerii Kan, Aku Visuri, Assam Boudjelthia, Lokmane Krizou, and Denzil Ferreira. 2019. Measuring Parkinson's Disease Motor Symptoms with Smartphone-Based Drawing Tasks. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers* (London, United Kingdom) *(UbiComp/ISWC '19 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 1182–1185. https:

//doi.org/10.1145/3341162.3344833

[66] Markku Laine, Ai Nakajima, Niraj Dayama, and Antti Oulasvirta. 2020. Layout as a service (LaaS): A service platform for self-optimizing web layouts. In *International Conference on Web Engineering*. Springer, 19–26.

[67] Gierad P. Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. PixelTone: A Multimodal Interface for Image Editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2185–2194. https://doi.org/10.1145/2470654.2481301

[68] Eric Laurier. 2004. Doing office work on the motorway. *Theory, Culture & Society* 21, 4-5 (2004), 261–277.

[69] Bongshin Lee, Arjun Srinivasan, John Stasko, Melanie Tory, and Vidya Setlur. 2018. Multimodal Interaction for Data Visualization. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (Castiglione della Pescaia, Grosseto, Italy) *(AVI '18)*. Association for Computing Machinery, New York, NY, USA, Article 11, 3 pages. https://doi.org/10.1145/3206505.3206602

[70] Po-shen Lee, Jevin D West, and Bill Howe. 2017. Viziometrics: Analyzing visual information in the scientific literature. *IEEE Transactions on Big Data* 4, 1 (2017), 117–129.

[71] Toby Jia-Jun Li, Amos Azaria, and Brad A. Myers. 2017. SUGILITE: Creating Multimodal Smartphone Automation by Demonstration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 6038–6049. https://doi.org/10.1145/3025453.3025483

[72] Toby Jia-Jun Li, Marissa Radensky, Justin Jia, Kirielle Singarajah, Tom M. Mitchell, and Brad A. Myers. 2019. PUMICE: A Multi-Modal Agent That Learns Concepts and Conditionals from Natural Language and Demonstrations. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 577–589. https://doi.org/10.1145/3332165.3347899

[73] Qin Lin, Nuo Pang, and Zhiying Hong. 2021. Automated Multi-Modal Video Editing for Ads Video. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 4823–4827. https://doi.org/10.1145/3474085.3479205

[74] Yuyu Lin, Jiahao Guo, Yang Chen, Cheng Yao, and Fangtian Ying. 2020. It Is Your Turn: Collaborative Ideation With a Co-Creative Robot through Sketch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376258

[75] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. 2020. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376739

[76] Andrés Lucero, Jussi Holopainen, and Tero Jokela. 2012. MobiComics: Collaborative Use of Mobile Phones and Large Displays for Public Expression. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (San Francisco, California, USA) *(MobileHCI '12)*. Association for Computing Machinery, New York, NY, USA, 383–392. https://doi.org/10.1145/2371574.2371634

[77] Dor Ma'ayan, Wode Ni, Katherine Ye, Chinmay Kulkarni, and Joshua Sunshine. 2020. How Domain Experts Create Conceptual Diagrams and Implications for Tool Design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376253

[78] Dor Ma'ayan, Wode Ni, Katherine Ye, Chinmay Kulkarni, and Joshua Sunshine. 2020. How domain experts create conceptual diagrams and implications for tool design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[79] Ann Mäkelä, Verena Giller, Manfred Tscheligi, and Reinhard Sefelin. 2000. Joking, Storytelling, Artsharing, Expressing Affection: A Field Trial of How Children and Their Social Network Communicate with Digital Images in Leisure Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 548–555. https://doi.org/10.1145/332040.332499

[80] Rainer Malkewitz. 1998. Head Pointing and Speech Control as a Hands-Free Interface to Desktop Computing. In *Proceedings of the Third International ACM Conference on Assistive Technologies* (Marina del Rey, California, USA) *(Assets '98)*. Association for Computing Machinery, New York, NY, USA, 182–188. https://doi.org/10.1145/274497.274531

[81] metadox. 2022. METADOX - Because playing in silence is no fun. Retrieved November 21, 2022 from https://metadox.pro/

[82] Brad A Myers, John F Pane, and Amy J Ko. 2004. Natural programming languages and environments. *Commun. ACM* 47, 9 (2004), 47–52.

[83] Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, and Michel Beaudouin-Lafon. 2015. Mid-Air Pointing on Ultra-Walls. *ACM Trans. Comput.-Hum. Interact.* 22, 5, Article 21 (aug 2015), 62 pages. https://doi.org/10.1145/2766448

[84] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I Lead, You Help but Only with Enough Details: Understanding User Experience of Co-Creation with Artificial Intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174223

[85] Margrethe H. Olson. 1983. Remote Office Work: Changing Work Patterns in Space and Time. *Commun. ACM* 26, 3 (mar 1983), 182–187. https://doi.org/10.1145/358061.358068

[86] Sharon Oviatt. 2006. Human-Centered Design Meets Cognitive Load Theory: Designing Interfaces That Help People Think. In *Proceedings of the 14th ACM International Conference on Multimedia* (Santa Barbara, CA, USA) *(MM '06)*. Association for Computing Machinery, New York, NY, USA, 871–880. https://doi.org/10.1145/1180639.1180831

[87] Sharon Oviatt and Philip R Cohen. 2015. The paradigm shift to multimodality in contemporary computer interfaces. *Synthesis lectures on human-centered informatics* 8, 3 (2015), 1–243.

[88] Sharon Oviatt, Rachel Coulston, and Rebecca Lunsford. 2004. When Do We Interact Multimodally? Cognitive Load and Multimodal Communication Patterns. In *Proceedings of the 6th International Conference on Multimodal Interfaces* (State College, PA, USA) *(ICMI '04)*. Association for Computing Machinery, New York, NY, USA, 129–136. https://doi.org/10.1145/1027933.1027957

[89] Andrea Papenmeier, Alfred Sliwa, Dagmar Kern, Daniel Hienert, Ahmet Aker, and Norbert Fuhr. 2020. 'A Modern Up-To-Date Laptop' - Vagueness in Natural Language Queries for Product Search. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 2077–2089. https://doi.org/10.1145/3357236.3395489

[90] Nicole Perterer, Christiane Moser, Alexander Meschtscherjakov, Alina Krischkowsky, and Manfred Tscheligi. 2016. Activities and Technology Usage While Driving: A Field Study with Private Short-Distance Car Commuters. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction* (Gothenburg, Sweden) *(NordiCHI '16)*. Association for Computing Machinery, New York, NY, USA, Article 41, 10 pages. https://doi.org/10.1145/2971485.2971556

[91] Bastian Pfleging, Maurice Rang, and Nora Broy. 2016. Investigating User Needs for Non-Driving-Related Activities during Automated Driving. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia* (Rovaniemi, Finland) *(MUM '16)*. Association for Computing Machinery, New York, NY, USA, 91–99. https://doi.org/10.1145/3012709.3012735

[92] Venkatesh Potluri, Liang He, Christine Chen, Jon E. Froehlich, and Jennifer Mankoff. 2019. A Multi-Modal Approach for Blind and Visually Impaired Developers to Edit Webpage Designs. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, PA, USA) *(ASSETS '19)*. Association for Computing Machinery, New York, NY, USA, 612–614. https://doi.org/10.1145/3308561.3354626

[93] Alex Quinn, B Bederson, Elizabeth Bonsignore, and Allison Druin. 2009. StoryKit: Designing a mobile application for story creation by children and older adults. *College Park, MD: Human Computer Interaction Lab, University of Maryland* (2009), 1–10.

[94] Casey Reas and Ben Fry. 2006. Processing: programming for the media arts. *Ai & Society* 20, 4 (2006), 526–538.

[95] Julie Rico and Stephen Brewster. 2010. Gesture and voice prototyping for early evaluations of social acceptability in multimodal interfaces. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. 1–9.

[96] Dmitri Rozgonjuk and Jon Elhai. 2018. Problematic Smartphone Usage, Emotion Regulation, and Social and Non-Social Smartphone Use. In *Proceedings of the Technology, Mind, and Society* (Washington, DC, USA) *(TechMindSociety '18)*. Association for Computing Machinery, New York, NY, USA, Article 35, 1 pages. https://doi.org/10.1145/3183654.3183664

[97] Nazmus Saquib, Rubaiat Habib Kazi, Li-yi Wei, Gloria Mark, and Deb Roy. 2021. Constructing Embodied Algebra by Sketching. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.

[98] Clemens Schartmüller, Klemens Weigl, Philipp Wintersberger, Andreas Riener, and Marco Steinhauser. 2019. Text Comprehension: Heads-Up vs. Auditory Displays: Implications for a Productive Work Environment in SAE Level 3 Automated Vehicles. In *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Utrecht, Netherlands) *(AutomotiveUI '19)*. Association for Computing Machinery, New York, NY, USA, 342–354. https://doi.org/10.1145/3342197.3344547

[99] Weinan Shi, Chun Yu, Shuyi Fan, Feng Wang, Tong Wang, Xin Yi, Xiaojun Bi, and Yuanchun Shi. 2019. VIPBoard: Improving Screen-Reader Keyboard for Visually Impaired People with Character-Level Auto Correction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300747

[100] Chengyu Su, Chao Yang, Yonghui Chen, Fupan Wang, Fang Wang, Yadong Wu, and Xiaorong Zhang. 2021. Natural multimodal interaction in immersive flow

visualization. *Visual Informatics* 5, 4 (2021), 56–66.

[101] Anne Sullivan, Mirjam Palosaari Eladhari, and Michael Cook. 2018. Tarot-Based Narrative Generation. In *Proceedings of the 13th International Conference on the Foundations of Digital Games* (Malmö, Sweden) *(FDG '18)*. Association for Computing Machinery, New York, NY, USA, Article 54, 7 pages. https://doi.org/10.1145/3235765.3235819

[102] Ke Sun, Chun Yu, Weinan Shi, Lan Liu, and Yuanchun Shi. 2018. Lip-interact: Improving mobile device interaction with silent speech commands. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology.* 581–593.

[103] Minghui Sun, Xiang Cao, Hyunyoung Song, Shahram Izadi, Hrvoje Benko, Francois Guimbretiere, Xiangshi Ren, and Ken Hinckley. 2011. Enhancing naturalness of pen-and-tablet drawing through context sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces.* 83–86.

[104] Wee-Kheng Tan and Yen-Cheng Chen. 2021. Tourists' work-related smartphone use at the tourist destination: making an otherwise impossible trip possible. *Current Issues in Tourism* 24, 11 (2021), 1526–1541.

[105] Ian Towers, Linda Duxbury, Christopher Higgins, and John Thomas. 2006. Time thieves and space invaders: Technology, work and the organization. *Journal of Organizational Change Management* (2006).

[106] Heli Väätäjä and Anssi A. Männistö. 2010. Bottlenecks, Usability Issues and Development Needs in Creating and Delivering News Videos with Smart Phones. In *Proceedings of the 3rd Workshop on Mobile Video Delivery* (Firenze, Italy) *(MoViD '10)*. Association for Computing Machinery, New York, NY, USA, 45–50. https://doi.org/10.1145/1878022.1878034

[107] Jan van der Kamp and Veronica Sundstedt. 2011. Gaze and Voice Controlled Drawing. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications* (Karlskrona, Sweden) *(NGCA '11)*. Association for Computing Machinery, New York, NY, USA, Article 9, 8 pages. https://doi.org/10.1145/1983302.1983311

[108] Daniel Vogel and Patrick Baudisch. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 657–666. https://doi.org/10.1145/1240624.1240727

[109] Jagoda Walny, Sheelagh Carpendale, Nathalie Henry Riche, Gina Venolia, and Philip Fawcett. 2011. Visual thinking in action: Visualizations as used on whiteboards. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2508–2517.

[110] Dakuo Wang, Liuping Wang, Zhan Zhang, Ding Wang, Haiyi Zhu, Yvonne Gao, Xiangmin Fan, and Feng Tian. 2021. "Brilliant AI Doctor" in Rural Clinics: Challenges in AI-Powered Clinical Decision Support System Deployment. 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 697, 18 pages. https://doi.org/10.1145/3411764.3445432

[111] Hao-Chuan Wang, Dan Cosley, and Susan R. Fussell. 2010. Idea Expander: Supporting Group Brainstorming with Conversationally Triggered Visual Thinking Stimuli *(CSCW '10)*. Association for Computing Machinery, New York, NY, USA, 103–106. https://doi.org/10.1145/1718918.1718938

[112] Lauren Wilcox, Jie Lu, Jennifer Lai, Steven Feiner, and Desmond Jordan. 2009. ActiveNotes: Computer-Assisted Creation of Patient Progress Notes. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI EA '09)*. Association for Computing Machinery, New York, NY, USA, 3323–3328. https://doi.org/10.1145/1520340.1520480

[113] Erik Wolf, Sara Klüber, Chris Zimmerer, Jean-Luc Lugrin, and Marc Erich Latoschik. 2019. " Paint that object yellow": Multimodal Interaction to Enhance Creativity During Design Tasks in VR. In *2019 International Conference on Multimodal Interaction.* 195–204.

[114] Yukang Yan, Chun Yu, Yingtian Shi, and Minxing Xie. 2019. PrivateTalk: Activating Voice Input with Hand-On-Mouth Gesture Detected by Bluetooth Earphones. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology.* 1013–1020.

[115] Jackie (Junrui) Yang, Monica S. Lam, and James A. Landay. 2020. DoThisHere: Multimodal Interaction to Improve Cross-Application Tasks on Mobile Devices. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 35–44. https://doi.org/10.1145/3379337.3415841

[116] Mingrui Ray Zhang, Ruolin Wang, Xuhai Xu, Qisheng Li, Ather Sharif, and Jacob O. Wobbrock. 2021. Voicemoji: Emoji Entry Using Voice for Visually Impaired People. 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 37, 18 pages. https://doi.org/10.1145/3411764.3445338

[117] Zheng Zhang, Ying Xu, Yanhao Wang, Bingsheng Yao, Daniel Ritchie, Tongshuang Wu, Mo Yu, Dakuo Wang, and Toby Jia-Jun Li. 2022. StoryBuddy: A Human-AI Collaborative Chatbot for Parent-Child Interactive Storytelling with Flexible Parental Involvement. 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 218, 21 pages. https://doi.org/10.1145/3491102.3517479

[118] Maozheng Zhao, Wenzhe Cui, IV Ramakrishnan, Shumin Zhai, and Xiaojun Bi. 2021. Voice and Touch Based Error-Tolerant Multimodal Text Editing and Correction for Smartphones. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 162–178. https://doi.org/10.1145/3472749.3474742

[119] Maozheng Zhao, Henry Huang, Zhi Li, Rui Liu, Wenzhe Cui, Kajal Toshniwal, Ananya Goel, Andrew Wang, Xia Zhao, Sina Rashidian, Furqan Baig, Khiem Phi, Shumin Zhai, IV Ramakrishnan, Fusheng Wang, and Xiaojun Bi. 2022. EyeSayCorrect: Eye Gaze and Voice Based Hands-Free Text Correction for Mobile Devices. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 470–482. https://doi.org/10.1145/3490099.3511103

[120] Tianshu Zhou and Jingsong Li. 2017. An Intelligent Writing Assistant Module for Narrative Clinical Records based on Named Entity Recognition and Similarity Computation. (2017).

[121] Chris Zimmerer, Philipp Krop, Martin Fischbach, and Marc Erich Latoschik. 2022. Reducing the Cognitive Load of Playing a Digital Tabletop Game with a Multimodal Interface. 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 434, 13 pages. https://doi.org/10.1145/3491102.3502062

[122] Chris Zimmerer, Erik Wolf, Sara Wolf, Martin Fischbach, Jean-Luc Lugrin, and Marc Erich Latoschik. 2020. Finally on Par?! Multimodal and Unimodal Interaction for Open Creative Design Tasks in Virtual Reality. In *Proceedings of the 2020 International Conference on Multimodal Interaction* (Virtual Event, Netherlands) *(ICMI '20)*. Association for Computing Machinery, New York, NY, USA, 222–231. https://doi.org/10.1145/3382507.3418850

A APPENDIX

A.1 Descriptions of Target Diagrams in Study 1

We provided the following natural language descriptions of the target diagrams to the users in Study 1.

A.1.1 Task 1. The carbon cycle diagram indicates how carbon exchanges among different parts of the earth:

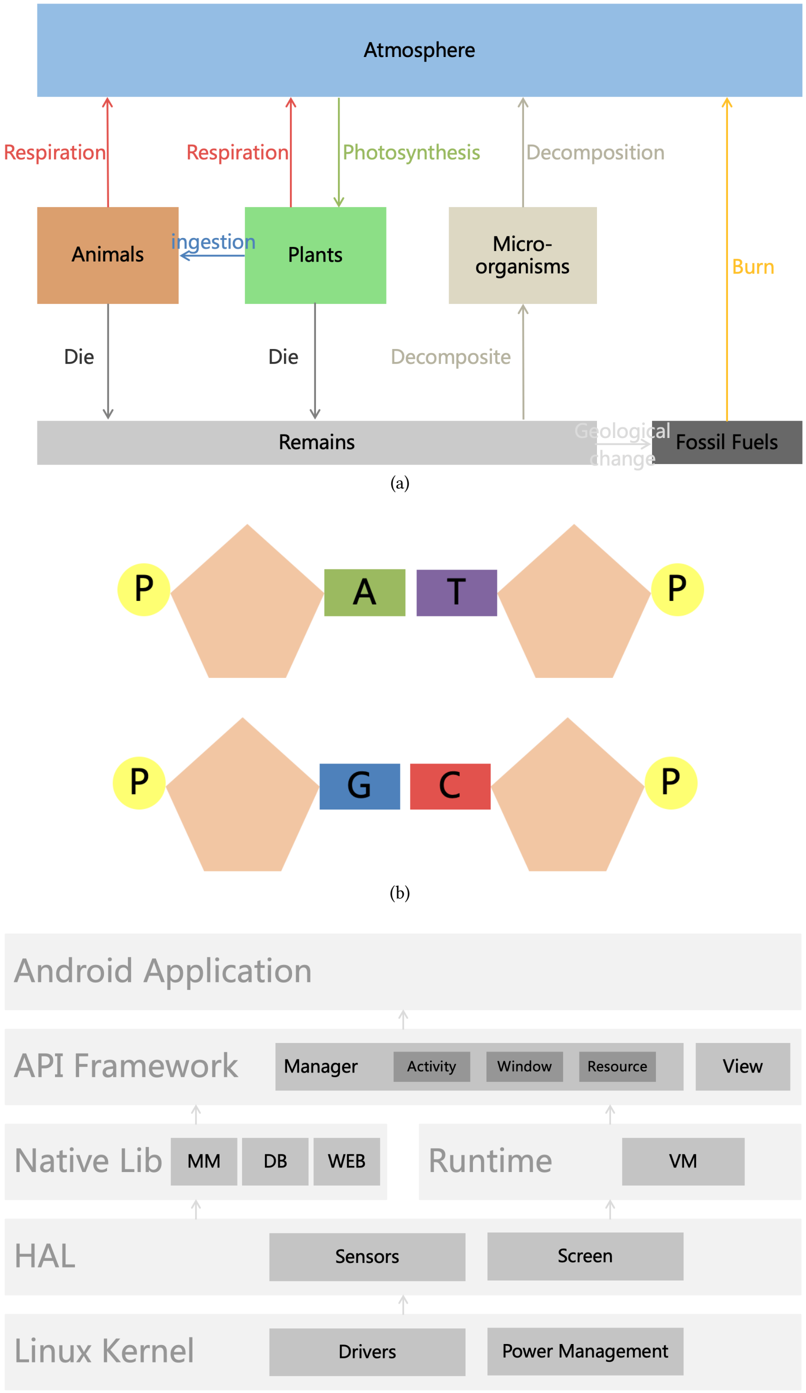
(1) Plants absorb carbon from the atmosphere via photosynthesis.
(2) Plants release carbon into the atmosphere via respiration.
(3) Animals absorb carbon from plants via ingestion.
(4) Animals release carbon into the atmosphere via respiration.
(5) Carbon moves to the remains after animals and plants die.
(6) Microorganisms decomposite the remains and get carbon.
(7) Microorganisms release carbon into the atmosphere via decomposition.
(8) The remains become fossil fuels after thousands of years of geological change.
(9) We burn fossil fuels and release carbon into the atmosphere.

A.1.2 Task 2. This diagram aims to demonstrate (1) the structures of four kinds of deoxyribonucleotides and (2) how they pair with each other.

A deoxyribonucleotide contains (1) a deoxyribose sugar, which is usually represented by a pentagon; (2) a phosphoryl group, which is usually represented by a circle marked "P"; (3) a nitrogenous base, which is usually represented by a rectangle. There are four kinds of nitrogenous bases (A/G/C/T, usually colored differently). The phosphoryl group and the nitrogenous base are connected at two vertices of the deoxyribose sugar (pentagon). These two vertices are not adjacent to each other. Nitrogenous bases G pairs with C and A with T. You should draw the paired bases closely.

A.1.3 Task 3. The Android system contains the following five parts (bottom-up):

(1) Linux Kernel, including Drivers and Power Management.
(2) HAL, including Sensors and the Screen.
(3) Library, including Native Library and Runtime. The Native Library includes MM, DB, and WEB. The Runtime consists of VM.
(4) API framework, including Manager and View. The Manager contains Activity, Window, and Resource.
(5) Android Application.



(a)

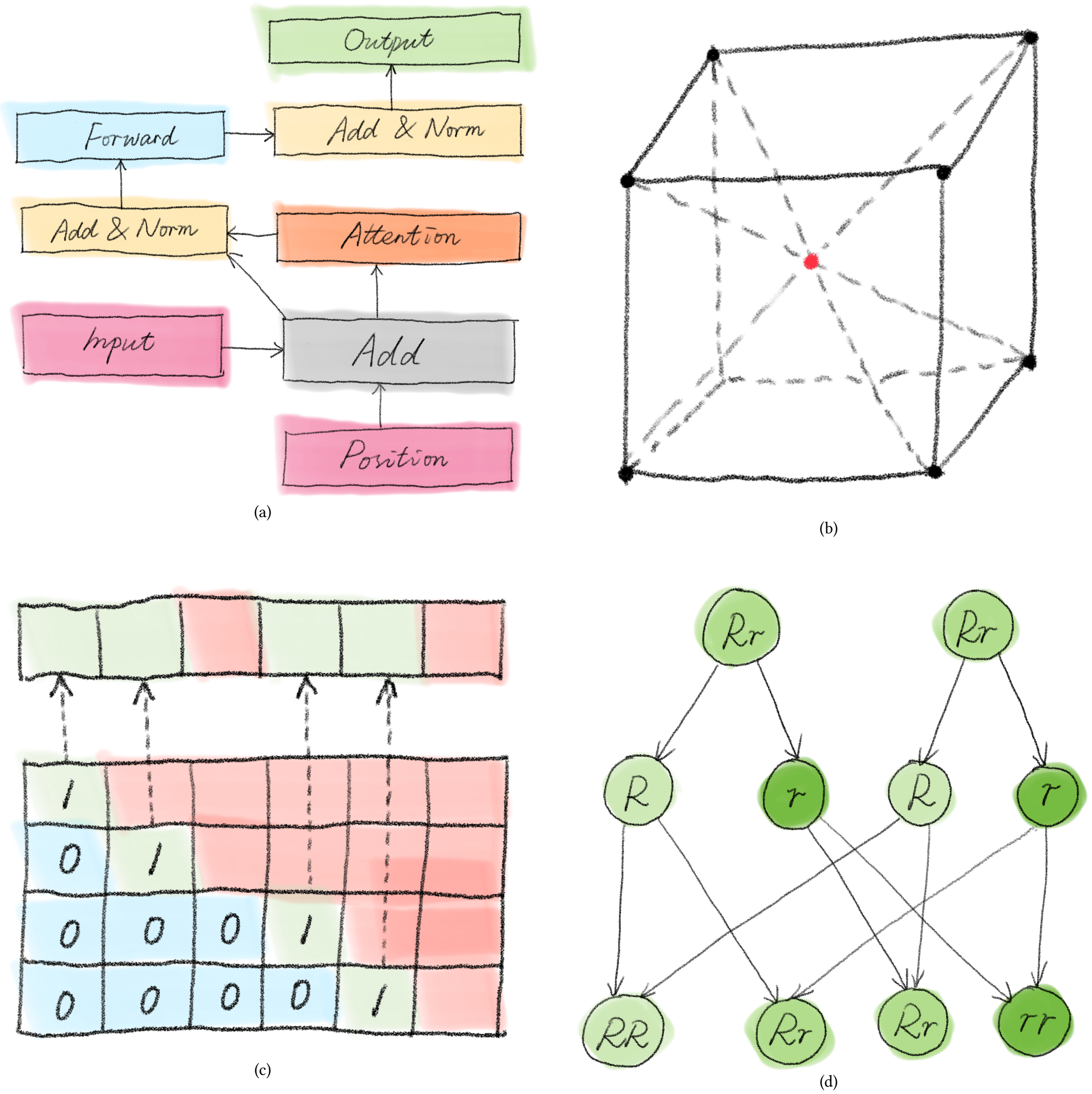
(b)

(c)

Figure 23: Example results of Study 1. (a) Task 1; (b) Task 2; (c) Task 3.

Table 11: The color scheme used in SGDiag. From left to right: lightness decrease. The user can manually adjust the lightness by "make it brighter/darker."

	+4	+3	+2	+1	0	-1	-2	-3	-4	-5
red	#ffebee	#ffcdd2	#ef9a9a	#e57373	#ef5350	#f44336	#e53935	#d32f2f	#c62828	#b71c1c
pink	#fce4ec	#f8bbd0	#f48fb1	#f06292	#ec407a	#e91e63	#d81b60	#c2185b	#ad1457	#880e4f
purple	#f3e5f5	#e1bee7	#ce93d8	#ba68c8	#ab47bc	#9c27b0	#8e24aa	#7b1fa2	#6a1b9a	#4a148c
blue	#e3f2fd	#bbdefb	#90caf9	#64b5f6	#42a5f5	#2196f3	#1e88e5	#1976d2	#1565c0	#0d47a1
cyan	#e0f7fa	#b2ebf2	#80deea	#4dd0e1	#26c6da	#00bcd4	#00acc1	#0097a7	#00838f	#006064
teal	#e0f2f1	#b2dfdb	#80cbc4	#4db6ac	#26a69a	#009688	#00897b	#00796b	#00695c	#004d40
green	#e8f5e9	#c8e6c9	#a5d6a7	#81c784	#66bb6a	#4caf50	#43a047	#388e3c	#2e7d32	#1b5e20
yellow	#fffde7	#fff9c4	#fff59d	#fff176	#ffee58	#ffeb3b	#fdd835	#fbc02d	#f9a825	#f57f17
orange	#fff3e0	#ffe0b2	#ffcc80	#ffb74d	#ffa726	#ff9800	#fb8c00	#f57c00	#ef6c00	#e65100
brown	#efebe9	#d7ccc8	#bcaaa4	#a1887f	#8d6e63	#795548	#6d4c41	#5d4037	#4e342e	#3e2723
grey	#ffffff	#f5f5f5	#eeeeee	#e0e0e0	#bdbdbd	#9e9e9e	#757575	#616161	#424242	#212121
bluegrey	#eceff1	#cfd8dc	#b0bec5	#90a4ae	#78909c	#607d8b	#546e7a	#455a64	#37474f	#263238



(a)

(b)

(c)

(d)

Figure 24: Hand-draw sketches for the four tasks in Study 2.

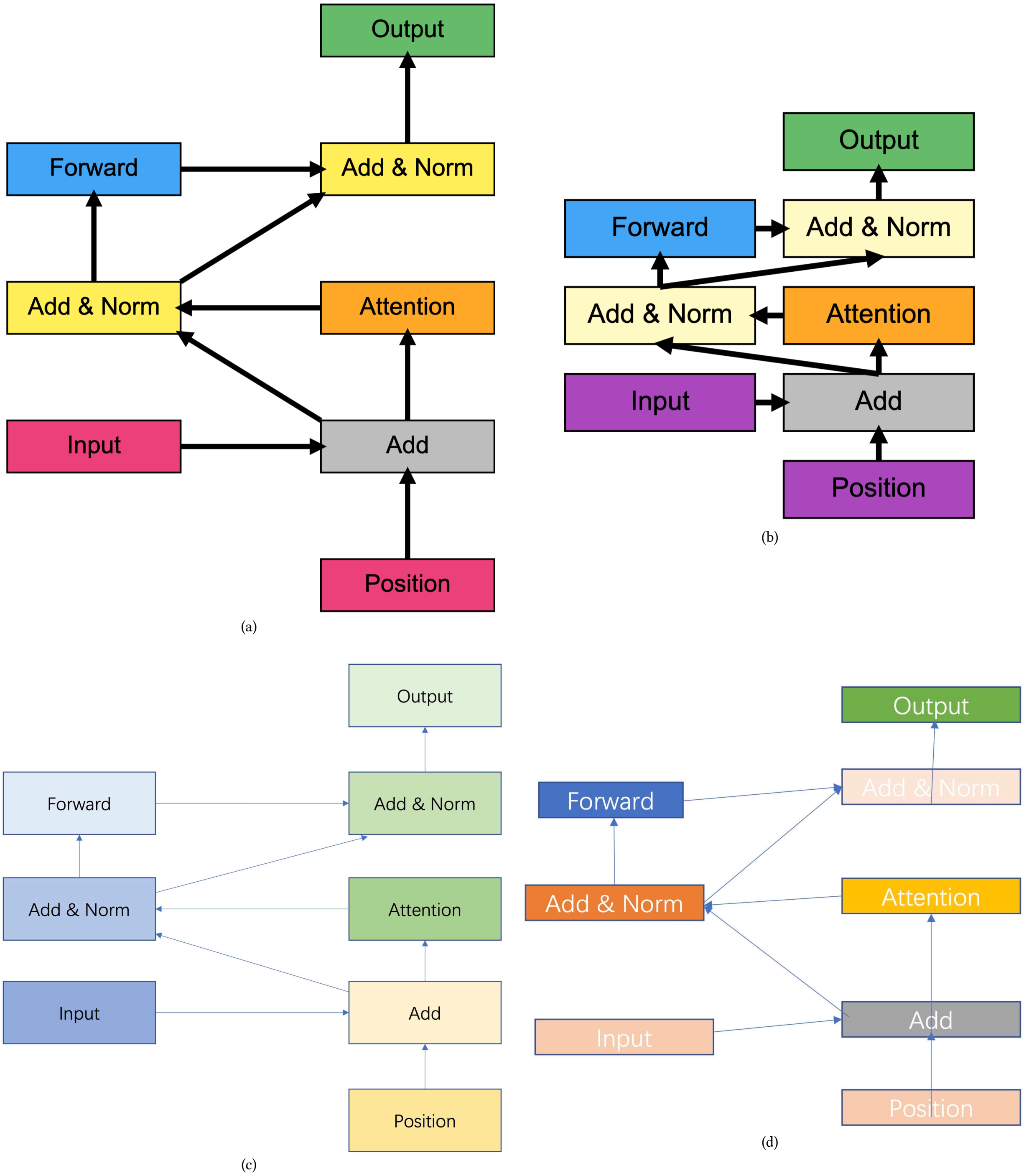


Figure 25: Results of Task 1, Study 2. (a) SGDiag result with the highest score; (b) SGDiag result with the lowest score; (c) PowerPoint result with the highest score; (d) PowerPoint result with the lowest score.

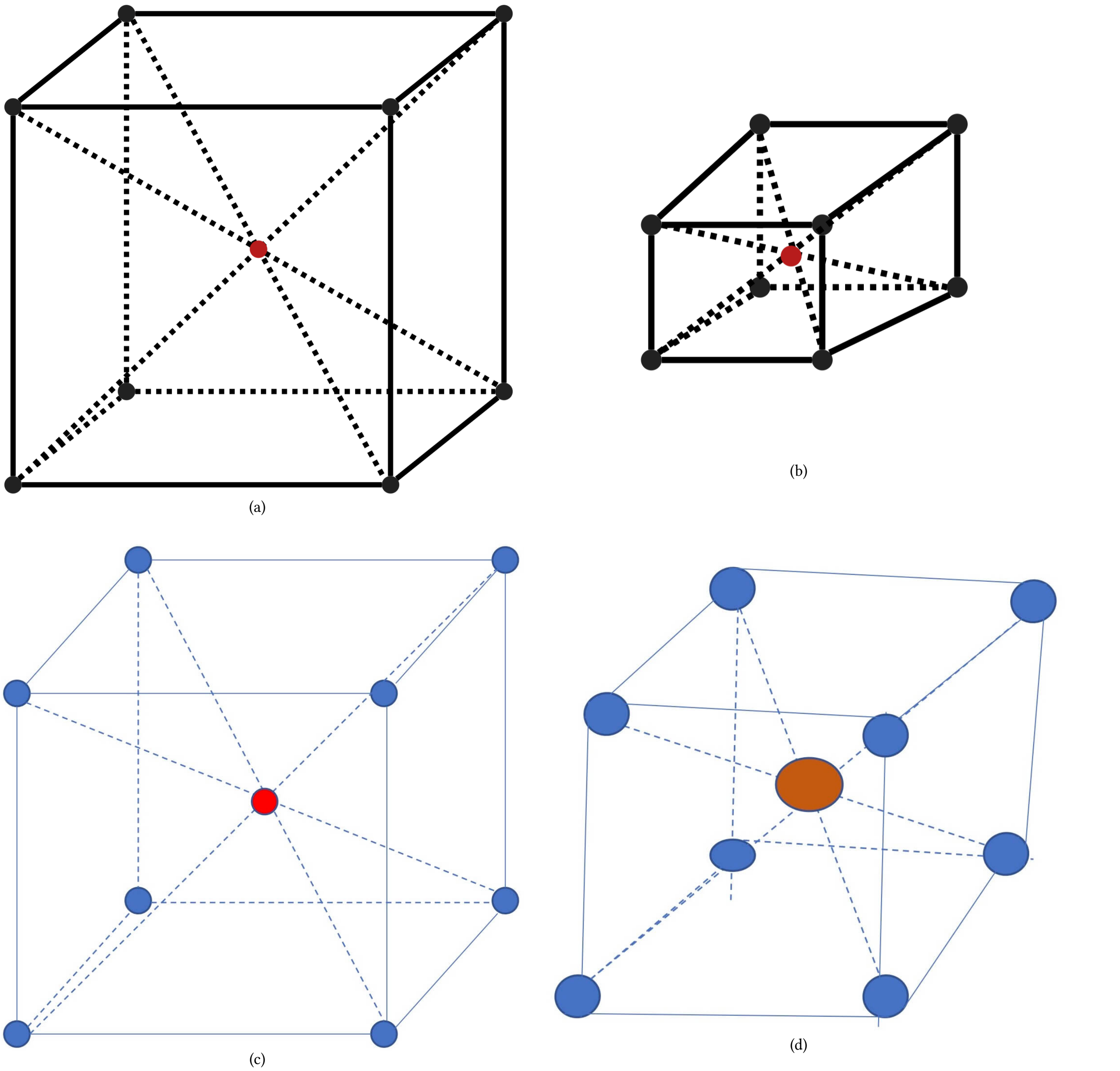


Figure 26: Results of Task 2, Study 2. (a) SGDiag result with the highest score; (b) SGDiag result with the lowest score; (c) PowerPoint result with the highest score; (d) PowerPoint result with the lowest score.

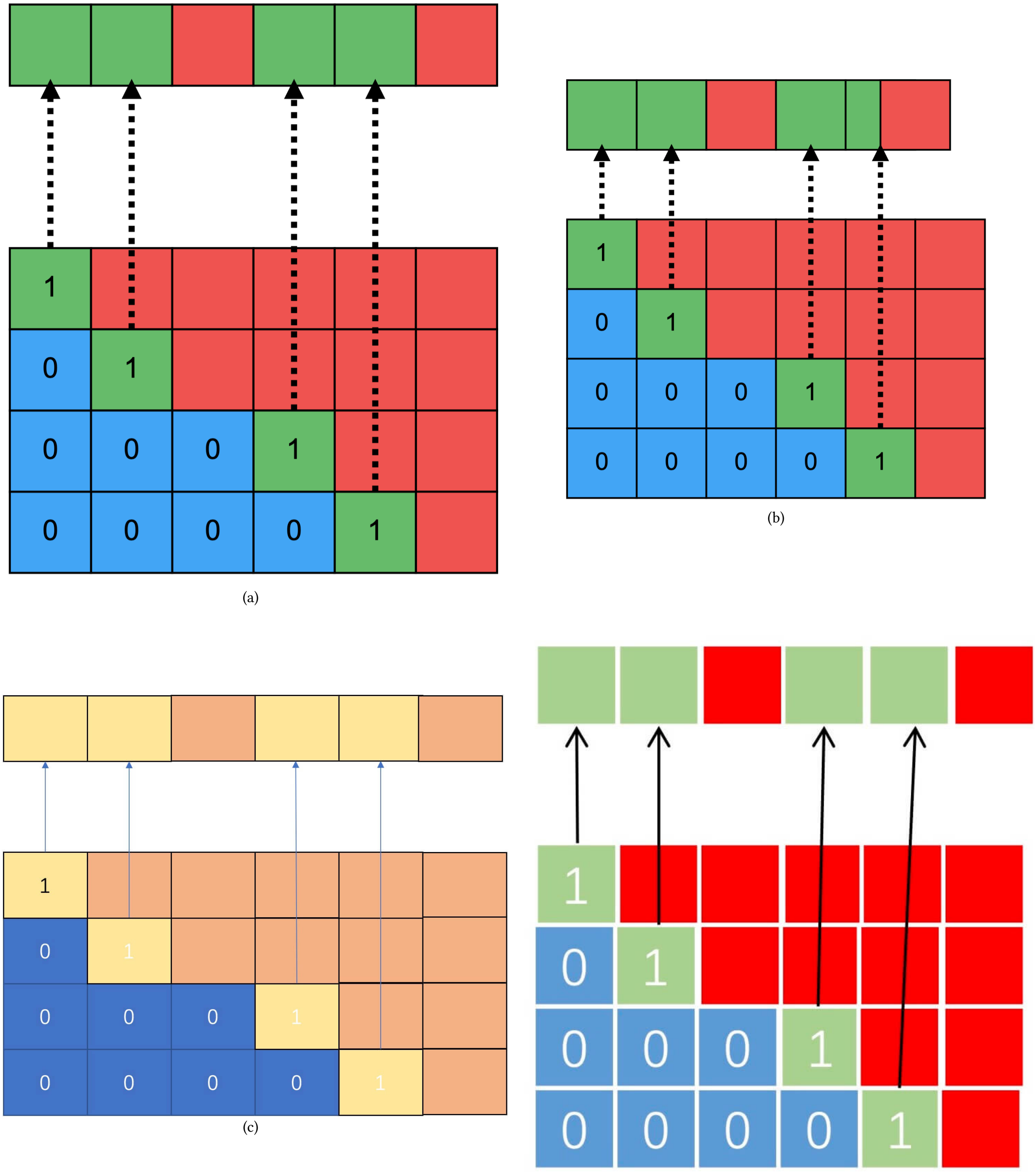


Figure 27: Results of Task 3, Study 2. (a) SGDiag result with the highest score; (b) SGDiag result with the lowest score; (c) PowerPoint result with the highest score; (d) PowerPoint result with the lowest score.

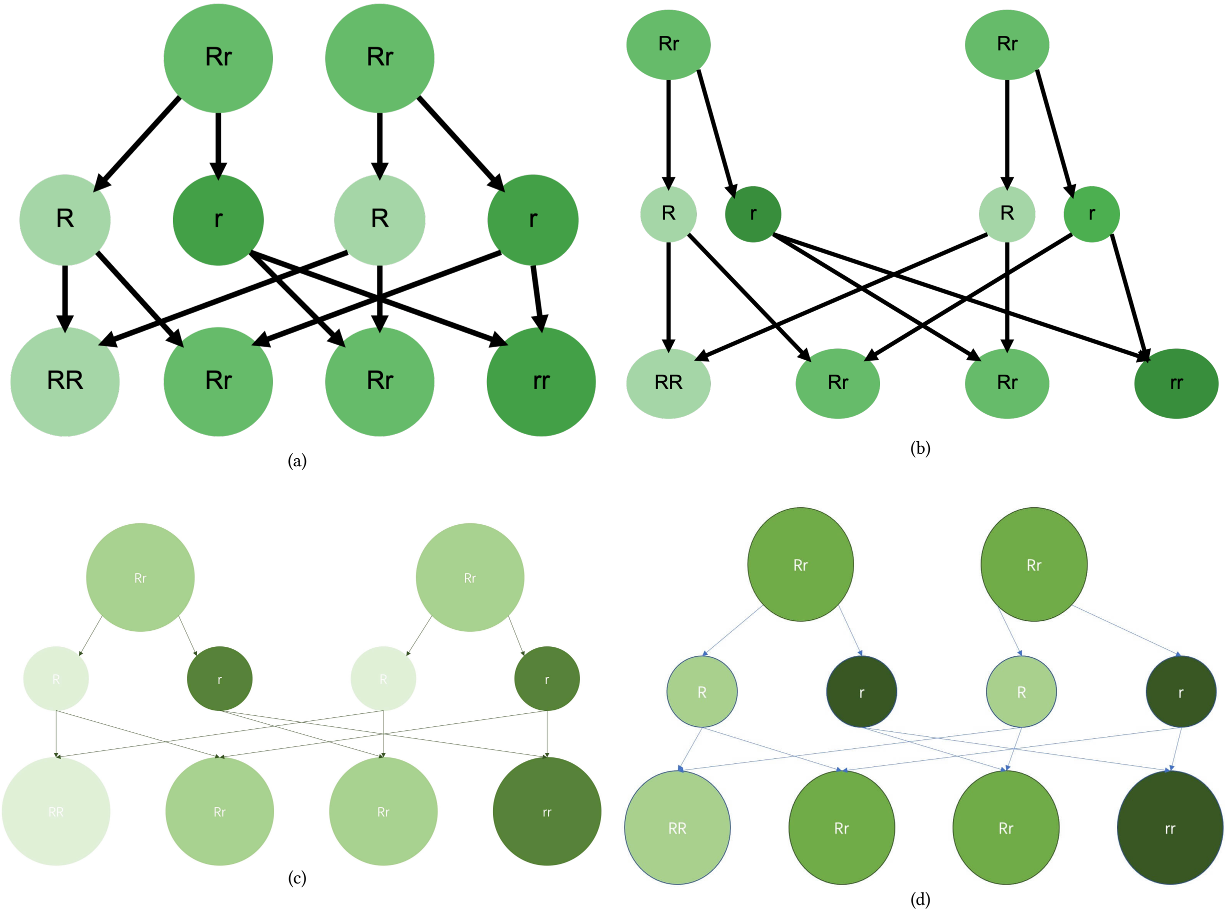


Figure 28: Results of Task 4, Study 2. (a) SGDiag result with the highest score; (b) SGDiag result with the lowest score; (c) PowerPoint result with the highest score; (d) PowerPoint result with the lowest score.