ExpressEdit: Video Editing with Natural Language and Sketching

Bekzat Tilekbay[1], Saelyne Yang[1], Michal Lewkowicz[2], Alex Suryapranata[1] and Juho Kim[1]

[1]*School of Computing, KAIST, Daejeon, Republic of Korea*
[2]*Yale University, New Haven, Connecticut, USA*

Abstract
Informational videos serve as a crucial source of conceptual and procedural knowledge for many people. While it is important to make informational videos instructive and engaging, editing such videos (e.g., trimming, overlaying text/image, etc.) can be difficult and time-consuming. Especially for novice video editors, who often struggle with expressing and executing their editing ideas. We present ExpressEdit, a system that facilitates editing informational videos via natural language text and sketching directly on the video frame by interpreting multimodal editing commands and suggesting applicable edits. Powered by a multimodal technical pipeline, the system interprets (1) temporal, (2) spatial, and (3) operational references in an NL editing command and spatial references from sketching. This work offers insights into building multimodal interfaces for video editing.

Keywords
video editing, human-AI interaction, multimodal input, LLM

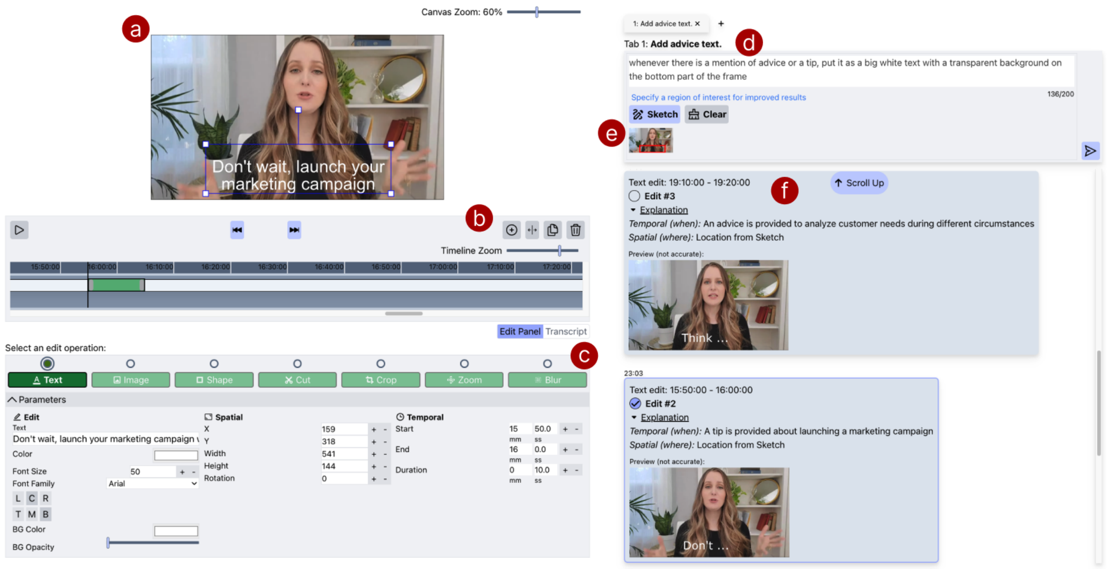
1. Introduction

Informational videos are videos that introduce, explain, or demonstrate conceptual or procedural knowledge [1, 2, 3]. 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.) [4, 5, 6, 7]. They have become a popular source of knowledge for many people due to their rich and engaging content [5].

However, editing informational videos is a tedious task that involves carefully organizing the footage, removing unnecessary parts, and finding and incorporating additional media assets [4]. While popular commercial tools for video editing offer all 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 [8, 9].

We investigate how multimodality – natural language (NL) and sketching – can be leveraged in the informational video editing scenario. We conducted a formative study with 10 video editors with diverse levels of expertise and collected 176 expressions of video editing requests in

Joint Proceedings of the ACM IUI Workshops 2024, March 18-21, 2024, Greenville, South Carolina, USA
tilekbay@kaist.ac.kr (B. Tilekbay); saelyne@kaist.ac.kr (S. Yang); michal.lewkowicz@yale.edu (M. Lewkowicz); alextio@kaist.ac.kr (A. Suryapranata); juhokim@kaist.ac.kr (J. Kim)
0009-0003-0002-5400 (B. Tilekbay); 0000-0003-1776-4712 (S. Yang); 0009-0008-3522-2259 (M. Lewkowicz); 0009-0008-9494-2256 (A. Suryapranata); 0000-0001-6348-4127 (J. Kim)



Don't wait, launch your marketing campaign

Select an edit operation: [✓ Text] Image Shape Cut Crop Zoom Blur

Edit

Text	Don't wait, launch your marketing campaign \
Color	
Font Size	50
Font Family	Arial
	L [✓ C] R
	T M [✓ B]
BG Color	
BG Opacity	

Spatial

X	159
Y	318
Width	541
Height	144
Rotation	0

Temporal

	mm	ss
Start	15	50.0
End	16	0.0
Duration	0	10.0

Tab 1: **Add advice text.**

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

Specify a region of interest for improved results

Text edit: 19:10:00 - 19:20:00
Edit #3
Explanation
Temporal (when): An advice is provided to analyze customer needs during different circumstances
Spatial (where): Location from Sketch
Preview (not accurate):

Text edit: 15:50:00 - 16:00:00
Edit #2
Explanation
Temporal (when): A tip is provided about launching a marketing campaign
Spatial (where): Location from Sketch
Preview (not accurate):

Figure 1: ExpressEdit is a multimodal system that enables users to edit videos using natural language and sketch. (a) Users can see the imported video and (b) use the timeline to navigate the video. (d) Users can describe an edit command using natural language in the *edit description box* and specify where in the video frame to apply the edit using the (e) *sketchpad*. ExpressEdit processes the edit command and (f) returns the result, showing the preview of the edit. (c) Users can manually revise resulting edits by adjusting the edit operation and its parameters.

the form of NL texts, sketches, and media assets. Focusing on edit expressions that initiate edits rather than revise or adjust applied edits, we found that editors feel comfortable expressing their general editing requests through NL text and use sketching on top of the frame to indicate specific locations or regions of interest.

Based on the findings from the formative study, we built 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 model-based technical pipeline that comprehends and executes the edits by extracting and interpreting three types of references from the multimodal command (Figure 1d,e): (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., *"put a text with the mentioned specifications of the laptop"*). ExpressEdit also provides the breakdown of the command into aforementioned types of references, as well as gives reasoning for each generated edit. From a user study with three novice video editors, we found ExpressEdit makes video editing more effective and allows the participants to try out several edit ideas efficiently.

2. Formative Study

To learn about (1) the role of natural language (NL) text and sketching in describing video editing requests and (2) their use cases in editing informational videos, we conducted a formative study with 10 video editors, where they were asked to express their edit requests that would improve the informativeness and engagement of a given video. We call these expressions *video editing commands*. We focused on editing expressions that initiate edits rather than revise or adjust applied edits. We believe future work can build on top of the initial investigation.

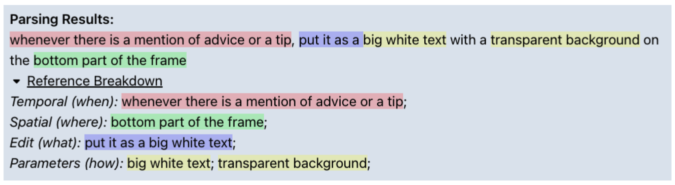
We recruited five novices who had edited at least 2 videos and five experienced editors who had edited at least 20 videos and 5 informational videos to cover a diverse range of editing commands since edit expressions such as attention to detail and vocabulary used can vary depending on the participant's editing expertise. We chose five archived informational live streams as raw footage for the study 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. To allow participants to describe their edit commands in both text and sketch, we used Google Slides[1], 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. After the main task, we conducted a short semi-structured interview to learn about the participant's experience performing the given task.

As a result, the collected 176 multimodal editing commands and the interview results revealed several patterns of multimodal video editing request expressions and the role of each modality provided within the tool. We found that the participants comfortably **described video editing intents using various modalities**: NL text, sketch, image, and graphics. In almost all the commands, the participants **referenced moments (e.g., timestamps, visual content, verbal content) in the video where edit should be applied using NL text**. Furthermore, they **used both NL text and sketching on top of the frame to reference the spatial location of the edits within the frame**. The participants **used NL text to describe edit operations and their parameters**. They mentioned the exact names of the operation (e.g., cut, text, image, etc.) or mentioned the main purpose or intended effect of the edit (e.g., highlight, emphasize, focus). For parameters of the edit operations, they gave either precise numbers or descriptions (e.g., large text, slow zoom). Lastly, participants frequently iterated on their commands to make them clearer by revisiting and redefining their expressions, for example, to make them more precise or to keep the consistency between several editing commands.

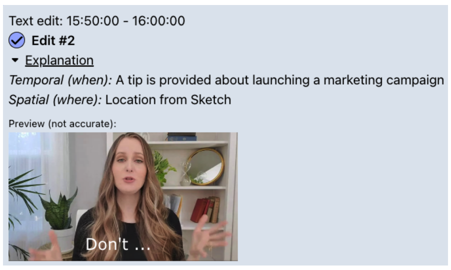
3. ExpressEdit: Interface

Based on formative study findings, we designed ExpressEdit, a multimodal video editing tool for editing informational videos. Our system interprets the user's editing command in the form of NL&S and suggests a set of edits with temporal location (where in the video timeline), spatial location (where in the video frame), and edit operation & parameters (which edit and how). To better understand the generated edits and iterate on the NL&S command, users can examine the breakdown of the parsing results as well as the reasoning for the temporal & spatial location of each generated edit. Additionally, users can manually adjust the generated edits by the system or create their own edits.

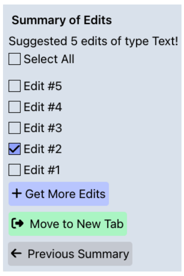
[1]Google Slides


Parsing Results:
whenever there is a mention of advice or a tip, put it as a big white text with a transparent background on the bottom part of the frame
▾ Reference Breakdown
Temporal (when): whenever there is a mention of advice or a tip;
Spatial (where): bottom part of the frame;
Edit (what): put it as a big white text;
Parameters (how): big white text; transparent background;

(a) The *examine* panel


Text edit: 15:50:00 - 16:00:00
✓ **Edit #2**
▾ Explanation
Temporal (when): A tip is provided about launching a marketing campaign
Spatial (where): Location from Sketch
Preview (not accurate):

(b) The *edit result* panel


Summary of Edits
Suggested 5 edits of type Text!
☐ Select All
☐ Edit #5
☐ Edit #4
☐ Edit #3
☑ Edit #2
☐ Edit #1

(c) The *summary* panel

Figure 2: ExpressEdit returns three types of responses after processing the user's edit command: (a) The *examine* panel analyzes the user's natural language command and shows which parts of the input correspond to the description of temporal location, spatial location, and edit operation type and parameters. (b) The *edit result* panel shows the preview of the resulting edit by providing a snapshot of the edit together with explanations on why the segment was selected for the edit to apply. (c) The *summary* panel allows the user to select edits the user wants to apply among generated edits.

3.1. User Scenario

To illustrate the envisioned user scenario of ExpressEdit, let's follow Lia, a businesswoman and a YouTube creator who wants to edit her video about entrepreneurship. She recorded a talking-head video that she wants to make more concise, engaging, and informative using ExpressEdit.

3.1.1. Creating a new edit

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

3.1.2. Describing the edit with Natural Language & Sketch

Lia decides to add text captions whenever she mentions valuable advice or tips. In the **Edit description** (Figure 1d), she types *"whenever there is a mention of advice or a tip, put it as a big white text with a transparent background on the bottom part of the frame"*. Additionally, Lia specifies the exact part of the frame where the text should appear using the **Sketch** function (Figure 1e) and draws the bounding box on the bottom half of the frame. She presses **Enter** to process the NL&S request and ExpressEdit provides (1) the breakdown of her command (Figure

2a), (2) a list of edits with respective reasoning and previews (Figure 1f) for each, and (3) a summary of the processing results with the checklist of generated edits (Figure 2b). Along with the summary, the system provides **Get More Edits** button that will generate more edits for the same video editing command, **Move to New Tab** button which moves the set of edits to a new tab (i.e., to a new layer on top of the video), and **Previous Summary** button which navigates to the summary of the previous request (if one exists).

3.1.3. Examining the results

To determine if the system understood her NL&S editing command, Lia examines the parsing results for the NL part. She looks at **Parsing Results** (Figure 2a) and sees that the parts of the NL command have colored backgrounds. To see the breakdown of the parsing results, she presses the **Reference Breakdown** and ExpressEdit shows **Temporal (when)** (i.e., *"whenever there is a mention of advice or a tip"*), **Spatial (where)** (i.e., *"bottom part of the frame"*), **Edit (what)** (i.e., *"put it as a big white text"*), and **Parameters (how)** (i.e., *"big white text"* and *"transparent background"*) references within the NL command, which assures her that the parsing was accurate.

3.1.4. Applying the generated edits

To decide on generated edits to apply, Lia quickly glances over the generated edits. For each generated edit, she looks at the **Reasoning** (*Temporal (when)* and *Spatial (where)* aspects) and **Preview** (Figure 2b) to decide if she wants to apply the edit. Additionally, the system allows her to test the edit by quickly "turning it on and off" by toggling the radio button. After making all the decisions on generated edits, Lia can press the **Get More Edits** button to ask for more edits for the same command or manually adjust/add her own edits.

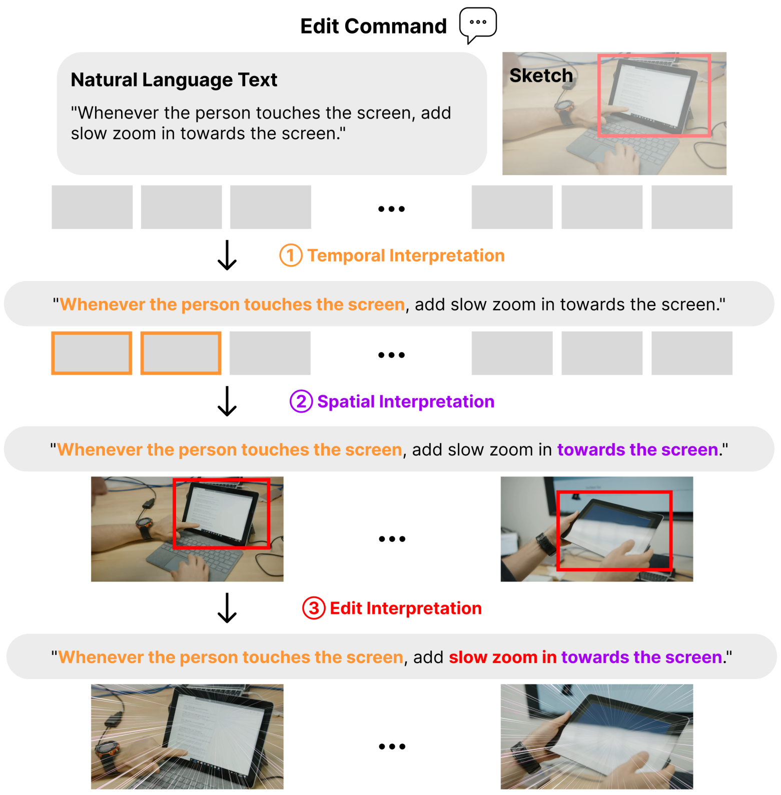
3.2. Edit operations

ExpressEdit supports seven edit operations: (1) text overlay, (2) image overlay, (3) shape overlay (i.e., circle, rectangle, star), (4) cutting out segments of the video, (5) zooming in/out, (6) cropping the video, and (7) blurring the video. We decided to focus only on visual edit operations as they cover important categories of parameters (temporal, spatial, edit-specific) common to other edit operations (e.g., audio-related edits, coloring edits). We believe this set of edit operations effectively demonstrates the feasibility of implementing various edit operations based on NL&S commands.

3.3. Implementation

ExpressEdit is implemented as a Web-based React[2] application. The backend server is based on Flask[3], which hosted the videos along with their transcript and processed users' requests. We obtained all the videos and transcripts from YouTube using the youtube-dl package [4].

[2]React
[3]Flask
[4]youtube-dl



Edit Command
Natural Language Text
"Whenever the person touches the screen, add slow zoom in towards the screen."
Sketch
1 Temporal Interpretation
"Whenever the person touches the screen, add slow zoom in towards the screen."
2 Spatial Interpretation
"Whenever the person touches the screen, add slow zoom in towards the screen."
3 Edit Interpretation
"Whenever the person touches the screen, add slow zoom in towards the screen."

Figure 3: Given a text and sketching edit command, our computational pipeline interprets its temporal, spatial, and edit operation and parameter references to implement the edit.

4. ExpressEdit: Pipeline

We designed our pipeline to interpret natural language and sketch (NL&S) descriptions of an edit (Figure 3). In order to facilitate the real-time interactions, we perform pre-processing to extract frame-level and clip-level metadata from the video that ExpressEdit uses to reason about video context. The metadata consists of segmentations of frames (based on Segment Anything model [10]) for every 1-second of the video and textual description of each 10-second clip (based on InternVideo [11] and BLIP-2 [12]) that are then summarized using GPT-3.5 [13]. The overview of the pipeline can be found in Appendix A.

We first parse the NL request and divide the language command into (1) temporal reference (i.e., any information in the NL command that could refer to a segment of the video), (2) spatial reference (i.e., any information in the NL command that could refer to location or region in the video frame), (3) edit operation reference (i.e., any information in the NL command that could indicate an edit operation to use), and (4) edit parameter reference (i.e., any information in the NL command that could refer to specific parameters of edit operation that was determined) with GPT-4 [14].

We interpret the temporal references using GPT-4 by decomposing them into "positional", "transcript-based", and "video-based" references, compiling all segments of the video that match these references, and pass the candidate segments (i.e., edits) further along the pipeline.

To interpret the spatial references in NL we first categorize them based on their dependency on the visual content of the video using GPT-4. For visual *content-dependent* references such as NL references to a specific object in the video, we extract representative frames for each candidate edit and then obtain the candidate spatial locations based on the segmentation that has the highest cosine similarity with the given NL text reference and sketch (if provided) using CLIP [15]. If no visual content-dependent references are detected and no sketch is given, we designate the top-left corner of the frame as the candidate spatial location. Then, we refine and resize each candidate spatial location with the visual *content-independent* spatial references (e.g., left, top, etc.) using GPT-4.

Lastly, using GPT-4, we identify the edit operations that are most suitable for the request based on edit operation references and our system's available edit operations: "text", "image", "shape", "blur", "cut", "crop", or "zoom". Then, we identify the video editing parameters corresponding to each predicted edit operation for candidate edits based on identified edit parameter references. To get text and images that are more appropriate to the video for "text" and "image" operations, we provide video context to guide the GPT-4 generation process of text and search query, respectively.

5. Pilot Study

We conducted a pilot user study with three novice video editors to (1) preliminarily evaluate the potential of ExpressEdit in supporting video editing workflows of novices and (2) obtain early feedback on the design of the system. The participants were allowed to freely use ExpressEdit for 25 minutes and edit a talking-head video about entrepreneurship[5] to make it more informative and engaging. We followed the think-aloud protocol and asked the participants to share their thoughts while editing the video.

During the pilot studies, we observed that novice video editors could intuitively use ExpressEdit to generate multiple edits based on their natural language and sketch (NL&S) command. When requesting new edits, the participants first confirmed that the NL&S editing command was understood correctly with **examine** panel (Figure 2a) that appears before any generated edits. Then, the **edit result** panel (Figure 2b) and 'turning on and off' feature allowed novices to quickly judge and add edits to the video. Furthermore, they were able to manually revise the generated edits by adjusting the edit operation and parameters. Additionally, the participants appreciated the convenience of applying edits with NL&S which effectively alleviated the manual efforts of making edits from scratch and allowed them to try out several edit ideas efficiently.

Since the pilot study lasted only 25 minutes, the participants made a few NL&S requests and spent the rest of the time applying the generated edits. Thus, we did not observe any notable use of the **summary** panel (Figure 2c), probably, due to its role of facilitating the management of many edits and the navigation between several NL&S requests. Additionally, because of the iterative nature of video editing, the participants had to constantly scroll through multiple panels whenever they wanted to turn on or off a specific generated edit. Since this behavior

[5]The video used for pilot study: How To SURVIVE As An Entrepreneur

made them feel less efficient, they wished for more effective support for navigation between generated edits.

6. Conclusion

We propose ExpressEdit, a multimodal video editing system that allows users to edit videos using natural language (NL) text and sketching on top of a video frame. The design of our system is motivated by findings from the formative study and the analysis of 176 multimodal expressions of edit requests. ExpressEdit can comprehend video editing commands and generate edits based on a technical pipeline comprised of CV and large language models that extracts and interprets (1) temporal, (2) spatial, and (3) operational references in an NL editing command and spatial references from sketching. Early feedback from the pilot study (N=3) suggests that ExpressEdit has the potential to greatly facilitate the expression of video editing intents for novice video editors.

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] L. Fiorella, R. E. Mayer, What works and doesn't work with instructional video, Computers in Human Behavior 89 (2018) 465–470. URL: https://www.sciencedirect.com/science/article/pii/S0747563218303376. doi:10.1016/j.chb.2018.07.015.

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

[3] A. Truong, F. Berthouzoz, W. Li, M. Agrawala, 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, 2016, pp. 497–507. URL: https://dl.acm.org/doi/10.1145/2984511.2984569. doi:10.1145/2984511.2984569.

[4] P.-Y. Chi, J. Liu, J. Linder, M. Dontcheva, W. Li, B. Hartmann, 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, 2013, pp. 141–150. URL: https://dl.acm.org/doi/10.1145/2501988.2502052. doi:10.1145/2501988.2502052.

[5] J. Kim, P. T. Nguyen, S. Weir, P. J. Guo, R. C. Miller, K. Z. Gajos, 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, 2014, pp. 4017–4026. URL: https://dl.acm.org/doi/10.1145/2556288.2556986. doi:10.1145/2556288.2556986.

[6] P. J. Guo, J. Kim, R. Rubin, 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, 2014, pp. 41–50. URL: https://dl.acm.org/doi/10.1145/2556325.2566239. doi:10.1145/2556325.2566239.

[7] M. Bétrancourt, K. Benetos, 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 (2018) 471–475. URL: https://www.sciencedirect.com/science/article/pii/S0747563218304102. doi:10.1016/j.chb.2018.08.035.

[8] T. Jokela, K. Mäkelä, M. Karukka, 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, 2007, pp. 482–489. URL: https://dl.acm.org/doi/10.1145/1378063.1378140. doi:10.1145/1378063.1378140.

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

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment Anything, 2023. URL: http://arxiv.org/abs/2304.02643. doi:10.48550/arXiv.2304.02643, arXiv:2304.02643 [cs].

[11] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang, S. Xing, G. Chen, J. Pan, J. Yu, Y. Wang, L. Wang, Y. Qiao, InternVideo: General Video Foundation Models via Generative and Discriminative Learning, 2022. URL: https://arxiv.org/abs/2212.03191v2.

[12] J. Li, D. Li, S. Savarese, S. Hoi, BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, 2023. URL: http://arxiv.org/abs/2301.12597. doi:10.48550/arXiv.2301.12597, arXiv:2301.12597 [cs].

[13] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, 2020. URL: https://arxiv.org/abs/2005.14165v4.

[14] OpenAI, GPT-4 Technical Report, 2023. URL: http://arxiv.org/abs/2303.08774. doi:10.48550/arXiv.2303.08774, arXiv:2303.08774 [cs].

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning Transferable Visual Models From Natural Language Supervision, 2021. URL: http://arxiv.org/abs/2103.00020. doi:10.48550/arXiv.2103.00020, arXiv:2103.00020 [cs].

A. Overview of the pipeline

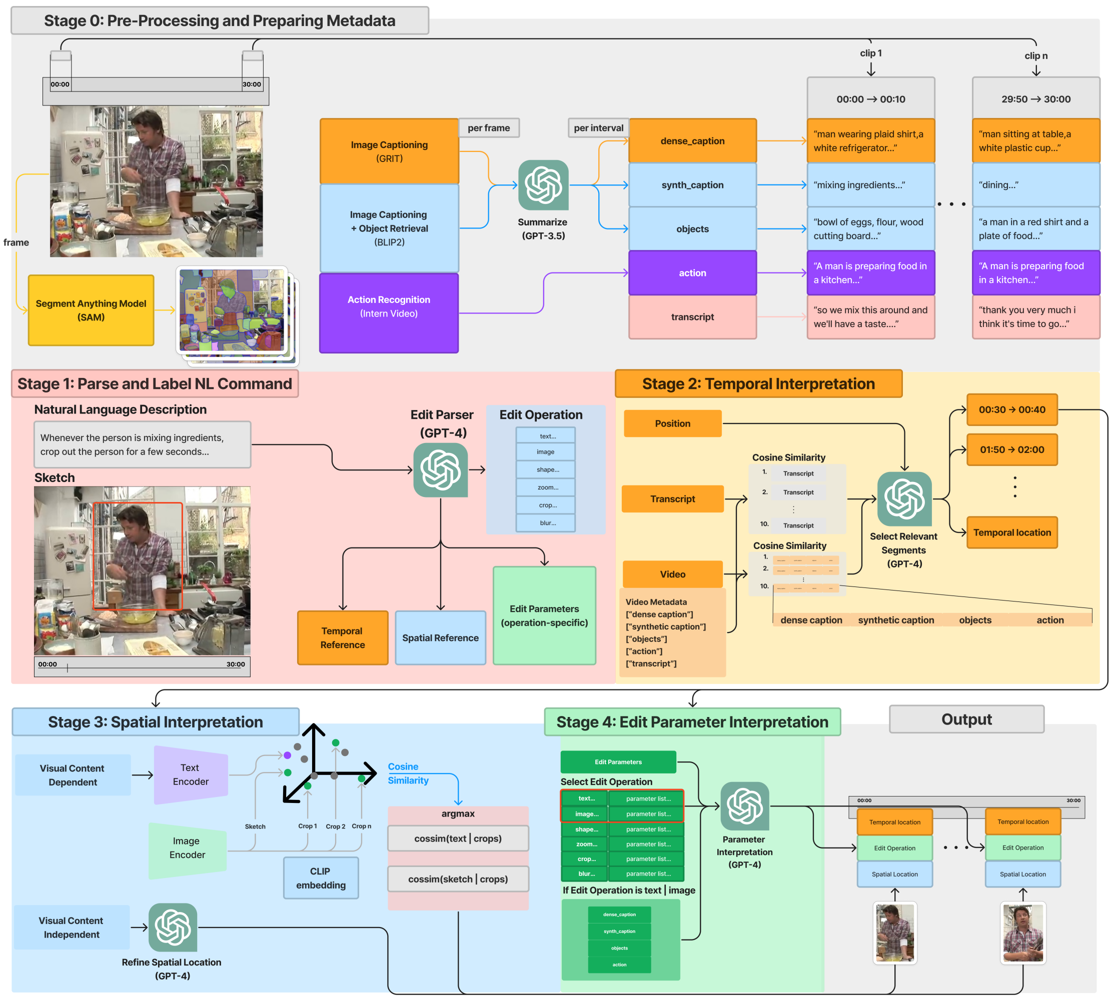

Stage 0: Pre-Processing and Preparing Metadata
frame
Segment Anything Model (SAM)
Image Captioning (GRIT)
Image Captioning + Object Retrieval (BLIP2)
Action Recognition (Intern Video)
per frame
per interval
Summarize (GPT-3.5)
dense_caption
synth_caption
objects
action
transcript
clip 1
clip n
00:00 → 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..."
"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
Edit Parser (GPT-4)
Edit Operation
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
Select Relevant Segments (GPT-4)
00:30 → 00:40
01:50 → 02:00
Temporal location
dense caption
synthetic caption
objects
action
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
If Edit Operation is text | image
Parameter Interpretation (GPT-4)
Output
Temporal location
Edit Operation
Spatial Location

Figure 4: 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.