ExpressEdit: Video Editing with Natural Language and Sketching

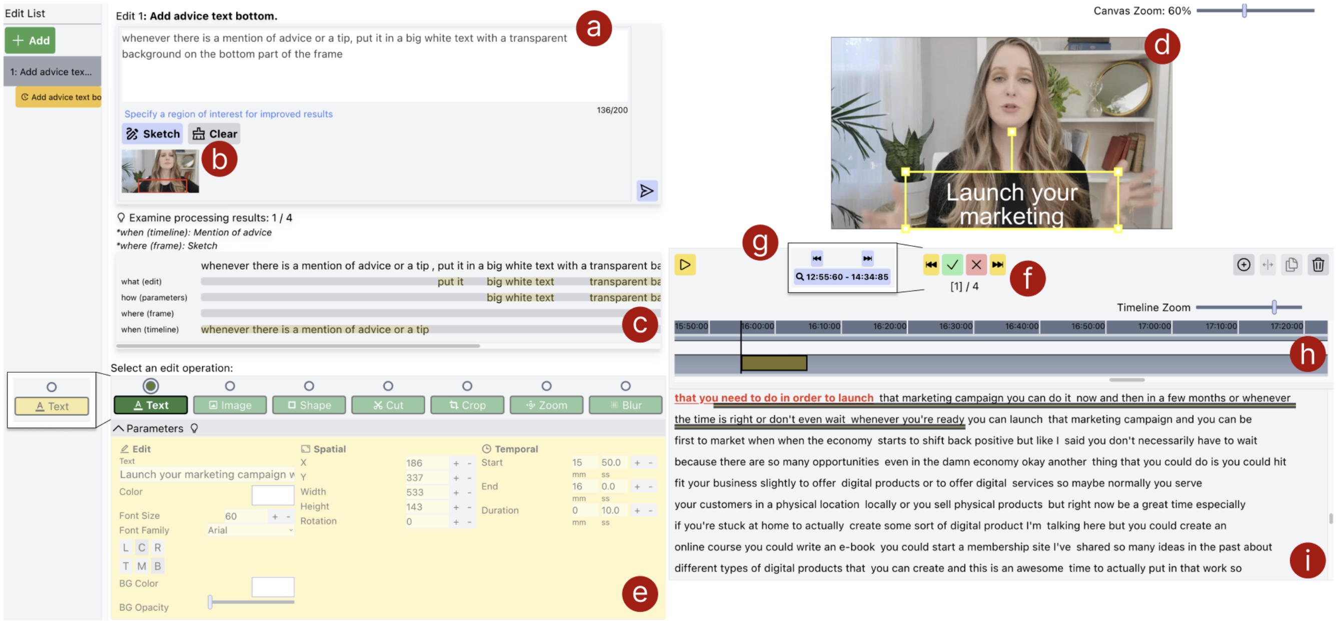
Bekzat Tilekbay
tilekbay@kaist.ac.kr
School of Computing, KAIST
Daejeon, Republic of Korea

Saelyne Yang
saelyne@kaist.ac.kr
School of Computing, KAIST
Daejeon, Republic of Korea

Michal Lewkowicz
michal.lewkowicz@yale.edu
Department of Computer Science,
Yale University
New Haven, Connecticut, USA

Alex Suryapranata
alextio@kaist.ac.kr
School of Computing, KAIST
Daejeon, Republic of Korea

Juho Kim
juhokim@kaist.ac.kr
School of Computing, KAIST
Daejeon, Republic of Korea



Edit 1: **Add advice text bottom.**

whenever there is a mention of advice or a tip, put it in a big white text with a transparent background on the bottom part of the frame

Examine processing results: 1 / 4
when (timeline): Mention of advice
where (frame): Sketch

	whenever there is a mention of advice or a tip , put it in a big white text with a transparent ba
what (edit)	put it big white text transparent ba
how (parameters)	big white text transparent ba
where (frame)	
when (timeline)	whenever there is a mention of advice or a tip

Edit		Spatial		Temporal		
Text	Launch your marketing campaign w	X	186	Start	15 mm	50.0 ss
Color		Y	337	End	16 mm	0.0 ss
Font Size	60	Width	533	Duration	0 mm	10.0 ss
Font Family	Arial	Height	143			
L C R		Rotation	0			
T M B						
BG Color						
BG Opacity						

that you need to do in order to launch that marketing campaign you can do it now and then in a few months or whenever the time is right or don't even wait whenever you're ready you can launch that marketing campaign and you can be first to market when when the economy starts to shift back positive but like I said you don't necessarily have to wait because there are so many opportunities even in the damn economy okay another thing that you could do is you could hit fit your business slightly to offer digital products or to offer digital services so maybe normally you serve your customers in a physical location locally or you sell physical products but right now be a great time especially if you're stuck at home to actually create some sort of digital product I'm talking here but you could create an online course you could write an e-book you could start a membership site I've shared so many ideas in the past about different types of digital products that you can create and this is an awesome time to actually put in that work so

Figure 1: ExpressEdit is a multimodal video editing system that allows users to edit videos with natural language and sketches. (a) The user can input an edit command using natural language in the *edit description box* and (b) optionally specify where in the video frame to apply the edit using the *sketch* function. Once the user submits the edit expression, ExpressEdit analyzes the user's natural language prompt and (c) shows which parts of the user's input it believes correspond to the user's intended edit operation, edit operation parameters, spatial location, and temporal location. Then, the *editor canvas* (d) shows the edit overlaid on the video frame and allows users to adjust the edit's spatial location directly by clicking and dragging. (e) Users can also manually adjust the resulting edit operation as well as its given parameters. (f) Users can navigate through the edit suggestions made by the system and accept or decline, (g) as well as quickly navigate through the video timeline. (h) The *timeline* shows the temporal location of applied edits and edit suggestions. The *transcript* (i) shows an alternative view of the temporal locations of the edits and edit suggestions, which allows users to precisely match the timing of their edits to the speaker's words.

ABSTRACT

Informational videos serve as a crucial source for explaining conceptual and procedural knowledge to novices and experts alike. When producing informational videos, editors edit videos by overlaying text/images or trimming footage to enhance the video quality and make it more engaging. However, video editing can be difficult and time-consuming, especially for novice video editors who often struggle with expressing and implementing their editing ideas.

IUI '24, March 18–21, 2024, Greenville, SC, USA

ACM ISBN 979-8-4007-0508-3/24/03
https://doi.org/10.1145/3640543.3645164

To address this challenge, we first explored how multimodality—natural language (NL) and sketching, which are natural modalities humans use for expression—can be utilized to support video editors in expressing video editing ideas. We gathered 176 multimodal expressions of editing commands from 10 video editors, which revealed the patterns of use of NL and sketching in describing edit intents. Based on the findings, we present ExpressEdit, a system that enables editing videos via NL text and sketching on the video frame. Powered by LLM and vision models, the system interprets (1) temporal, (2) spatial, and (3) operational references in an NL command and spatial references from sketching. The system implements the interpreted edits, which then the user can iterate on. An observational study (N=10) showed that ExpressEdit enhanced the ability of novice video editors to express and implement their edit ideas. The system allowed participants to perform edits more efficiently and generate more ideas by generating edits based on user's multimodal edit commands and supporting iterations on the editing commands. This work offers insights into the design of future multimodal interfaces and AI-based pipelines for video editing.

CCS CONCEPTS

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

KEYWORDS

video editing, human-AI interaction, multimodal input

ACM Reference Format:
Bekzat Tilekbay, Saelyne Yang, Michal Lewkowicz, Alex Suryapranata, and Juho Kim. 2024. ExpressEdit: Video Editing with Natural Language and Sketching. In *29th International Conference on Intelligent User Interfaces (IUI '24), March 18–21, 2024, Greenville, SC, USA.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3640543.3645164

1 INTRODUCTION

Informational videos introduce, explain, or demonstrate conceptual or procedural knowledge [31, 39, 97]. They encompass a broad range of topics such as cooking, health, programming, and craft, and can be produced in various formats (e.g., lecture, tutorial, q&a, demonstration, etc.) [12, 25, 36, 54]. Informational videos have become a popular source of knowledge for the general public due to their rich and engaging content [54], shared in various platforms such as YouTube [70] and learning platforms such as Khan Academy [2], edX [30], and Coursera [46].

However, producing informational videos involves a tedious and complex process that consists of multiple stages such as planning, scripting, filming, and editing [22, 40]. The editing stage is particularly tedious as it requires organizing footage, removing unnecessary parts, and finding and incorporating additional media assets [25] to ensure that the video delivers the intended knowledge in an informative and engaging manner.

While the popular commercial tools for video editing offer the necessary instruments to implement a variety of edits, for novices, these tools are difficult to learn and use, as they require great manual effort and have steep learning curves [18, 51]. A stream of work has been introduced to make video editing less tedious and more efficient for novices such as automatically applying appropriate edits [9, 25, 60, 73, 97] or bootstrapping the editing process by generating videos [22–24, 52, 61, 98, 108]. While these systems allow users to generate video edits efficiently, they do so in an automated or semi-automated manner which provides limited control over the process, which in turn inhibits the user's ability to express their editing intents.

On the other hand, natural language (NL) (e.g., text, speech) and other modalities (e.g., sketching, gestures) have been found to be effective for supporting intent expression and intuitive use for novices. Several multimodal interfaces were introduced for various creative tasks such as image editing [59], web styling [55], data visualization [62], drawing [99], and creative writing [26]. Although there has been work on multimodal interfaces for video production-related tasks like video review [82] or navigation [19, 34, 83, 107], the design of multimodal interfaces for video editing has been under-explored.

We address this gap by investigating how multimodal interfaces can be leveraged in the informational video editing scenario. To explore the benefits and challenges of expressing video editing requests in a multimodal manner, we conducted a formative study with 10 video editors with diverse levels of expertise and collected 176 expressions of video editing requests in the form of NL texts, sketches, and media assets. We found that editors feel comfortable expressing their general editing requests through NL text (176 out of 176) and use sketching on top of the frame (78 out of 176) to indicate specific locations or regions of interest.

The results of the formative study motivated the design of ExpressEdit, a multimodal interactive system for editing informational videos. It supports the expression of video editing requests through NL text and sketching on top of a frame, and is powered by a Computer Vision and Large Language Models-based technical pipeline that interprets and implements the edits by extracting three types of references from the multimodal edit command: (1) temporal location (e.g., *"whenever he mentions laptop"*), (2) spatial location within the frame (e.g., *"near the head"*), and (3) references to edit operations and their parameters (e.g., *"text with the mentioned specifications of the laptop"*). Additionally, ExpressEdit provides a breakdown of the command into the aforementioned types of references, gives clear reasoning for each generated edit (Figure 1c), and allows manual manipulation of those edits.

We evaluated our technical pipeline by constructing a ground truth dataset from 50 selected expressions of video editing requests from the formative study[1]. The reference detection accuracy of our pipeline was higher than 0.68 for the three types of reference we support and yielded a 0.68 recall score on the temporal interpretation, 0.56 mIOU score on spatial interpretation, and 0.82 F-1 score on edit operation interpretation.

To evaluate the effectiveness of ExpressEdit in editing informational videos, we conducted an observational study with 10 novice video editors. We found that ExpressEdit facilitated the expression and implementation of video edits, provided a useful starting point to build upon, and allowed participants to feel more creative.

[1] The dataset and our implementation of the pipeline can be found at https://expressedit.kixlab.org/

Furthermore, the breakdown of the command and manual manipulation of the edits allowed users to iterate on their commands and polish their edits.

Our paper makes the following contributions:

- Formative study results demonstrating the role of NL text and sketching on top of the frame for expressing video editing requests.
- Design of ExpressEdit, a multimodal system that supports editing informational videos by enabling expression of video editing requests through NL text and sketching on top of the frame.
- A CV and LLM-based technical pipeline that understands and implements video edits by parsing and interpreting (1) temporal locations, (2) spatial locations, and (3) editing operations and their parameters from the NL text and sketch command.
- Results of the technical pipeline evaluation and the observational study that demonstrate the effectiveness and usefulness of ExpressEdit in editing informational videos.

2 RELATED WORK

Our work proposes a multimodal video editing system that allows users edit videos with natural language and sketching. We review related work on video editing systems and multimodal interfaces for creative tasks.

2.1 Video Editing Systems

Video editors have access to a plethora of editing tools [4, 7, 28, 45, 71] that support extensive set of functionalities that can be used to express and implement a variety of edits. Unfortunately, these tools require manual effort and have steep learning curves which make editing difficult for novices with limited knowledge [13, 16, 18, 40, 49, 51, 53]. Thus, various systems have been designed to help novices with video editing. Earlier systems relied on metadata to simplify the editing process by providing multiple views with different semantic content and levels of abstraction [16]. Several automatic approaches facilitated the video editing process by automatically applying edits based on predetermined markers [25], placing transitions and cuts in interview videos [9], adding visuals to audio travel podcasts [103], selecting appropriate clips for dialogue-driven scenes [60], adding lyric text to music videos [73], and placing cuts by matching the user's voice-over annotations with relevant segments of the videos [97]. Other systems bootstrapped the editing process by generating videos from documents [22, 23], web pages [24, 52], text-based instructions [108], recipe texts [98], and articles [61] or synthesized talking head videos of puppets [32] and used deep learning methods to automatically generate speech animations [95]. However, these automated approaches restrict the editor's control over the editing process by providing only predetermined input formats for interactions (e.g., markers, annotations), which in turn inhibits the expressiveness.

To alleviate the manual effort in video editing, researchers introduced text-based editing systems to allow users edit the video as if they were editing a text document [9, 33, 33, 42, 43, 60, 81, 100, 103, 105]. Such a method quickly found its application in popular video editing tools such as Adobe Premiere Pro [4], Descript [28], Imvidu [44], and Remotion [5]. Still, text-based editors mainly facilitate navigation and require manual effort to edit the videos end-to-end. With ExpressEdit's multimodal interface, we aim to support novices by letting them easily express their video editing requests while providing sufficient control over the process and alleviating the manual effort required for editing.

2.2 Multimodal Interaction

Multimodal Interactions have been extensively researched by the HCI community and were found effective in reducing the burden and improving efficiency along with user satisfaction [1, 78, 109]. Related work has integrated multimodal interactions in various creative domains such as image editing [59], design [80], creative writing [26], drawing [99], visualization [62, 94], and web styling [55]. These approaches employ direct manipulation, speech-based, gesture-based, and gaze-based interactions to facilitate intent expression and intuitive use, as well as lowering the usability barrier for novices. Similarly, researchers have developed multimodal interactions for video-related tasks such as video review [82], annotation [72], navigation [19, 34, 83, 107], augmenting live storytelling, live presentations, and sports videos [20, 65, 91]. In particular, there has been work to support video editing tasks such as annotating footage with speech and using the annotations to place cuts between footage [97], mapping videos to 2D latent spaces to facilitate pattern identification [66], and utilizing pen gestures in editing process [13].

However, these work fall short of addressing the video editing process holistically and provide limited insights into how to design multimodal interfaces for the combination of natural language (NL) and sketching interactions. Our work addresses this gap by conducting a formative study to learn about how video editors can utilize these modalities and presents a multimodal system ExpressEdit that demonstrates the effectiveness of the approach.

On the other hand, the machine learning community has been active in researching NL-based video generation and editing [8, 17, 38, 41, 63, 86, 87]. However, these work focus on the effectiveness of the AI model and usually support video editing tasks that manipulate the video on a pixel level (e.g., replacing objects, style transfer, etc). In this paper, we focus more on the interaction process with NL and sketching and address video editing tasks that frequently occur in informational videos.

3 FORMATIVE STUDY

To learn about (1) the role of natural language (NL) text and sketching in expressing video editing requests and (2) their use cases in editing informational videos, we conducted a formative study with video editors, where participants were asked to express their edit requests that would improve the informativeness and engagement of a given video. We call these expressions *edit commands*.

3.1 Participants

We recruited 10 video editors (3 females, 7 males, mean age 24.7) with prior experience editing informational videos. We recruited 5 novices who had edited at least two videos and watched informational videos regularly and 5 experienced editors who had edited at least 20 videos and 5 informational videos (Table 1). We recruited both novice and experienced editors to cover a diverse range of edit commands since edit expressions such as attention to detail and vocabulary used can vary depending on the participant's editing expertise and ensured that novice editors are also regular viewers of informational videos so that they have an understanding of the types of edits that should be included in such videos. In the recruitment form, we asked about participants' editing experience, the topics of informational videos they have edited or watched, and their demographic information. The editors were recruited through Upwork [47], a freelancing platform, and university community postings. Novice and experienced participants were compensated with 30,000 KRW (approximately 25 USD) and 50,000 KRW (approximately 40 USD) for a 100-minute study, respectively.

3.2 Study Materials

We chose five archived informational live streams as raw footage for the study. The videos covered various topics, knowledge characteristics (i.e., procedural or conceptual), and the main channel of information (i.e., visual or verbal) (Table 2). We chose archived live streams as they are usually unedited and closely resemble a continuous stream of raw footage, which allows for tasks closer to real-world video editing settings. We assigned the videos to each participant that best fit their interests as closely as possible (Table 1).

To allow participants to express their edit commands in both text and sketch, we used Google Slides [35], a popular slide authoring tool. We chose the tool because of its functionalities of adding text, images, and shapes, which could be used in expressing edit commands. Participants were also allowed to take a screenshot of a frame of the video and sketch over it.

3.3 Procedure

The study was conducted either online through Zoom [48] or in person depending on the participant's preference (Table 1). During the study, the participants were first asked to skim through the given video and its transcript for 30 minutes to become familiar with the content. Then, they were given a quick tutorial on the basic functionalities of Google Slides such as adding text and shapes. For the next hour, participants were tasked to produce 20 semantically unique expressions of edit commands as if they were explaining them to another video editor. We set 20 commands as a target number but did not prolong the study if participants could not reach the amount. They were free to switch between the video and the slides during the study and were asked to put each edit command into a single slide. After participants completed the main task, we conducted a semi-structured interview for about 10 minutes to learn about the participant's experience performing the given task compared to their previous experiences.

3.4 Findings and Analysis

We analyzed the collected edit expressions in terms of their patterns and usages. Below, we summarize the findings from the study.

3.4.1 Expressing edit commands with multi-modalities. Overall, the participants expressed edit commands using various modalities: NL text, sketch, image, and graphics. All of the commands contained NL text to express the edit they want to implement, such as *"18:08 - cut at the point where he starts talking about leeks"*. The participants also added sketches on top of video frames to refer to part of the frame, such as by adding a rectangular shape tool with an accompanying text *"Reduce the prominence of white in **these parts** of the video due to overexposure."* They also added sketches, images, and graphics to describe the content they wanted to add, such as images of an okay-sign icon with the accompanied text *"Add in **hand emoji** for delicious in the top left corner whenever he says the word "delicious" for a split second"*.

Most participants felt comfortable expressing their edit commands with the given tool. For P5 and P9 it was a common practice to either plan out the edits with NL and sketching or communicate the edit descriptions through NL in collaborative settings. However, P10 and P7 noted that it would be cumbersome to articulate all the details of the edit with just a text. P9 also mentioned that he would prefer to work on a tablet for sketching, and P5 wished to use more sophisticated image editing tools (e.g., Adobe Photoshop [3]) to express the exact visual effects he wanted.

3.4.2 Participants consistently referenced moments in the video with NL text. Most frequently, participants specified one or several timestamps in the video where they wanted to apply the edit, resembling how they would apply edits on the timeline in existing video editing systems. However, when participants did not know the exact moments or wanted to refer to multiple moments with the same edit, they opted to give higher-level references to the visual content of the video (e.g., actions, objects, or their descriptions) or verbal content of the video (e.g., transcript or sounds).

3.4.3 Participants used both NL text and sketching on top of the frame to reference the spatial location of the edits. The participants specified a spatial location in NL text (e.g., "top-left corner", "zoom into pan") or sketches on top of the relevant frames. They mainly used visuals to directly illustrate the edit, similar to how they would implement edits on canvas in typical video editing systems. Among 176 multimodal edit commands collected, 97 contained visuals such as sketches, images, and graphics. Out of the 97 visual commands, 78 contained a frame of the video as a reference and a sketch on top of it (e.g., free-form, shapes, images).

3.4.4 Participants used NL text to refer to edit operations and their parameters. There were several types of references to edit operations and their parameters. Most commonly, the participants referred to them by directly including the name of the operation (e.g., cut, text, image, etc.) that is supported in existing video editing systems. Alternatively, they mentioned the main purpose or intended effect of the edit (e.g., highlight, emphasize, focus). In terms of the parameters of the edit operations, they mostly gave general descriptions (e.g., large text, slow zoom) of the intended effects. Occasionally, experienced editors mentioned precise numbers or specific terminology (e.g., "slow down by 50%", "jump cut"). For edits that contained additional media (e.g., text content, image source), participants used references to the transcript or explicitly described the content.

3.4.5 Participants frequently iterated on their edit commands to make them clearer. During the study, we observed that participants frequently revisited and refined their edit commands, for example, to make them more precise or to keep the consistency between

Participant	Experience	Edited inform. videos #	Assigned video	Study format
P1	Novice	0	FV1	In Person
P2	Novice	0	FV2	In Person
P3	Novice	0	FV3	In Person
P4	Novice	0	FV4	Online
P5	Novice	0	FV5	Online
P6	Experienced	12	FV1	Online
P7	Experienced	13	FV2	In Person
P8	Experienced	10	FV3	In Person
P9	Experienced	120	FV4	In Person
P10	Experienced	13	FV5	In Person

Table 1: Information about formative study participants including their experience level, the number of informational videos they edited, the assigned live stream video for the formative study, and the format of the study.

multiple commands. However, they mentioned that the task setting lacked the real-time iterative nature of video editing, where they could repeatedly improve the edits they had implemented.

3.5 Design Goals

Based on the findings from the formative study, we identified three design goals for a system that supports video editing via natural language (NL) text and sketching:

- DG-1: Enable expression of edit commands through natural language and sketching on top of the frame;
- DG-2: Based on the multimodal edit command, support interpretation of temporal locations, spatial locations within the frame, and editing operations with parameters;
- DG-3: Support iteration on multimodal edit commands and manual editing of interpretation results;

4 EXPRESSEDIT: INTERFACE

Following the identified design goals, we designed ExpressEdit, a multimodal video editing system that allows the expression of edit commands through natural language and sketching on top of the frame (NL&S). Our system interprets the user's request in the form of NL&S (DG-1) and suggests a set of edits with temporal location (when in the video), spatial location (where in the video frame), and edit operation & parameters (which edit and how) (DG-2). To better understand the suggested edits and iterate on the NL&S command, users can examine the summary of the system's processing results that shows how each part of the NL part of the command was interpreted by the system and the reasoning behind each suggested edit's temporal and spatial location (DG-3). Additionally, users can manually adjust the suggested edits by the system or create their own edits (DG-3).

4.1 User Scenario

To illustrate how ExpressEdit can be used, let's follow Lia, a businesswoman and a YouTube creator who wants to edit her video about entrepreneurship. She recorded a talking-head video (i.e., a popular style for informational videos that centers on the speaker's face and upper body [33]) that she wants to make more informative and engaging using ExpressEdit.

4.1.1 Creating a new edit. To start editing the video, Lia first uploads her recorded footage with the transcript to ExpressEdit and comes up with the first edit that she wants to implement. She presses the **Add** button on the **Edit list** panel and creates a new layer on top of the video where she can apply her edits. Edits within a single layer will be of a single edit operation and cannot intersect with each other temporally.

4.1.2 Describing the edit with NL & Sketch. Lia decides to first add text captions whenever she mentions valuable advice or tips. In the **Edit description** (Figure 1a), she types *"whenever there is a mention of advice or a tip, put it in a big white text with a transparent background on the bottom part of the frame"*. Additionally, she specifies the exact part of the frame where she wants the text to appear using the **Sketch** function (Figure 1b) and draws the bounding box on the bottom half of the frame. She presses **Enter** to process the NL&S request and gets (1) a breakdown of her command on the **Examine** panel (Figure 1c), (2) a suggested edit operation highlighted by yellow on the **Edit Operation** panel, (3) set of edits on synchronized **Timeline** (Figure 1h) and **Transcript** (Figure 1i) that indicate where the edits should be applied. Additionally, the system summarizes Lia's description and adds a new entry under the current edit in **Edit List** panel that saves her description for future revisits & refinements. This happens every time Lia makes a new description for the edit and the suggestions (e.g., edit operations, edits) will be saved in the corresponding entry. The system automatically chooses the suggested edit operation and matches the player's position to the start of the first suggested edit.

4.1.3 Examining the results. To determine if the system interpreted her NL&S command correctly, Lia examines the breakdown of the NL part of her command, along with the elements of the interface that are highlighted with yellow color. She sees that on the first row of the breakdown **what (edit)** the *"text"* is outlined and notices that the *Text* operation was automatically chosen in the **Edit operation** panel. In the second row **how (parameters)**, the part of the NL command with a description of the appearance of the text is highlighted *"a big white text with a transparent background"*. She also confirms that the edit parameters are set appropriately by looking at the **Editor Canvas** (Figure 1d) and **Parameters Panel** (Figure 1e). The part of the NL command referencing the edit's spatial location within the frame is highlighted on the third row **where (frame)**. The outlined text *"the bottom half of the frame"* and the position of the edit on the **Editor Canvas** assures her that the location was interpreted correctly. She also reads the quick reasoning for the spatial location on the top of the breakdown rows which says *Sketch* indicating that her sketch was used to decide the location of her edit. Lastly, Lia sees that the reference *"whenever there is a mention of advice or a tip"* is outlined on the fourth row labeled **when (timeline)** and glances at the snippet of the transcript of the video in the **Transcript** panel that highlighted the exact moment where she was talking about "marketing campaign". She also reads the quick reasoning for suggesting the edit on the top of the breakdown rows: *"Mention of advice"*. Thus, she ensured that the NL&S command was interpreted as she expected.

4.1.4 Manual Manipulation & Editing. To go through all the suggested edits in the video, Lia jumps between them with **Previous**

Video	Topics	Knowledge Characteristics	Content Format	Title
FV1	cooking	procedural	visual & verbal	At Home for the Holidays with Gordon Ramsay [89]
FV2	educational	procedural	verbal	Language Learning Live Stream [67]
FV3	programming	procedural	visual & verbal	Livestream: Getting Started with C++ (Episode 1) [27]
FV4	health	conceptual	verbal	Surgeon does Live QA | Hair Loss Awareness Month [68]
FV5	product review	conceptual	visual & verbal	Microsoft Surface Go - Classic LIVE Unboxing [96]

Table 2: The table shows the information for live-stream videos selected for the formative study including topics, knowledge characteristics, content formats, and titles with links.

and **Next** buttons (Figure 1g). She glances at the transcript for each edit and judges whether to insert a text on the video in that segment or not and expresses her decision with **Accept** and **Reject** buttons (Figure 1f). After making all the decisions, she is left with four text edits that are applied to the video. She remembers that there was one more important piece of advice missed by the initial set of suggested edits, so she roughly navigates to the part of the video where she thinks the segment is in the **Timeline** and presses the **Search more** button (Figure 1g) that gives her the exact moment when she mentions the advice. She accepts the suggested edit and goes on to manually adjust its parameters. She specifies the exact temporal boundaries of the edit by dragging them on the **Transcript**, then adjusts the position and size of the text on the **Editor Canvas**, and slightly rephrases the text content in the **Parameters** panel. After finalizing the edit, she moves on to the next edit that she has by pressing the **Next** button that now jumps between the applied edits.

4.2 Edit operations

ExpressEdit supports 7 edit operations: (1) text insertion, (2) image insertion, (3) shape insertion (circle, rectangle, star), (4) cutting out segments of the video, (5) zooming in/out, (6) cropping the video, and (7) blurring the video. These were the 7 visual edit operations that formative study participants frequently mentioned in their edit commands. We decided to focus only on visual edit operations as they cover the important types of parameters (temporal, spatial, edit-specific) that commonly appear in other kinds of edits (e.g. audio-related edits, coloring edits). We believe that this set of edit operations effectively demonstrates the feasibility of implementing various kinds of edits based on NL&S commands.

4.3 Implementation

ExpressEdit is implemented as a Web-based React [74] application powered by KonvaJS [57] for canvas manipulations and MobX [75] for state management. The backend server was implemented using Flask [79], which hosted the videos along with their transcript and processed users' requests. All the videos and transcripts that were used in this work were obtained from YouTube with youtube-dl package [106] for Python.

5 EXPRESSEDIT: PIPELINE

Based on the findings from our formative study, we designed our pipeline to interpret natural language edit commands. We also support sketching on top of the frame to allow users more effectively convey the spatial location for a particular edit along with their natural language command. We also performed technical evaluation of the pipeline by constructing a ground truth dataset from 50 multimodal edit commands selected from the formative study.

5.1 Video Pre-processing

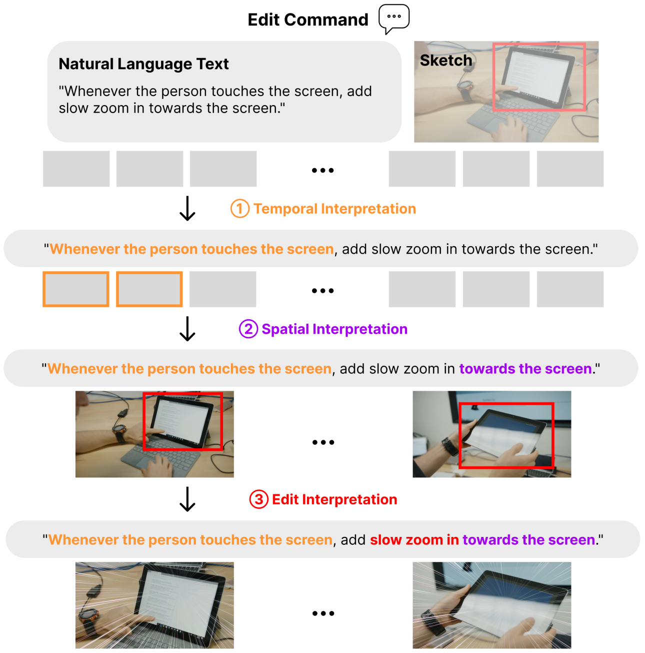
In order to facilitate the real-time system interactions, we perform pre-processing to extract frame-level and clip-level metadata from a video that ExpressEdit uses to reason about video context during the interaction (see Stage 0 in Figure 3). For the frame-level data, we sample video frame every 1.0 seconds and use the Segment Anything model [56] to run automatic instance segmentation at two levels of granularity. We then threshold the segmentations based on instance crop size to remove instance masks that are too small. For the clip-level metadata, we run activity recognition on 10-second clips using InternVideo [101]. We also employ an image captioning model BLIP-2 [64] to generate textual descriptions of the video contents and list the most salient objects in the scene at each second in the 10-second intervals. We then use GPT-3.5 [10] to summarize the resulting captions across the 10-second clips to remove redundant information about the visual content.

5.2 Parsing Edit Command

We first use GPT-4 [77] to parse the NL command and divide the it into the following types of references as illustrated in Stage 1 of Figure 3:

(1) **Temporal reference**: any information in the NL command that could refer to a segment of the video;
(2) **Spatial reference**: any information in the NL command that could refer to location or region in the video frame;
(3) **Edit Operation reference**: any information in the NL command that could indicate an edit operation to use;
(4) **Edit Parameter reference**: any information in the NL command that could refer to specific parameters of edit operation that was determined;

We then use GPT-4 to classify the edit operation that is most suitable for the request based on our system's available operations: "text", "image", "shape", "blur", "cut", "crop", or "zoom." These references are then further interpreted and linked to sections within the video as outlined in the following sections.



"Whenever the person touches the screen, add slow zoom in towards the screen."

"Whenever the person touches the screen, add slow zoom in towards the screen."

"Whenever the person touches the screen, add slow zoom in towards the screen."

"Whenever the person touches the screen, add slow zoom in towards the screen."

Figure 2: Given a text and sketching edit command, our computational pipeline interprets its temporal, spatial, and edit operation and parameter references to implement the edit.

5.3 Temporal Interpretation

In the next stage of the pipeline, the temporal references are decomposed into the following categories as seen in Stage 2 (Figure 3):

1. *Positional* references can be either distinct timecodes or abstract temporal designations (e.g., "intro", "ending");
2. *Transcript-based* references constitute either direct or indirect references to the video's transcript;
3. *Video-based* labels are assigned to descriptions that refer to actions or visual descriptions specific to the video;

Given the volume of video metadata, if the label is either *transcript-based* or *video-based* we filter the 10 most relevant temporal segments of the textual clip-level metadata using cosine similarity along the dense captions and transcript segments, respectively using SentenceTransformers framework [90]. We then use GPT-4 to compile all segments of the video that match the temporal reference and pass the candidate segments further along the pipeline.

5.4 Spatial Interpretation

Upon completion of Stage 2, where we obtain a list of candidate temporal segments, Stage 3 initiates the parsing of spatial references. As shown in Figure 3, we categorize these references as:

1. *Visual-Content Dependent* references to specific to objects, elements, or regions within the video frame;
2. *Visual-Content Independent* references to specific locations or positions that are relative to the frame, but not contingent on the video's visual content.

For segments with "visual-content dependent" information, we extract representative frames from the center of the time-coded range. We then obtain the corresponding instance crops from the frame-level metadata for each frame. If the user provides a sketch, the text and sketch are encoded into a shared embedding space alongside all instance crops using CLIP [88]. Without a user-provided sketch, only the textual reference and instance crops are encoded in the embedding space. The spatial location within the frame that aligns most closely with the parsed command and/or sketch is determined by finding the instance crop having the highest cosine similarity to the user's input.

In the absence of "visual-content dependent" references, if a sketch is provided, it becomes the candidate spatial location. If neither is available, the full frame is returned as the final spatial location. Additionally, if the NL command contains only "visual-content independent" spatial references, we employ GPT-4 to refine and resize the region of interest based on the command and frame boundaries.

5.5 Edit Operation and Parameters Interpretation

In Stage 4, as illustrated in Figure 3, we leverage the relevant segments from the NL command to modify the parameters corresponding to each predicted edit operation. These modification requests can be grouped into three categories:

- *Explicit* specifications for parameters like "12px" or "Introduction";
- *Relative* adjustments relative to current settings such as "5 seconds longer" or "10% less";
- *Abstract* (i.e., general) directives that don't have specific values associated to them, like "shorter" or "longer";

Particular emphasis is given to operations involving text and images. For these operations, we provide the command, its context, and the relevant video content to guide the generation process. In the case of text, this aids in determining the textual display within the video. For images, it assists in formulating an appropriate search query to source the required images for the video.

5.6 Implementation

Our implementation of the pipeline can be divided into an offline pre-processing component and an online component. While the offline component is run locally with all the necessary models, the online component was implemented as a Flask server where we used LangChain framework [58] to perform edit command processing. The prompts used for each stage of the pipeline can be found in Appendix C

5.7 Pipeline Evaluation

We evaluated the performance of our CV and LLM-based pipeline through comparison with the ground truth dataset that consisted of 50 multimodal edit commands selected from the formative study. Table 3 shows the summary of the results for parsing accuracy of the NL commands into (1) temporal, (2) spatial, (3) operational, and (4) parametric parts; and interpretation accuracy for each type of

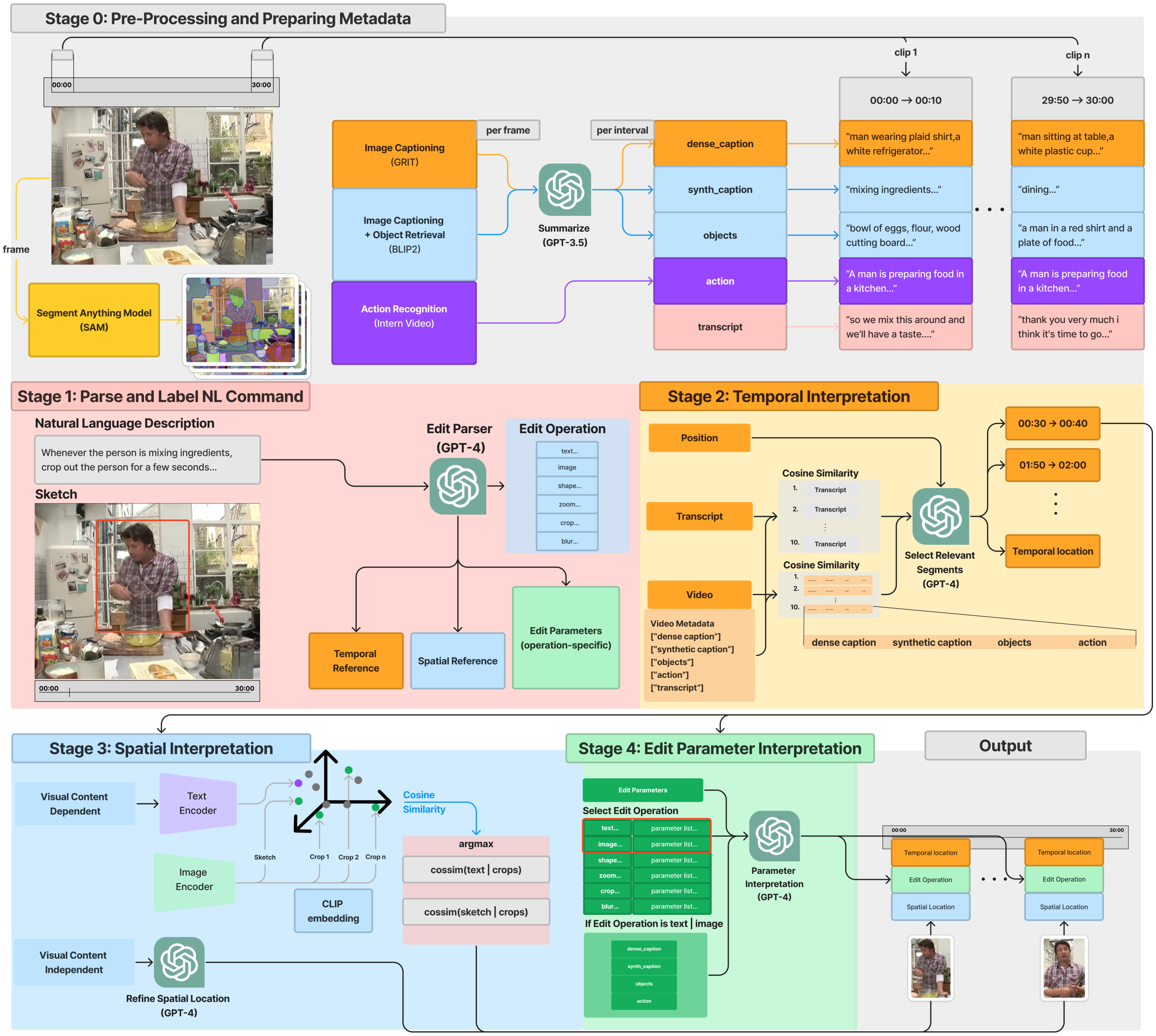

Stage 0: Pre-Processing and Preparing Metadata
00:00
30:00
frame
Segment Anything Model (SAM)
Image Captioning (GRIT)
Image Captioning + Object Retrieval (BLIP2)
Action Recognition (Intern Video)
per frame
per interval
Summarize (GPT-3.5)
dense_caption
synth_caption
objects
action
transcript
clip 1
clip n
00:00 → 00:10
29:50 → 30:00
"man wearing plaid shirt,a white refrigerator..."
"mixing ingredients..."
"bowl of eggs, flour, wood cutting board..."
"A man is preparing food in a kitchen..."
"so we mix this around and we'll have a taste...."
"man sitting at table,a white plastic cup..."
"dining..."
"a man in a red shirt and a plate of food..."
"A man is preparing food in a kitchen..."
"thank you very much i think it's time to go..."
Stage 1: Parse and Label NL Command
Natural Language Description
Whenever the person is mixing ingredients, crop out the person for a few seconds...
Sketch
00:00
30:00
Edit Parser (GPT-4)
Edit Operation
text...
image
shape...
zoom...
crop...
blur...
Temporal Reference
Spatial Reference
Edit Parameters (operation-specific)
Stage 2: Temporal Interpretation
Position
Transcript
Video
Video Metadata ["dense caption"] ["synthetic caption"] ["objects"] ["action"] ["transcript"]
Cosine Similarity
1. Transcript
2. Transcript
10. Transcript
Cosine Similarity
dense caption
synthetic caption
objects
action
Select Relevant Segments (GPT-4)
00:30 → 00:40
01:50 → 02:00
Temporal location
Stage 3: Spatial Interpretation
Visual Content Dependent
Text Encoder
Image Encoder
Sketch
Crop 1
Crop 2
Crop n
CLIP embedding
Cosine Similarity
argmax
cossim(text | crops)
cossim(sketch | crops)
Visual Content Independent
Refine Spatial Location (GPT-4)
Stage 4: Edit Parameter Interpretation
Edit Parameters
Select Edit Operation
text...
image...
shape...
zoom...
crop...
blur...
parameter list...
If Edit Operation is text | image
dense_caption
synth_caption
objects
action
Parameter Interpretation (GPT-4)
Output
00:00
30:00
Temporal location
Edit Operation
Spatial Location

Figure 3: Our computational pipeline can be split into offline and online components. Stage 0 represents the video pre-processing stage. Stages 1 - 4 represent the online components that use GPT-4 for NL command parsing and a CLIP module for the interpretation of spatial references specifically in Stage 2. The diagram illustrates the pipeline at inference time assuming all the video metadata has been generated in the pre-processing stage — processing a user's natural language and sketch input to generate edit suggestions throughout the timeline.

reference. For all the experiments we used *gpt-4-0613* with temperature 0.0.

5.7.1 Ground Truth construction. To construct our ground truth dataset, we selected 50 expressions of editing requests (out of 179) from our formative study dataset. Our inclusion criteria were as follows:

(1) The edit request's operation reference maps to one of the system's supported editing operations.
(2) The edit request's temporal reference is either *positional*, *transcript-based*, or *video-based*.
(3) The edit request is fully self-contained. In other words, it does not reference other edit commands.

Reference Type	Parsing Performance Average (STD)	Interpretation Performance Average (STD)			
		F1	Precision	Recall	mIOU / mIOU (T=0.1)
Temporal (when in the video)	0.79 (0.27)	0.55 (0.40)	0.52 (0.40)	0.68 (0.43)	-
Temporal (when in the video) *margin=10s*	-	0.57 (0.38)	0.56 (0.39)	0.71 (0.41)	-
Spatial (where in the frame)	0.68 (0.30)	-	-	-	0.56 (0.40) / 0.86
Edit Operation	0.72 (0.25)	0.82 (0.32)	0.84 (0.32)	0.83 (0.33)	-
Edit Parameters	0.74 (0.20)	-	-	-	-

Table 3: Performance of the CV&LLM-based technical pipeline in terms of parsing and interpretation of (1) temporal, (2) spatial, (3) edit operation, and (4) edit parameters. We calculated the spatial interpretation performance separately with ground truth edit segments instead of predicted segments.

Two authors analyzed each multimodal edit command to extract parts that refer to temporal locations, spatial locations, edit operations, and parameters from the textual component. Then, they interpreted each reference by (1) detecting moments in the video, (2) choosing appropriate spatial locations within the frame, and (3) selecting suitable edit operations. The full analysis and interpretation procedure is listed in Appendix A

5.7.2 Metrics. To measure the performance of the parsing and interpretation stages of the pipeline we calculate cosine similarities, F-1 scores (along with precision and recall), and mean Intersection-Over-Union (mIOU) between predicted results and ground truth.

For each type of reference (i.e., temporal, spatial, operational, parameter) we compute the text embeddings of both the predicted parsing result and the ground truth and calculate the cosine similarity between them with SentenceTransformers framework [90].

To assess the accuracy of the interpretation of the temporal references, we compute F-1 scores between predicted and ground-truth segments similar to Yang et al. [104] Since our pipeline mainly predicts short segments of the video with a 10-second duration and the small time differences between the predicted segment and ground truth are insignificant (as users usually refine the boundaries anyway), we count the predicted segment as true positive if it intersects with the ground truth segment. Generally, it is also useful if the predicted segment is near the desired moment in the video, thus we also report the F-1 scores with relaxed true positive criteria: the distance between the predicted segment and ground truth is within 10 seconds.

As spatial interpretation happens for each predicted segment, we isolated the performance of the spatial interpretation by computing the mIOU between the predicted and ground truth locations for the ground truth segments of each data point. Since the ground truth bounding boxes of locations are approximate and the exact width and height of the boxes usually depend on the content, we also report the ratio of mIOU scores larger than 0.1 which ensures that there is some intersection between the prediction and ground truth.

Since our ground truth dataset contains edit commands that required a combination of edit operations, our pipeline is designed to suggest multiple of them. Thus, we calculate F-1 scores to measure the edit operation interpretation performance. Since all the edit operations are from the set of seven operations that ExpressEdit supports (Section 4.2), we consider the prediction to be true positive if it exactly matches one of the operations in the ground truth.

5.7.3 Results. Our pipeline achieved reasonable average cosine similarity scores for all the reference types that the system supports. The parsing spatial reference parts from the NL command have the lowest score and highest variability among all (M=0.68, STD=0.30), which might be due to the fact that we had fewer data points with NL spatial references (22 out of 50) and ground truth contained generic references such as "over the video" and "on the screen" that did not contribute concrete specifications to the spatial location. The scores for parsing temporal references (M=0.79, STD=0.27), edit operation references (M=0.72, STD=0.25), and parameter references (M=0.74, STD=0.20) have relatively similar standard deviations with operation references having the lower score. The average F-1 score for interpreting edit operation references is 0.82 (STD=0.32) and the precision and recall are 0.84 (STD=0.32) and 0.83 (STD=0.33), respectively. For interpreting temporal references, the recall is 0.68 (STD=0.43) and 0.71 (STD=0.41) with a 10-second margin. Although the F-1 score (M=0.55, STD=0.40) and precision (M=0.52, STD=0.40) are much lower, they are less important compared to recall as finding and making sure that all the important moments in the video are covered (recall) is usually much more challenging and time-consuming compared to evaluating if a short snippet is relevant (precision). Thus, even with lower precision but reasonable recall, our pipeline can still be useful for video editing tasks ExpressEdit supports. Finally, the average mIOU for interpreting spatial references is 0.56 (STD=0.40) and the ratio of mIOUs larger than 0.1 is 0.86, which shows that the pipeline is generating reasonable bounding boxes for the edits.

6 USER EVALUATION

ExpressEdit's main features are designed to enhance the expression of edit commands and facilitate their implementation. We conducted an observational study with novice video editors, who are more likely to face challenges in expressing and implementing their editing ideas. In particular, we aim to address the following research questions:

- RQ-1: How do users edit videos with ExpressEdit?
- RQ-2: How well does ExpressEdit understand and implement multimodal commands?

- RQ-3: How useful is ExpressEdit for novices in editing informational videos?

6.1 Participants

We recruited 10 participants (5 females, 5 males, mean age 20.4) who identified themselves as novice editors through university community service postings. The reported number of years of experience and the number of videos each participant edited prior to the study are shown in Table 4. We also verified that the participants are regular viewers of informational videos and can write and speak in English fluently, as our system accepts NL commands in English.

For the study, we provided two video footage and reference video pairs, each with similar topics and characteristics. Participants were tasked with making the given video footage more engaging and informative using their own ideas or ideas derived from the provided reference video. Similar to videos for the formative study, the selected footage videos varied in terms of (1) knowledge characteristics (procedural/conceptual), and (2) formats (visual/verbal), as outlined in Table 5. The footage video contained a minimal number of edits, while the reference video contained a range of edit ideas relevant to the footage and supported by our system.

6.2 Procedure

The study was conducted in person and the entire session (audio and screen) was recorded. After the introduction of the study, the participants watched a 10-minute-long tutorial on the system and had 10 more minutes to complete a set of simple tasks where they were asked to use each edit operation we support at least once and make at least two NL&S edit commands to the system. After the tutorial, participants skimmed through the assigned footage and the reference for 10 minutes and performed the main task of improving the video by making it more engaging and informative for 40 minutes. They were encouraged to think aloud and describe the intentions behind their actions as they were implementing edits. Right after the main task, participants completed the post-survey and we interviewed them about their experience using the system, advantages & disadvantages of the system, and their previous experiences using other editing systems compared to using ExpressEdit. The study lasted 100 minutes, and each participant was compensated with 50,000 KRW (approximately 40 USD).

6.3 Collected Data

The post-survey contained 7-point Likert-scale questions about participants' confidence in using the system and the perceived helpfulness of the main features of the system. Additionally we also inquiry participants about their experience with the AI tool [102] and more general System Usability Scale (SUS) [92]. We also rated ExpressEdit in terms of CSI [21] for editing informational videos by novice editors. We did not use paired-factor comparison but rather measured the scores from 7-point Likert-scale responses to the questions. We also skipped the "Collaboration" scale as our system is not designed with collaboration features in mind. Finally, we ask about the perceived load of the task with the NASA-TLX survey [37].

To analyze the workflows of the participants while editing an informational video, we logged participants' interactions with the system. We labeled each time segment between interaction logs based on stages of the video editing process that ExpressEdit supports (i.e., ideating, describing, examining, and manual editing). Namely, "ideating" represents time segments where users were switching between edit layers in "Edit List" and history points, which may suggest that participants are actively ideating on what edit to request or implement. The "describing" label was given to segments where we captured interactions with the "Edit Description" panel (e.g., describing the command with NL, sketching on top of the frame). The "examining" segments occur after the user receives the processing results of their multimodal command and starts examining whether to accept or reject the suggestions. Finally, the "manual editing" label represents segments where the user actively manipulates the edits by using the "Canvas", "Timeline", "Transcript", or the "Parameters" panel.

We decided not to evaluate the outcome videos, as we did not expect participants to complete the task within the given time limit. Rather, our analysis focused more on the video editing process and the experience of the participants using ExpressEdit.

Participant	Years of Experience	Edited videos #	Assigned video
P1	0	2	EV1
P2	1	6	EV1
P3	1	3	EV1
P4	3	5	EV1
P5	2	2	EV1
P6	5	5	EV2
P7	2	5	EV2
P8	3	4	EV2
P9	1	15	EV2
P10	1	1	EV2

Table 4: The table shows the information about user evaluation participants including the number of years of experience, the reported number of videos they edited, and the assigned video for the study.

7 RESULTS

Below, we present the results to answer the research questions derived from the post-survey, interview, and observations taken during the study. The post-survey questions and respective results can be found in Appendix B.

7.1 RQ-1: Editing Patterns and Workflows with ExpressEdit

7.1.1 Usage Patterns. All participants started describing their multimodal command with NL and followed it up with the sketch only when they thought it was necessary, which accounted for 25.98% (STD=22.45%) of the commands. Many participants performed bulk editing, which applies multiple edits at once by referring to multiple moments within the video (P2, P3, P9, P6, P5), either by using general descriptions (e.g., "whenever she uses her hand gestures") or timestamps of the set of moments (e.g., "add an image of a lamp at the following frames: 9:04, 14:57, 15:22, 15:43, 18:07"). The participants also iterated on their commands by refining and adjusting

Video	Knowledge Characteristic	Content Format	Title	Reference Video
EV1	procedural	visual & verbal	Jamie Oliver live - pasta [76]	Learn To Cook In Less Than 1 Hour [29]
EV2	conceptual	verbal	How To SURVIVE As An Entrepreneur [84]	the mindset shift that will finally change your work-life [85]

Table 5: The table shows the information for the footage videos selected for the user evaluation including knowledge characteristics (procedural or conceptual), content formats (visual or verbal), and video links with respective reference video links.

them when needed (P3, P4, P5, P7, P8, P6, P10). The 'Examine' feature especially facilitated the understanding of how the system parsed the commands (P1, P2, P3, P4, P5, P9, P10) (M=6.1/7, STD=0.74) (Table 10) and gave rationale behind each suggested edit which helped to decide whether to accept or reject them (P6).

On average, the participants created 5.2 (STD=1.62) multimodal commands and made 9.3 (STD=2.54) requests which included iterations on the commands. In total, 16.6 (STD=6.7) edits are applied to different segments of the video from the commands. They accepted 45.98% (STD=24.8%) of the suggested edits, and 58.09% (STD=22.55%) of their final applied edits were adjusted versions of those edits. The summary of the ExpressEdit's usage is shown in Table 6.

7.1.2 Usage Workflows. There were notable usage patterns of ExpressEdit identified during the study. The participants first (1) *ideated* about what edits to implement, (2) *described* their edit intents, (3) *examined* the results returned from the system, and (4) *manually edited* the video.

The participants spent a roughly equal amount of time ideating and describing (M=13.9+17.4=31.3%), examining (M=33.2%), and manually editing (M=35.4%) with ExpressEdit. Interestingly, different behaviors emerged between the two videos, EV1 and EV2. The novices assigned to EV1 mostly examined (M=37.9%) the processed results of the commands while spending a relatively small amount of time manually editing (M=29.2%). On the other hand, the novice participants assigned to EV2 spent more time manually applying the edits (M=38.5%) as opposed to examining the processing results (M=27.0%). This can be explained by the characteristics of the videos. Since EV1 is more visually complex, users need more time to examine the suggested edits by playing and rewinding the part of the video when the edit is applied multiple times to ensure that it fits well. However, in EV2, users could relatively easily decide whether to accept or reject the edit by looking at the transcript or just seeing how the edit looks on the single frame, as the video was somewhat static for the most part. We summarize the aggregated average time spent by participants (for each assigned video) on the editing stages with the pie plot (Figure 4).

7.2 RQ-2: Understanding and Implementation of Edit Commands

7.2.1 Participants thought that ExpressEdit understood their commands well. Overall, the participants felt that the system understood their commands (P6, P3, P8, P4, P5, and P9), especially when the commands were detailed and specific (P6, P1, P2) or when they were based on the transcript or the visual content of the video (P4). At the same time, several participants acknowledged that the system had difficulty understanding broader or less detailed commands (P2, P5).

This aligns with the mixed responses we got from the post-survey regarding how well the system understood the commands (M=4.5/7, STD=1.1) (Table 8). P6, P1, P2, P9, and P10 thought that the misunderstanding might have happened due to their inadequate description of the command. Moreover, at the beginning of the task, a few participants were not sure about what kind of commands they could give (P7, P8) to the system, so they experimented with different commands to test how well the system understands. Fortunately, they said they were able to express their edits more effectively towards the end of the study as they got used to the system more.

7.2.2 Participants were satisfied with the ExpressEdit's implementation of their commands but adjusted the results to match the video better. The participants were generally satisfied with the implementation of the commands (P6, P2, P3, P5, P9, P10) and noted that they got better results when they gave *better* commands (P1, P7), although most of the system-suggested edits were modified or adjusted based on the participant's preference on how the edit fits the video (i.e., timing, duration, edit parameters). The post-survey results show that the ExpressEdit's implementation of the commands is indeed useful in terms of addressing the implementation of the commands (M=4.5/7, STD=1.4) (Table 8), satisfaction with the quality of the output (M=5/7, STD=1.5) and the performance (M=5.5/7, STD=1.1) (Table 8). P6 and P3 mentioned that they usually got more suggested edits than they were expecting, which was generally viewed as more favorable compared to not getting enough suggestions. Interestingly, P4 said that in certain cases, it was more efficient to manually create edits by using suggestions as markers of important moments in the video, especially for cutting out long segments of the video that cover multiple suggestions.

7.3 RQ-3: Usefulness of ExpressEdit

7.3.1 ExpressEdit improved the efficiency of editing videos. Consistent with our survey results, the participants felt that ExpressEdit enhanced their ability to communicate and implement edits (P1, P6, P3, P5, P9) (M=5.6/7, STD=1.17), and the system was easy to use for editing videos (P2, P3, P5, P8, P9) (M=5.6/7, STD=1.17) (Table 10). Almost all the participants found that they were able to build upon the edits suggested by the system (P1, P2, P4, P5, P6, P7, P8, P9). Specifically, it helped with locating the parts of the video where the edit should be applied (P8, P6, P9).

The participants also appreciated the sketching on top of the frame, which was mainly used to specify the location of the edit in the frame (P2, P4, P6, P7, P8, P9). P8 and P9 mentioned that it was easier to sketch than manually adjusting the position of the edit (e.g., textbox, image). P4 also acknowledged the convenience of sketching and styling with the command, which reduced the manual work when creating multiple edits.

Moreover, P6, P7, and P8 noted that describing the command and looking through the suggested edits inspired them and gave more ideas on what edits they could apply. This also aligns well with survey results that shows that the system helped participants think through edits (M=5.7/7, STD=1.42), feel in control of the system (M=6.1/7, STD=0.74), and collaborate with it (M=5.7/7, STD=1.34). The questions and the summary of scores are listed in Table 9

Despite its advantages, several participants noted that it might be difficult to do more complex and smooth edits (P2) with ExpressEdit and that it could miss some moments in the video where they wanted to apply an edit (P4).

7.3.2 Comparison with Previous Experiences. When asked about how the editing experience with ExpressEdit differed from previous experiences, many participants noted that the system was much simpler and easier to use. They also mentioned that the system focused on all the basic functionalities to edit an informational video (P1, P3, P5, P9), unlike Adobe Premiere Pro, which has sophisticated but complicated instruments that might confuse novices (P2, P8). However, the more experienced participants noted that ExpressEdit lacked some important editing functionalities to feel fully comfortable editing with the system (P1, P3, P4, P5, P9, P6, P8, P10). In terms of workflow, participants noted that they focused more on what edit to apply rather than sticking to the more common chronological order of editing (e.g., adding edits in chronological order of the video) (P4, P5, P9). Moreover, some participants appreciated that, unlike in other video editing tools, they were not starting from scratch but the system was making the initial edits (P1, P3, P5, P9). P8 felt more creative with the ExpressEdit's suggested edits, *"It made my editing process more creative."*

7.3.3 Overall feedback on ExpressEdit. The participants felt confident in their ability to use the system (M=5.5/7, STD=1.08), to plan and implement edits (M=5.4/7, STD=0.84), and to revisit and refine edits to improve the quality of the video (M=5.7/7, STD=1.25). However, they had mixed responses about editing different types of videos other than informational videos (M=4.5/7, STD=1.27), producing a quality video (M=4.5/7, STD=0.97), and translating the edits that they envisioned (M=4.7/7, STD=1.16). During the interviews, participants noted that the main reason was the limited editing functionality of ExpressEdit and that the quality videos should contain such as smooth animations, transitions, and engaging audio (P2, P9, P6, P1). The summary of the self-confidence scores is listed in Table 7. Overall, the system was easy to use, with the SUS usability score of 75.7 (STD=10.00). CSI results show that the system engaged the participants and allowed them to explore multiple options (Table 11). In terms of NASA-TLX, the participants felt moderate mental demand (M=4/7, STD=1.49) and put some effort into doing the tasks (M=3.8/7, STD=1.81), but relatively low workload for the other factors (Table 12).

8 DISCUSSION

In this paper, we present ExpressEdit, a multimodal system for editing informational videos. It is powered by CV and LLM-based pipeline that supports multimodal reasoning and real-time processing of natural language and sketching (NL&S) commands. We discuss the importance of balancing expressiveness and control, how ExpressEdit supports the creative process, how video characteristics affect the system usage, considerations for utilizing AI for video editing, and limitations and future work.

8.1 Balancing Expressiveness and Control

With ExpressEdit, we observed that the participants were able to edit videos efficiently and in a creative way when the system appropriately restricts the NL command usage to express the video editing requests to three defining aspects of a video edit; temporal location, spatial location, and edit operation and parameters. The participants gradually figured out what kind of commands the system supports and effectively used them in a relatively short duration of our study. Moreover, the three types of references facilitated the iteration of the edit commands and helped users detect where the system misinterpreted the command and fix it. Thus, we highlight the importance of balancing the sense of control that users have and the expressiveness that the system supports. While more advanced language models can allow video editing systems to support a broader set of NL expressions, the trade-off between expressiveness and control should be considered.

8.2 Supporting Different Phases of the Creative Process

The creative process consists of three iterative phases: ideation, execution, and evaluation [14]. ExpressEdit mainly supports the execution stage by allowing users to express and implement their edits more naturally and efficiently based on multimodal edit commands. However, our system can also help with the ideation and evaluation phases. For instance, users can describe the text content that they want on the video, and the system can suggest moments in the video and spatial location in a frame where such content could appear. It can help users judge if such an edit idea is appropriate and iterate on the idea at a low cost, which can then serve as a starting point for users to work on. Moreover, ExpressEdit can be useful at the evaluation stage of the creative process. It helps users evaluate if an edit would be appropriate or look *good* by automatically generating the edits suitable for each context (e.g., the appropriate spatial location and parameters). As such, ExpressEdit can support each of the creative process phases in video editing through multimodality.

8.3 Impact of Video Characteristics

From the user study, we observed that the participants exhibited varying editing behaviors when working with different videos, possibly due to the distinct characteristics of the videos — Participants with visually complex videos spent more time examining the processed results and less time manually editing, while the participants with verbally-oriented videos demonstrated the converse pattern of behavior (Section 7.1.2). It suggests that the workflow and design of the system can be improved in a way that reflects the video characteristics, such as displaying the processed results in an easy-to-skim manner with visually complex videos. Furthermore, types of edit operations that are effective or frequently used could be

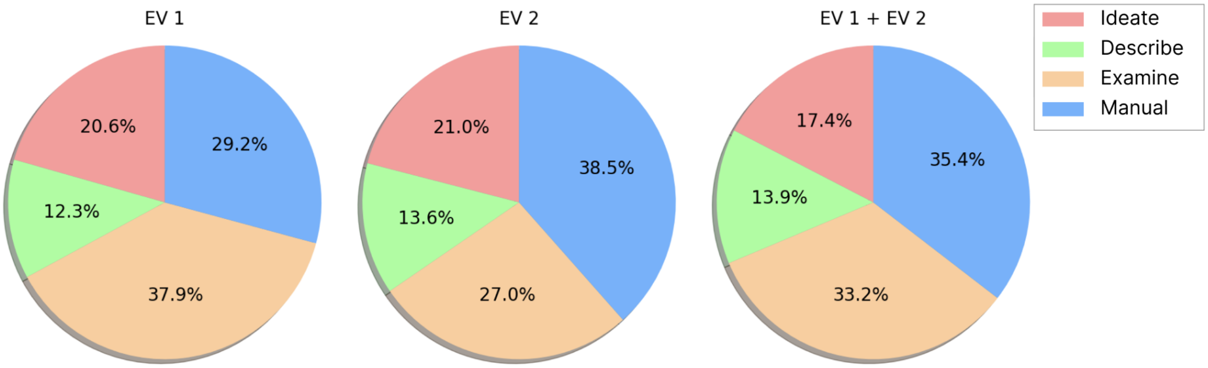


Figure 4: The pie plot shows the distribution of time spent in (1) *ideating* about what edits to implement, (2) *describing* their edit requests, (3) *examining* the suggested edits returned by the system, and (4) *manually editing* the video. The time spent for each editing process stage is derived from user interaction logs.

Participant	Requests #	Requests with Sketch %	Edit Commands #	Avg. Command Iterations #	Suggested Edits #	Accepted Suggestions %	Applied Edits #	Applied Suggestions %
P1	10	0.0%	5	1.00	38	39.5%	18	66.7%
P2	10	40.0%	6	0.67	30	56.7%	18	83.3%
P3	5	0.0%	5	0.00	12	83.3%	19	26.3%
P4	10	20.0%	4	1.50	40	15.0%	11	45.5%
P5	14	14.3%	8	0.75	60	28.3%	16	56.3%
P6	8	37.5%	3	1.67	19	31.6%	13	46.2%
P7	8	75.0%	5	0.60	22	72.7%	32	31.3%
P8	12	33.3%	7	0.71	55	29.1%	16	100.0%
P9	7	28.6%	6	0.17	14	78.6%	17	58.8%
P10	9	11.1%	3	2.00	16	25.0%	6	66.7%
Total	93	-	52	-	306	-	166	-
Avg. (STD)	9.3 (2.54)	26.0% (22.45)	5.2 (1.62)	0.91 (0.64)	30.6 (17.15)	45.98% (24.8)	16.6 (6.70)	58.1% (22.55)

Table 6: The table shows the qualitative summary of the work done by each participant in the user evaluation. The number of processing requests, the percentage of requests with a sketch, the number of individual edit commands, the average number of iterations on those commands, the number of suggested edits by the system for the session, the percentage of accepted suggested edits, the number of final applied edits, and the percentage of initially suggested edits among them.

different depending on the knowledge characteristic of the video content, such as procedural (e.g., cooking videos) and conceptual (e.g., mathematics lecture videos). Creating a system tailored to a particular video type can enhance the user experience by providing relevant editing suggestions.

8.4 Using AI for Video Editing

When developing AI-infused video editing tools for novices, it is crucial to acknowledge the ethical considerations such as inappropriate use of the system and its broader impact on novice users.

As existing work has shown [15, 50], AI models can generate inappropriate output that can have significant consequences when infused into interactive systems such as ours. For example, our CV and LLM-based pipeline can potentially generate edits with harmful or hallucinated content when editing educational videos. We follow common approaches to address these issues [6, 69] by providing a rationale behind each generated output of our pipeline and allowing users to manually revise the edits. Nevertheless, we emphasize the importance of addressing AI bias and ethics when developing AI-infused video editing tools.

As we observed in our study, ExpressEdit can effectively support novice video editors in creating video edits. While it is important to lower the barrier to engaging with video editing tools, it is equally important to ensure that the system allows novices to improve their skills and confidence in video editing. Thus, we acknowledge the potential of users' overreliance on AI-based interpretation of the edit commands and reduced self-confidence in video editing [11, 55]. We partly address this aspect by enabling manual video editing in ExpressEdit, however, additional features within the system that prompt the users to reflect on the generated edits [93] can promote better deliberate learning and understanding of video editing practice.

8.5 Limitations and Future Work

Although our study revealed the usefulness of ExpressEdit in video editing, there are a few limitations of our system and the study. In the formative study, we focused on edit commands that initiate edits rather than revise or adjust applied edits. We believe future work can build on top of our initial investigation . In ExpressEdit, the pipeline handles interpreting three reference types in an NL command. While these types appeared the most in our formative study, we also observed other uses of NL by participants, such as describing edit rationale and intended effect on the audience. Furthermore, the system supports sketching on top of the frame mainly to indicate the region of interest within the single frame. As it was observed in our study, there could be other cases where sketching is used, such as sketching the content that should be added or indicating the movement by sketching on multiple keyframes. Lastly, participants felt limited by the seven edit operations, which were chosen based on their frequency in the formative study. Thus, future work can investigate improving the pipeline to include a more diverse range of edit intents, and to extend the role of sketching and the set of supported edit operations beyond visual effects such as animation and audio manipulation.

We conducted an observational study with 10 novice video editors to evaluate how effective the pipeline is and how useful the system is. Future work can also evaluate ExpressEdit with a deployment study or in a more comparative setting to get more statistical insights into the benefits of the system. Moreover, the participants worked on pre-selected footage videos as opposed to their own videos. While this scenario is close to real-world tasks that video editors encounter, our participants had difficulties becoming familiar with the content of the videos and generating edit ideas within the duration of the study. To eliminate such challenges, ExpressEdit can be evaluated with the participants' videos, similar to the evaluation by Huh et al. [43].

9 CONCLUSION

We propose ExpressEdit, a multimodal video editing system that allows users to edit videos using natural language (NL) and sketching on top of a video frame. Based on findings from the formative study and the analysis of 176 multimodal edit commands, our technical pipeline powered by CV and large language models is designed in a way that it extracts and interprets (1) temporal, (2) spatial, and (3) operational references in an NL command and spatial references from sketching. Our system implements the interpreted edits which then users can iterate on. Our study with 10 participants shows that ExpressEdit helps users generate and express edit ideas and implement them effectively. We believe that our work opens up new opportunities in natural language video editing and multimodal interfaces.

ACKNOWLEDGMENTS

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-01347,Video Interaction Technologies Using Object-Oriented Video Modeling).

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 (Jan. 2022), 103915. https://doi.org/10.1016/j.robot.2021.103915

[2] Khan Academy. 2024. *Khan Academy.* https://www.khanacademy.org Accessed: 2024-01-25.

[3] Adobe. 2024. *Adobe Photoshop.* https://www.adobe.com/products/photoshop.html Accessed: 2024-01-25.

[4] Adobe. 2024. *Adobe Premiere Pro.* https://www.adobe.com/products/premiere.html Accessed: 2024-01-25.

[5] Remotion AG. 2024. *Remotion.* https://www.remotion.dev Accessed: 2024-01-25.

[6] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. https://www.microsoft.com/en-us/research/publication/guidelines-for-human-ai-interaction/

[7] Apple. 2024. *Final Cut Pro.* https://www.apple.com/final-cut-pro Accessed: 2024-01-25.

[8] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. 2022. Text2LIVE: Text-Driven Layered Image and Video Editing. https://doi.org/10.48550/arXiv.2204.02491

[9] Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2012. Tools for placing cuts and transitions in interview video. *ACM Transactions on Graphics* 31, 4 (July 2012), 67:1–67:8. https://doi.org/10.1145/2185520.2185563

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. https://arxiv.org/abs/2005.14165v4

[11] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z. Gajos. 2021. To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 188:1–188:21. https://doi.org/10.1145/3449287

[12] Mireille Bétrancourt and Kalliopi Benetos. 2018. Why and when does instructional video facilitate learning? A commentary to the special issue "developments and trends in learning with instructional video". *Computers in Human Behavior* 89 (Dec. 2018), 471–475. https://doi.org/10.1016/j.chb.2018.08.035

[13] Diogo Cabral and Nuno Correia. 2017. Video editing with pen-based technology. *Multimedia Tools and Applications* 76, 5 (March 2017), 6889–6914. https://doi.org/10.1007/s11042-016-3329-y

[14] Linda Candy. 2013. Evaluating Creativity. 57–84. https://doi.org/10.1007/978-1-4471-4111-2_4

[15] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. 2023. A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT. https://doi.org/10.48550/arXiv.2303.04226

[16] Juan Casares, A. Chris Long, Brad A. Myers, Rishi Bhatnagar, Scott M. Stevens, Laura Dabbish, Dan Yocum, and Albert Corbett. 2002. Simplifying video editing using metadata. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '02).* Association for Computing Machinery, New York, NY, USA, 157–166. https://doi.org/10.1145/778712.778737

[17] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. 2023. StableVideo: Text-driven Consistency-aware Diffusion Video Editing. https://doi.org/10.48550/arXiv.2308.09592

[18] Gael Chandler. 2004. *Cut by cut: editing your film or video.* Michael Wiese Productions, Studio City, CA.

[19] Minsuk Chang, Mina Huh, and Juho Kim. 2021. RubySlippers: Supporting Content-based Voice Navigation for How-to Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21).* Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445131

[20] Zhutian Chen, Shuainan Ye, Xiangtong Chu, Haijun Xia, Hui Zhang, Huamin Qu, and Yingcai Wu. 2022. Augmenting Sports Videos with VisCommentator. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (Jan. 2022), 824–834. https://doi.org/10.1109/TVCG.2021.3114806

[21] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Transactions on Computer-Human Interaction* 21, 4 (June 2014), 21:1–21:25. https://doi.org/10.1145/2617588

[22] Peggy Chi, Tao Dong, Christian Frueh, Brian Colonna, Vivek Kwatra, and Irfan Essa. 2022. Synthesis-Assisted Video Prototyping From a Document. In

Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22). Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3526113.3545676

[23] Peggy Chi, Nathan Frey, Katrina Panovich, and Irfan Essa. 2021. Automatic Instructional Video Creation from a Markdown-Formatted Tutorial. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 677–690. https://doi.org/10.1145/3472749.3474778

[24] Peggy Chi, Zheng Sun, Katrina Panovich, and Irfan Essa. 2020. Automatic Video Creation From a Web Page. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 279–292. https://doi.org/10.1145/3379337.3415814

[25] Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann. 2013. DemoCut: generating concise instructional videos for physical demonstrations. In *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13)*. Association for Computing Machinery, New York, NY, USA, 141–150. https://doi.org/10.1145/2501988.2502052

[26] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching Stories with Generative Pretrained Language Models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–19. https://doi.org/10.1145/3491102.3501819

[27] Codecademy. 2018. Livestream: Getting Started with C++ (Episode 1). Video. Retrieved on 2024-01-25 from https://www.youtube.com/live/OKQpOzEY_A4

[28] Descript. 2024. *Descript*. https://www.descript.com Accessed: 2024-01-25.

[29] Nick DiGiovanni. 2023. Learn To Cook In Less Than 1 Hour. Video. Retrieved on 2024-01-25 from https://youtu.be/zhI7bQyTmHw

[30] edX LLC. 2024. *edX*. https://www.edx.org Accessed: 2024-01-25.

[31] Logan Fiorella and Richard E. Mayer. 2018. What works and doesn't work with instructional video. *Computers in Human Behavior* 89 (Dec. 2018), 465–470. https://doi.org/10.1016/j.chb.2018.07.015

[32] Ohad Fried and Maneesh Agrawala. 2019. Puppet Dubbing. https://arxiv.org/abs/1902.04285v1

[33] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. 2019. Text-based editing of talking-head video. *ACM Transactions on Graphics* 38, 4 (July 2019), 68:1–68:14. https://doi.org/10.1145/3306346.3323028

[34] Ankit Gandhi, Arijit Biswas, Kundan Shrivastava, Ranjeet Kumar, Sahil Loomba, and Om Deshmukh. 2016. Easy Navigation through Instructional Videos using Automatically Generated Table of Content. In *Companion Publication of the 21st International Conference on Intelligent User Interfaces (IUI '16 Companion)*. Association for Computing Machinery, New York, NY, USA, 92–96. https://doi.org/10.1145/2876456.2879472

[35] Google. 2024. *Google Slides*. https://slides.google.com Accessed: 2024-01-25.

[36] Philip J. Guo, Juho Kim, and Rob Rubin. 2014. How video production affects student engagement: an empirical study of MOOC videos. In *Proceedings of the first ACM conference on Learning @ scale conference (L@S '14)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/2556325.2566239

[37] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, Peter A. Hancock and Najmedin Meshkati (Eds.). Human Mental Workload, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[38] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. 2022. Imagen Video: High Definition Video Generation with Diffusion Models. https://doi.org/10.48550/arXiv.2210.02303

[39] P. T. Hove. 2014. Characteristics of instructional videos for conceptual knowledge development. https://www.semanticscholar.org/paper/Characteristics-of-instructional-videos-for-Hove/c377da3ea8c08dbe79cd36927b25154ecb51cb48

[40] Xian-Sheng Hua, Zengzhi Wang, and Shipeng Li. 2005. LazyCut: content-aware template-based video authoring. In *Proceedings of the 13th annual ACM international conference on Multimedia (MULTIMEDIA '05)*. Association for Computing Machinery, New York, NY, USA, 792–793. https://doi.org/10.1145/1101149.1101318

[41] Nisha Huang, Yuxin Zhang, and Weiming Dong. 2023. Style-A-Video: Agile Diffusion for Arbitrary Text-based Video Style Transfer. https://doi.org/10.48550/arXiv.2305.05464

[42] Bernd Huber, Hijung Valentina Shin, Bryan Russell, Oliver Wang, and Gautham J. Mysore. 2019. B-Script: Transcript-based B-roll Video Editing with Recommendations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300311

[43] Mina Huh, Saelyne Yang, Yi-Hao Peng, Xiang 'Anthony' Chen, Young-Ho Kim, and Amy Pavel. 2023. AVscript: Accessible Video Editing with Audio-Visual Scripts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3544548.3581494

[44] Imvidu. 2024. *Imvidu*. https://imvidu.com Accessed: 2024-01-25.

[45] Apple Inc. 2024. *iMovie*. https://www.apple.com/ca/imovie Accessed: 2024-01-25.

[46] Coursera Inc. 2024. *Coursera*. https://www.coursera.org Accessed: 2024-01-25.

[47] Upwork Global Inc. 2024. *Upwork*. https://www.upwork.com/ Accessed: 2024-01-25.

[48] Zoom Video Communications Inc. 2024. *Zoom*. https://zoom.us Accessed: 2024-01-25.

[49] Amir Jahanlou and Parmit K Chilana. 2022. Katika: An End-to-End System for Authoring Amateur Explainer Motion Graphics Videos. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3491102.3517741

[50] Maurice Jakesch, Advait Bhat, Daniel Buschek, Lior Zalmanson, and Mor Naaman. 2023. Co-Writing with Opinionated Language Models Affects Users' Views. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3544548.3581196

[51] Tero Jokela, Kaj Mäkelä, and Minna Karukka. 2007. Empirical observations on video editing in the mobile context. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology (Mobility '07)*. Association for Computing Machinery, New York, NY, USA, 482–489. https://doi.org/10.1145/1378063.1378140

[52] Murat Kalender, M. Tolga Eren, Zonghuan Wu, Ozgun Cirakman, Sezer Kutluk, Gunay Gultekin, and Emin Erkan Korkmaz. 2018. Videolization: knowledge graph based automated video generation from web content. *Multimedia Tools and Applications* 77, 1 (Jan. 2018), 567–595. https://doi.org/10.1007/s11042-016-4275-4

[53] Kimia Kiani, Parmit K. Chilana, Andrea Bunt, Tovi Grossman, and George Fitzmaurice. 2020. "I Would Just Ask Someone": Learning Feature-Rich Design Software in the Modern Workplace. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 1–10. https://doi.org/10.1109/VL/HCC50065.2020.9127288 ISSN: 1943-6106.

[54] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 4017–4026. https://doi.org/10.1145/2556288.2556986

[55] Tae Soo Kim, DaEun Choi, Yoonseo Choi, and Juho Kim. 2022. Stylette: Styling the Web with Natural Language. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3491102.3501931

[56] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. https://doi.org/10.48550/arXiv.2304.02643

[57] KonvaJS. 2024. *KonvaJS*. https://konvajs.org Accessed: 2024-01-25.

[58] LangChain. 2024. *LangChain*. https://www.langchain.com Accessed: 2024-01-25.

[59] 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 (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2185–2194. https://doi.org/10.1145/2470654.2481301

[60] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Transactions on Graphics* 36, 4 (July 2017), 130:1–130:14. https://doi.org/10.1145/3072959.3073653

[61] Mackenzie Leake, Hijung Valentina Shin, Joy O. Kim, and Maneesh Agrawala. 2020. Generating Audio-Visual Slideshows from Text Articles Using Word Concreteness. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376519

[62] 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 (AVI '18)*. Association for Computing Machinery, New York, NY, USA, 1–3. https://doi.org/10.1145/3206505.3206602

[63] Seung Hyun Lee, Sieun Kim, Innfarn Yoo, Feng Yang, Donghyeon Cho, Youngseo Kim, Huiwen Chang, Jinkyu Kim, and Sangpil Kim. 2023. Soundini: Sound-Guided Diffusion for Natural Video Editing. https://doi.org/10.48550/arXiv.2304.06818

[64] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. https://doi.org/10.48550/arXiv.2301.12597 arXiv:2301.12597

[cs].

[65] Jian Liao, Adnan Karim, Shivesh Singh Jadon, Rubaiat Habib Kazi, and Ryo Suzuki. 2022. RealityTalk: Real-Time Speech-Driven Augmented Presentation for AR Live Storytelling. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3526113.3545702

[66] David Chuan-En Lin, Fabian Caba Heilbron, Joon-Young Lee, Oliver Wang, and Nikolas Martelaro. 2022. VideoMap: Video Editing in Latent Space. https://arxiv.org/abs/2211.12492v1

[67] Steve Kaufmann lingosteve. 2019. Language Learning Live Stream. Video. Retrieved on 2024-01-25 from https://www.youtube.com/live/3_nLdcHBJY4

[68] Doctor Gary Linkov. 2022. Surgeon does Live QA | Hair Loss Awareness Month. Video. Retrieved on 2024-01-25 from https://www.youtube.com/live/sz8Lo3NY1m0

[69] Zachary C. Lipton. 2017. The Mythos of Model Interpretability. https://doi.org/10.48550/arXiv.1606.03490

[70] Google LLC. 2024. *YouTube*. https://www.youtube.com Accessed: 2024-01-25.

[71] Blackmagic Design Pty. Ltd. 2024. *DaVinci Resolve*. https://www.blackmagicdesign.com/products/davinciresolve Accessed: 2024-01-25.

[72] Cui-Xia Ma, Yong-Jin Liu, Hong-An Wang, Dong-Xing Teng, and Guo-Zhong Dai. 2012. Sketch-Based Annotation and Visualization in Video Authoring. *IEEE Transactions on Multimedia* 14, 4 (Aug. 2012), 1153–1165. https://doi.org/10.1109/TMM.2012.2190389

[73] Jiaju Ma, Anyi Rao, Li-Yi Wei, Rubaiat Habib Kazi, Hijung Valentina Shin, and Maneesh Agrawala. 2023. Automated Conversion of Music Videos into Lyric Videos. https://doi.org/10.1145/3586183.3606757

[74] Meta. 2024. *React*. https://react.dev Accessed: 2024-01-25.

[75] MobX. 2024. *MobX*. https://mobx.js.org Accessed: 2024-01-25.

[76] Jamie Oliver. 2011. Jamie Oliver live - pasta. Video. Retrieved on 2024-01-25 from https://youtu.be/b3TVLNNqgdc

[77] OpenAI. 2023. GPT-4 Technical Report. https://doi.org/10.48550/arXiv.2303.08774

[78] 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 (ICMI '04)*. Association for Computing Machinery, New York, NY, USA, 129–136. https://doi.org/10.1145/1027933.1027957

[79] Pallets. 2024. *Flask*. https://flask.palletsprojects.com Accessed: 2024-01-25.

[80] Lihang Pan, Chun Yu, Zhe He, and Yuanchun Shi. 2023. A Human-Computer Collaborative Editing Tool for Conceptual Diagrams. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–29. https://doi.org/10.1145/3544548.3580676

[81] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. SceneSkim: Searching and Browsing Movies Using Synchronized Captions, Scripts and Plot Summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/2807442.2807502

[82] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2016. VidCrit: Video-based Asynchronous Video Review. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. Association for Computing Machinery, New York, NY, USA, 517–528. https://doi.org/10.1145/2984511.2984552

[83] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video digests: a browsable, skimmable format for informational lecture videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14)*. Association for Computing Machinery, New York, NY, USA, 573–582. https://doi.org/10.1145/2642918.2647400

[84] Gillian Perkins. 2020. How To SURVIVE As An Entrepreneur. Video. Retrieved on 2024-01-25 from https://www.youtube.com/live/oYMAX90kNkU

[85] Gillian Perkins. 2023. the mindset shift that will finally change your work-life. Video. Retrieved on 2024-01-25 from https://youtu.be/T8LE3SpZdag

[86] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. 2023. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. https://doi.org/10.48550/arXiv.2303.09535

[87] Bosheng Qin, Juncheng Li, Siliang Tang, Tat-Seng Chua, and Yueting Zhuang. 2023. InstructVid2Vid: Controllable Video Editing with Natural Language Instructions. https://doi.org/10.48550/arXiv.2305.12328

[88] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. https://doi.org/10.48550/arXiv.2103.00020

[89] Gordon Ramsey. 2021. At Home for the Holidays with Gordon Ramsay. Video. Retrieved on 2024-01-25 from https://www.youtube.com/live/kdN41iYTg3U

[90] Nils Reimers. 2024. *SentenceTransformers*. https://www.sbert.net Accessed: 2024-01-25.

[91] Nazmus Saquib, Rubaiat Habib Kazi, Li-Yi Wei, and Wilmot Li. 2019. Interactive Body-Driven Graphics for Augmented Video Performance. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300852

[92] Jeff Sauro. 2011. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. CreateSpace Independent Publishing Platform. Open Library ID: OL26858541M.

[93] Hyungyu Shin, Eun-Young Ko, Joseph Jay Williams, and Juho Kim. 2018. Understanding the Effect of In-Video Prompting on Learners and Instructors. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173893

[94] 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 (Dec. 2021), 56–66. https://doi.org/10.1016/j.visinf.2021.12.005

[95] Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. 2017. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics* 36, 4 (July 2017), 93:1–93:11. https://doi.org/10.1145/3072959.3073699

[96] Linus Tech Tips. 2018. Microsoft Surface Go - Classic LIVE Unboxing. Video. Retrieved on 2024-01-25 from https://www.youtube.com/live/4LdIvyfzoGY

[97] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. QuickCut: An Interactive Tool for Editing Narrated Video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. Association for Computing Machinery, New York, NY, USA, 497–507. https://doi.org/10.1145/2984511.2984569

[98] Prateksha Udhayanan, Suryateja Bv, Parth Laturia, Dev Chauhan, Darshan Khandelwal, Stefano Petrangeli, and Balaji Vasan Srinivasan. 2023. Recipe2Video: Synthesizing Personalized Videos from Recipe Texts. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2267–2276. https://doi.org/10.1109/WACV56688.2023.00230 ISSN: 2642-9381.

[99] Jan Van Der Kamp and Veronica Sundstedt. 2011. Gaze and voice controlled drawing. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*. ACM, Karlskrona Sweden, 1–8. https://doi.org/10.1145/1983302.1983311

[100] Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, and Ariel Shamir. 2019. Write-a-video: computational video montage from themed text. *ACM Transactions on Graphics* 38, 6 (Nov. 2019), 177:1–177:13. https://doi.org/10.1145/3355089.3356520

[101] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. 2022. InternVideo: General Video Foundation Models via Generative and Discriminative Learning. https://arxiv.org/abs/2212.03191v2

[102] Tongshuang Wu, Michael Terry, and Carrie J. Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. https://doi.org/10.48550/arXiv.2110.01691

[103] Haijun Xia, Jennifer Jacobs, and Maneesh Agrawala. 2020. Crosscast: Adding Visuals to Audio Travel Podcasts. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 735–746. https://doi.org/10.1145/3379337.3415882

[104] Saelyne Yang, Jisu Yim, Juho Kim, and Hijung Valentina Shin. 2022. CatchLive: Real-time Summarization of Live Streams with Stream Content and Interaction Data. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–20. https://doi.org/10.1145/3491102.3517461

[105] Xinwei Yao, Ohad Fried, Kayvon Fatahalian, and Maneesh Agrawala. 2020. Iterative Text-based Editing of Talking-heads Using Neural Retargeting. https://doi.org/10.48550/arXiv.2011.10688

[106] youtube-dl developers. 2024. *youtube-dl*. https://ytdl-org.github.io/youtube-dl Accessed: 2024-01-25.

[107] Yaxi Zhao, Razan Jaber, Donald McMillan, and Cosmin Munteanu. 2022. "Rewind to the Jiggling Meat Part": Understanding Voice Control of Instructional Videos in Everyday Tasks. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3491102.3502036

[108] Mingyuan Zhong, Gang Li, Peggy Chi, and Yang Li. 2021. HelpViz: Automatic Generation of Contextual Visual Mobile Tutorials from Text-Based Instructions. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 1144–1153. https://doi.org/10.1145/3472749.3474812

[109] 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 (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3491102.3502062

A GROUND TRUTH CONSTRUCTION FOR THE TECHNICAL PIPELINE

When constructing the ground truth dataset to evaluate the technical pipeline, the following procedure was used to analyze each of the 50 edit commands:

(1) if the edit command contained a screenshot of the frame with the timestamp and a clear indication of where the edit should be within the frame, then we used that information directly within the edit command's spatial and temporal parameters.
(2) if the edit command did not contain a screenshot of the frame, then we analyzed the natural language part of the command. For references that mentioned explicit timestamps (e.g., "at 15:57,...") or absolute positions within the video frame (e.g., "top left corner"), we used that information directly within the edit command's spatial and temporal parameters.
(3) For NL commands with more conditional language (e.g., "whenever the camera is facing down to the pan"), we manually identified parts of the video that satisfied these conditions. For edit commands with vague or non-existent spatial references, we placed the edits in the frame such that it did not block the important content in the video (e.g., speakers, objects that they are interacting with).

B USER EVALUATION RESULTS

Criteria	Avg. (STD)
1. I feel confident in my ability to navigate and use the system.	5.5 (1.08)
2. I am confident in my skills in editing various types of videos (visual/verbal) using this system.	4.5 (1.27)
3. I am confident that I can produce quality videos using this system.	4.5 (0.97)
4. I am confident that I can plan edits with the system and implement them.	5.4 (0.84)
5. I am confident that I can translate what I envision into actual video edits with this system.	4.7 (1.16)
6. I am confident that I can revisit/refine my video edits to achieve better results with this system.	5.7 (1.25)

Table 7: The table shows the survey results for self-confidence ratings of the participants from the user evaluation in terms of a 7-point Likert-scale.

Criteria	Avg. (STD)
1. The system understood my edit commands (in natural language and sketched form) well.	4.5 (1.08)
2. The system implemented my edit commands (in natural language and sketched form) well.	4.5 (1.43)
3. I am satisfied with the outcome quality of the edit command (natural language request and sketching) processing.	5 (1.49)
4. I am satisfied with the performance of the edit command (natural language request and sketching) processing.	5.5 (1.08)

Table 8: The table shows the survey results for questions about the perceived performance of the ExpressEdit's processing in terms of a 7-point Likert-scale.

Criteria	Avg. (STD)
1. I am satisfied with my final results from the system; they met the task goal.	3.9 (1.66)
2. The system helped me think through what kinds of output I would want to complete the task goal, and how to complete the task.	5.7 (1.42)
3. The system is transparent about how it arrives at its final result; I could roughly track its progress.	5.6 (1.35)
4. I felt I had control creating with the system. I can steer the system towards the task goal.	6.1 (0.74)
5. In the system, I felt I was collaborating with the system to come up with the outputs.	5.7 (1.34)

Table 9: The table shows the survey results for self-perceived experience with AI tool [102] in terms of a 7-point Likert-scale.

Criteria	Avg. (STD)
1. It was easy to understand and use the Edit description (natural language request and sketching) features in the system.	5.6 (1.17)
2. The Edit description (natural language request and sketching) features enhanced my ability to communicate and implement video edits.	5.6 (1.17)
3. The Examine (breakdown of the natural language request) feature helped me understand how the Edit description feature (natural language request and sketching) works.	6.3 (0.48)
4. The Examine (breakdown of the natural language description) feature helped me iterate (refine/adjust) the Edit descriptions (natural language request and sketching) I made.	6.1 (0.74)

Table 10: The table shows the survey results for questions about the usefulness of the "Edit Description" and "Examine" features of ExpressEdit in terms of a 7-point Likert-scale.

Criteria	Avg. (STD)
Enjoyment	6.1 (1.02)
Exploration	6.1 (1.05)
Expressiveness	4.8 (1.36)
Immersion	4.6 (1.67)
Results Worth Effort	5.0 (1.59)

Table 11: The table shows the Creativity Support Index scores for Enjoyment, Exploration, Expressiveness, Immersion, and Results Worth Effort in terms of a 7-point Likert-scale.

Video	Mental	Physical	Temporal	Effort	Performance	Frustration
EV1	4 (1.58)	2.4 (1.14)	4.4 (1.52)	3.8 (2.17)	2.4 (0.55)	1.6 (0.89)
EV2	4 (1.58)	1.4 (0.55)	3.4 (2.89)	3.8 (1.64)	4.4 (1.52)	1.8 (1.79)
Total	4 (1.49)	1.9 (0.99)	3.9 (2.23)	3.8 (1.81)	3.4 (1.51)	1.7 (1.34)

Table 12: The table shows the reported NASA-TLX scores (i.e., average (std)) from the user evaluation in terms of a 7-point Likert-scale.

C PROMPTS USED AS PART OF THE TECHNICAL PIPELINE

C.1 Parsing Edit Command (Stage 1 in Figure 3)

```
You are a video editor's assistant who is trying to understand the natural language command in the
    context of a given video. You will do it step-by-step.

Step 1: Identify the list of edit operations that the command is referring to:
- choose only among "text", "image", "shape", "blur", "cut", "crop", "zoom"
- make sure that the edit operation is only one of the above
- if none of the above edit operations is directly relevant, give the one that is most relevant to the
    command (e.g. "highlight" -> "shape" with type parameter "star")

Step 2: You have to identify 3 types of references from the command (Note: if there is a reference that
    contains noun-references such as this, that, it, etc. you will have to identify the noun that it
    refers to and replace the noun-reference with the noun.):
1. Temporal reference: any information in the command that could refer to a segment of the video:
- explicit timecodes or time ranges
- explicit mentions or implicit references to the transcript of the video
- description of the actions that happen in the video
- visual description of objects, moments, and frames in the video

2. Spatial reference: any information in the command that could refer to location or region in the video
    frame:
- specific locations or positions relative to the frame
- specific objects or areas of interest

3. Edit Parameter reference: any information in the command that could refer to specific parameters of
    an edit operation that was identified ([text, image, shape, blur, cut, crop, zoom]).
- text: content, font style, font color, or font size
- image: visual keywords
- shape: type of shape
- blur: degree of blur to apply
- cut: no parameters
- crop: how much to crop
- zoom: how long to perform the zooming animation

Step 3-1: You will classify each temporal reference into one of the following:
1. "position": reference in the form of a timecode (e.g. "54:43", "0:23"), time segment (e.g.
    "0:00-12:30", "from 43:30 to 44:20") or more abstract temporal position (e.g. "intro", "ending",
    "beginning part of the video")
2. "transcript": reference to transcript both implicit or explicit
3. "video": reference to specific action in the video or visual description of the frame, object, or
    elements
4. "other": reference to other temporal information that does not fall into the above categories

Step 3-2: You will classify each spatial reference into one of the following:
1. "visual-dependent": reference to specific objects, elements, or regions in the video frame that
    depend on the visual content of the video
2. "independent": reference to specific locations or positions relative to the frame independent of the
    visual content of the video
3. "other": any other spatial information that does not fall into the above categories

Step 4: Format the output based on the result of each step.

{ few-shot examples... }
```

Prompt 1: The prompt for Stage 1.

C.2 Temporal Interpretation (Stage 2 in Figure 3)

```
You are a video editor's assistant who is trying to understand natural language temporal reference in
    the video. You will do it step-by-step.

First step: Identify the type of temporal reference based on the user's command.
1. Timecode: a specific time in the video
2. Time range: a range of time in the video
3. More high level temporal reference: a reference to a generic event in the video (introduction,
    ending, etc.)

Second step: Identify the timecode or time range with additional context.
Note 1: If the temporal reference is just a timecode, output any 10 second interval containing the
    timecode.
Note 2: If there are more than one segment of video that matches the temporal reference, output all of
    them in a list.

{ few-shot examples... }
```

Prompt 2: The prompt for Stage 2.

```
You are a video editor's assistant who is trying to understand the natural language reference of the
    video editor to some part of the video given the original context of the reference and relevant
    snippets of the transcript of the video.

Instruction:
Locate the snippets of the transcript that are relevant to the editor's command and original context,
    and return the positions of those snippets from the list along with short explanation of how each
    one is relevant to editor's command and original context.

Note 1: If there are no relevant snippets, return an empty array [].
Note 2: If there is more than one snippet that is relevant to the editor's command, output all of them
    in a list with respective indexes and explanations.

{ few-shot examples... }
```

Prompt 3: The prompt for Stage 2 transcript references.

```
You are a video editor's assistant who is trying to understand the natural language reference of the
    video editor to some part of the video given the set of most relevant visual descriptions of
    10-second clips of the video and original context of the command. Visual description of a 10-second
    clip consists of an action label which is a main action happening, an abstract caption which is an
    abstract description of the clip, and the dense captions, which are list of descriptions of objects
    that are present. Try taking into account each of them.

Instruction:
Locate the visual descriptions that are relevant to the editor's command and original context of the
    command, and return the positions of those descriptions from the list along with short explanation
    of how each is relevant to editor's command.

Note 1: If there are no relevant viusal descriptions, return an empty array [].
Note 2: If there is more than one description that is relevant to the editor's command and original
    context, output all of them in a list.

{ few-shot examples... }
```

Prompt 4: The prompt for Stage 2 video references.

C.3 Spatial Interpretation (Stage 3 in Figure 3)

```
You are a video editor's assistant who is trying to understand editor's natural language description of
    the spatial location within the frame. The description is based on the rectangle that is already
    present in the frame. You will have to refine its location and resize (if necessary) based on the
    command.
You will be given the initial location of the rectangle in the frame: x, y, width, height, where (x, y)
    are coordinates of the top-left corner, and (width, height) are just width and height. Also, you
    will be given a command that describes the desired spatial location of the rectangle in the frame,
    the original context of the command, and the boundaries of the frame (e.g. width=1280, height=720)

You will do it step-by-step.
1. Refine the location of the rectangle (x, y coorindates) based on the command, original context of the
    command, and boundaries of the frame (make sure not to exceed the boundaries);
2. Resize the rectangle (width, height) based on the command, original context of the command, and
    boundaries of the frame (make sure not to exceed the boundaries);

Perform each step one-by-one and output the final location of the rectangle in the frame in appropriate
    format.

{ few-shot examples... }
```

Prompt 5: The prompt for Stage 3.

C.4 Edit Operation and Parameters Interpretation (Stage 4 in Figure 3)

```
You are a video editor's assistant who is trying to understand video edit parameter change requests in
    natural language. You are given a natural language command from the editor, the original context of
    the command, and initial values of the video edit parameters. You have to appropriately change the
    parameters to satisfy the command within its original context. You will do it step-by-step.

Step 1: Identify the type of each edit parameter change based on the user's command. There are three
    types of video edit parameter change requests:
1. Explicit: explicit values for a parameter (e.g. 12px, 10%, "Introduction", etc.)
2. Relative: a relative change to a parameter (e.g. 5 seconds longer, 10% less, fewer words, etc.)
3. Abstract: an abstract change to a parameter (e.g. shorter, longer, more, less, etc.)

Step 2: Transform each type of parameter change request into parameter values based on the "Initial
    parameters" provided and output the adjusted set of video edit parameters.

{ few-shot examples... }
```

Prompt 6: The prompt for Stage 4.

```
You are a video editor's assistant who is trying to understand natural language request of the editor to
    come up with search query for images to put in the video. You are given a command from the editor,
    the original context of the command, and relevant content from the video. Relevant content is a list
    of snippets from the transcript and visual description (what action is happening, abstract caption,
    and descriptions of objects) of 10-second segments. You must generate the search query for the image
    to be displayed based on the editor's command, original context, and relevant content.

Note 1: If no relevant search query can be generated that satisfies the command, output only the command.
Note 2: Make sure that the search query is not too long, since it should be seen by the editor. Keep it
    under 100 characters.
```

```
{ few-shot examples... }
```

Prompt 7: The prompt for Stage 4 image search query.

```
You are a video editor's assistant who is trying to understand natural language request of the editor to
    find a text to display in the video. You are given a command from the editor, the original context
    of the command, and relevant content from the video. Relevant content is a list of snippets from the
    transcript and visual description (what action is happening, abstract caption, and descriptions of
    objects) of 10-second segments. You must generate the text to be displayed based on the editor's
    command, original context, and relevant content.

Note 1: If no relevant text can be generated that satisfies the command, output the input command itself
    with reasonable formatting.
Note 2: Make sure that text is not too long, since it will be displayed on the screen. Keep it under 100
    characters.

{ few-shot examples... }
```

Prompt 8: The prompt for Stage 4 text insertion parameter interpretation.