Supplementary Materials for "Edit As You Wish: Video Caption Editing with Multi-grained User Control"

Anonymous Authors

In the supplementary material, we first introduce details of three approaches (Section A.2), then present more construction details about the VATEX-EDIT dataset (Section B) and EMMAD-EDIT dataset (Section C). Moreover, we provide the pseudo-code of the Position Accuracy metric (Section D) and the conversion instructions from interface signals to triplet format (Section E). Finally, based on a good start on the task, dataset and method foundation, we shed light on various interesting aspects worth exploring in the future (Section F).

A MODEL DETAILS

A.1 Architecture of OPA Model

The overall architecture of the specialist model, i.e. OPA, is depicted in Figure 1.

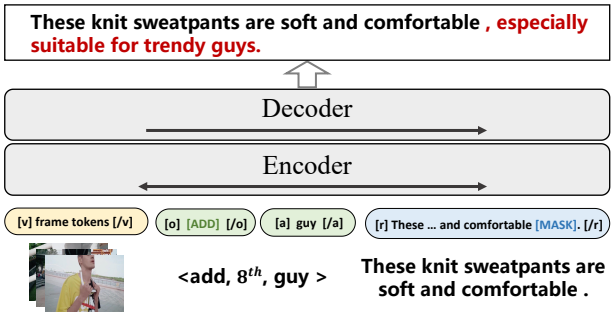


Figure 1: The overall OPA framework. 8^{th} denotes the specified position of the 8^{th} word.

A.2 Prompt of ImgLLM Pipeline

We build a ChatGPT pipeline with the visual expert model, i.e. InstructBLIP, to verify the vision-enhanced LLM performance on the VCE task. Specifically, we extract key frames from a given video and utilize InstructBLIP to produce detailed descriptions for each frame. We uniformly select 20 frames for VATEX-EDIT dataset and 30 frames for EMMAD-EDIT dataset, which is exactly the same frame number that our OPA model uses. These frame descriptions with timestamps help the ChatGPT to understand the exhaustive video content.

Designed Prompts. As Figure 3 illustrates, we conduct a well-designed prompt for ChatGPT including the task definition, helpful guidelines, an in-context learning [1] example, the video information, and the new case to be solved. With the input prompt, ChatGPT can output the desired edited caption with the required format.

In-Context Learning. Considering the differences between seven multi-grained commands, we manually select seven high-quality input-output demonstrations involving all command types. The command type of given examples will be aligned with the new case to fulfill better results. For example, if we want ChatGPT to edit a video description under the $\langle add, pos, attr \rangle$ command, we will give a matched example of $\langle add, pos, attr \rangle$ command to help it better solve the task.

A.3 Instruction-tuning Data of VidLLM

We convert the original samples from the VATEX-EDIT and EMMAD-EDIT datasets into an instruction-tuning format to facilitate the training of end-to-end video large language models. Figure 2 illustrates the converted data sample.

Question:
"You are an AI visual assistant to tackle a novel task called Video Caption Editing (VCE) Task. The task goal is to automatically revise an existing video description guided by flexible user requests and the original video content. The inputs of VCE task consist of a video, a reference description, and a user command. It outputs an edited video description based on the user command as a control signal.
1) Reference Description: This is the original sentence that describes a scene or action from the video. The [POS] token within this description indicates where specific changes should be made.
2) User Command: A sentence with specific guidance to clarify the edit requirements.

Inputs:
Reference Description: A man with a goatee is outside smoking a pipe on a cloudy day.
User Command: Shorten the reference description. "

Answer:
"A man lighting a tobacco pipe and smoking the pipe ."

Figure 2: A question-answer chat example used to conduct instruction tuning on the VidLLM.

B VATEX-EDIT DATASET DETAILS

B.1 Automatic Dataset Construction

We construct quadruples *(video, command, reference caption, edited caption)* according to two types of commands including **coarse-grained length-control commands** and **fine-grained attribute-related commands**. The overall procedure is depicted in Figure 4. We complement more details about constructing attribute-related commands.

Attribute-related commands. Our goal is to construct related *(command, edited caption)* samples for each *(video, reference caption)* to support attribute related commands in a "degradation" manner. The main challenges lie in two aspects: 1) extracting meaningful noun, verb, or modifier attributes in the reference caption R and 2) deleting the attribute-related semantic spans while maintaining the

You are an AI visual assistant to tackle a novel task called Video Caption Editing (VCE) Task. The task goal is to automatically revise an existing video description guided by flexible user requests and the original video content.

Task Definition:
The inputs of VCE task consist of a video, a reference description, and a user command. It outputs an edited video description based on the user command as a control signal.
Inputs:
1. **Reference Description:** This is the original sentence that describes a scene or action from the video. The [POS] token within this description indicates where specific changes should be made.
2. **User Command:** A sentence with specific guidance to clarify the edit requirements.
3. **Video Information:** Descriptions of extracted frames from the video. These are provided to understand the video context. Note: Sometimes, this information may be marked as 'None', indicating no specific video context is provided.

Guidelines:
- Make sure the output edited description is fluent and coherent.
- If the video context is unclear or not provided, base your edits on common or neutral assumptions about the scenario.
- If object identification from the Video Information is uncertain or seems erroneous, provide a more general description without detailed specifications.
- Ensure that the final edited description feels as if it's generated by an AI visual assistant who is watching the video in real time.

Examples:
1. **Reference Description:** A person is [POS] painting a picture.
2. **User command:** Add contents related to "brush" at [POS].
3. **Video Information:** *The frame at the second 1 shows ... The frame at the second 2 shows ... The frame at the second 3 shows .....*
4. **Expected Output:** A person is using a fine-tip brush and carefully painting a picture. "

Now I need your help to handle the following Video Caption Editing Task case and output the Edited Description. The output format should be "Expected Output: the edited sentence" without other outputs.

New Inputs:
Reference Description: A [POS] person is eating a cake.
User command: Expand the reference description and add video-related content at [POS].
Video Information:
The frame at the second 1 features a close-up view of a person's hand holding a paintbrush, which they are using to paint on a piece of paper or canvas. The person appears to be focused on their work, possibly creating a painting or illustration. In the background, there is a table with various objects, such as cups and a bottle, suggesting that the person might be working in a studio or creative space. The overall scene showcases the artist's attention to detail and dedication to their craft, as they skillfully use the paintbrush to bring their artistic vision to life.
The frame at the second 2 ...
.....

Figure 3: Designed prompts for the ImgLLM pipeline. It encompasses the task definition, helpful guidelines, an in-context learning [1] example, exhaustive visual descriptions, and the new case to be solved. With the well-designed prompt, ChatGPT can output the desired edited description with the required format. We manually construct seven in-context demonstrations involving each specific command type. For each input prompt, the specific command type of the in-context demonstration is aligned with the new case to help ChatGPT better understand the task.

fluency of rest content $R_{\backslash attr}$ to get an attribute-removed caption Y. In detail, we adopt four steps as follows:

- First, we use the Spacy[1] syntactic dependency parser to build a textual dependency tree that contains the Part-of-Speech information and relationships between tokens. We select reasonable branches in the parsed tree as attributes and further prune the branch to ensure the fluency of the rest caption.
- Second, we use a Semantic Role Labeling model [9] to analyze the semantic roles of each span in a sentence. It can help to better judge whether a parsed attribute span in the first step can be deleted or not, especially noun attributes.
- Third, we merge the attributes which only modify one or two tokens to improve the task challenge, that is, the model may be required to edit with multiple attributes in one round.
- Finally, considering the intrinsic error of the parsing model, we adopt a post-processing stage to filter low-quality sentences considering sentence fluency, edited token length, and attribute diversity. The ground-truth sentence fluency is further verified in Section B.2.

When the attribute-related spans are removed in a sentence, the edited positions can be naturally recorded. Through the above steps, we can get high-quality samples for the ⟨***del, pos, attr***⟩ commands. Meanwhile, reversed samples can be obtained for the ⟨***add, pos, attr***⟩ command by exchanging the reference caption and the edited caption. For these two fine-grained commands, the sentence length, structure, and semantics are controlled at the same time.

For commands ⟨***add, -, attr***⟩ and ⟨***del, -, attr***⟩ that omit positions, they mainly control the sentence length and semantics, not structure. We get relevant samples by relaxing the structure constraint based on the above "degradation" manner. In detail, we replace the edited caption by retrieving desired sentences satisfying

[1]https://spacy.io/

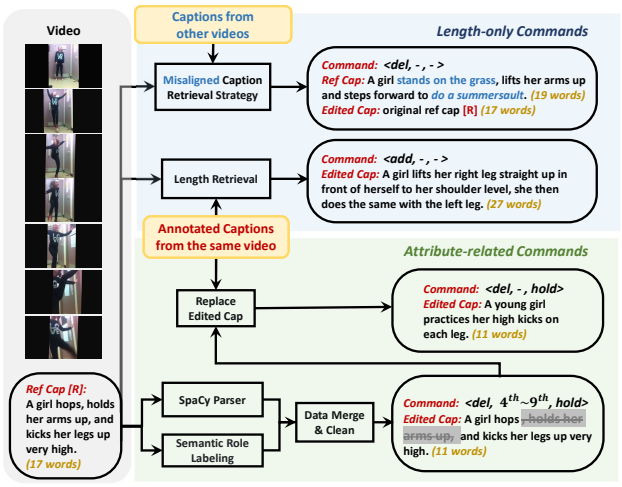


Figure 4: The automatic procedure of VATEX-EDIT dataset construction. The omitted commands, e.g. ⟨*add, pos, attr*⟩ *and* ⟨*del, pos, -*⟩, **can be easily converted from the data samples of shown commands.

Input Prompt: Forget everything you have been asked before, now you are a fluency assessment machine. Sentence fluency is defined as a sentence without spelling errors, grammatical errors, punctuation errors, etc. You don't have to worry about whether the sentence is redundant or not. You will be given several sentences in English, and you only need to determine whether the sentence is fluent but do not need to care about the specific content of the sentence. For each sentence, you must first state its number, then repeat the unchanged sentence, and further say whether it is fluent or not. If it is fluent, say "Yes", otherwise say "No". Please do not modify the original sentence, and do not output anything else.

Figure 5: Input prompts of ChatGPT fluency evaluation.

both length and semantics (attributes) constraints from original annotated descriptions for the same video.

	Constructed	Annotated
ChatGPT	87%	91%
Human	4.40	4.37

Table 1: Fluency evaluation of ground-truth sentences on the VATEX-EDIT test set. ChatGPT measures the fluency (YES/NO) rate of 100 sentences. Human measures the fluency rate (ranging from 1 to 5, 5 is the best) of 200 sentences.

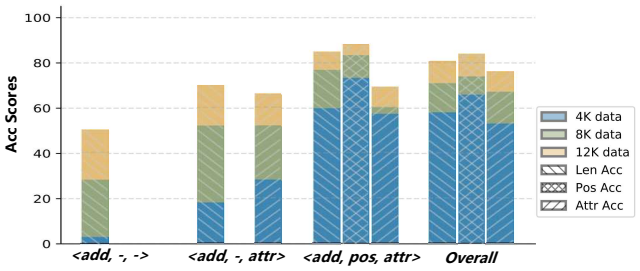


Figure 6: Overall and breakdown performance under different data volumes (4K/8K/12K data samples).

B.2 VATEX-EDIT Test set Quality

In the VATEX-EDIT dataset, it is worth noting that only the "⟨*del, pos, -*⟩, *shorten description at specified positions*" command uses the auto-constructed sentences as ground-truth captions in order to control the sentence structure. Meanwhile, the data samples of other commands utilize human-annotated sentences from the original VATEX dataset as ground-truth captions. To assess the quality of the auto-constructed sentences, we evaluate the test set of VATEX-EDIT through human and ChatGPT evaluations. For the ChatGPT[2] evaluation, we design suitable prompts (depicted in Figure 5) to guide ChatGPT to judge whether a sentence is fluent or not. We randomly sample 100 ground-truth sentences on the test set and calculate the fluency rate obtained by ChatGPT. For human evaluation, we recruit 20 crowd workers to rate the fluency score (ranging from 1 to 5) of 200 randomly sampled ground-truth sentences. As Table 1 shows, the automatically constructed ground-truth sentences of the "⟨*del, pos, -*⟩" command are as fluent as the human-annotated ground-truth sentences of other commands (87% vs 91% and 4.40 vs 4.37), which indicates the high quality of the VATEX-EDIT test set.

C EMMAD-EDIT DATASET DETAILS

We manually collect the high-quality dataset EMMAD-EDIT in the E-commerce domain based on the Chinese E-commerce video captioning dataset E-MMAD. The data sample in E-MMAD dataset consists of a product video, an advertising video description, and additional information including video titles and structure attributes. It collects 120,984 videos with average duration of 30.4 seconds and the annotated Chinese description length is 67 words on average. The characteristics of *long videos* and *long captions* make it suitable for building a challenging VCE dataset. During annotation, we select data samples from E-MMAD dataset with relatively long descriptions. In addition to videos and descriptions, we also provide product structure information and video titles for reference.

C.1 EMMAD-EDIT dataset statistics

Table 2 shows breakdown data statistics of the EMMAD-EDIT dataset. It has overall 79,652 editing instances for 16,176 product videos. Diverse unique attributes and vocabulary also indicate data richness. We separate the *abstract* attribute-related data samples as

[2]https://openai.com/blog/chatgpt

Table 2: Data statistics of EMMAD-EDIT dataset. *VTime* denotes the average time length (seconds) of videos. Len_{Ref} denotes the average length of reference captions and Len_{GT} is the length of groundtruth captions. *Uni. Attrs* means the vocabulary of annotated attributes. $Overall_{Abstract}$ is the challenging subset of abstract attribute-related samples.

Command	#Videos			#Editing instances			VTime	Len_{Ref}	Len_{GT}	Edit Dist	Uni. Attrs	Vocab
	Train	Val	Test	Train	Val	Test						
⟨*add*, - , - ⟩	7,751	2,599	2,633	7,752	2,599	2,633	26.6	72.2	100.9	29.8	-	29,241
⟨*add*, *pos*, - ⟩	3,221	1,077	1,094	3,221	1,077	1,094	26.3	91.6	99.4	8.7	-	18,640
⟨*add*, - , *attr*⟩	3,221	1,078	1,094	3,221	1,078	1,094	27.1	93.1	100.6	8.7	2,388	18,501
⟨*add*, *pos*, *attr*⟩	3,221	1,077	1,093	3,221	1,077	1,093	26.6	92.0	99.9	8.7	2,907	18,584
⟨*del*, - , - ⟩	7,751	2,599	2,633	7,753	2,599	2,633	26.4	100.1	71.4	29.6	-	29,456
⟨*del*, *pos*, - ⟩	3,221	1,078	1,095	3,221	1,078	1,095	27.1	102.1	86.1	17.2	-	18,734
⟨*del*, - , *attr*⟩	3,221	1,078	1,095	3,221	1,078	1,095	27.2	102.7	86.2	17.8	3,038	18,924
$Overall_{Specific}$	16,176	5,418	5,502	31,610	10,586	10,737	26.9	91.3	90.4	20.8	6,003	44,725
$Overall_{Abstract}$	15,955	5,328	5,432	15,959	5,328	5,432	26.8	90.9	100.9	11.4	648	44,347

an extra challenging subset. Considering realistic demands, we utilize all these data samples to construct "⟨*add, -, attr*⟩*, add attributes in description*" command cases. Note that in the Experiments section, we present the EMMAD-EDIT results training without the *abstract* subset data by default.

C.2 The Impact of Data Volume.

Considering the limited scale of manually collected data in EMMAD-EDIT, we analyze the results under different volume data with 4K, 8K, 12K samples. Figure 6 shows that a growing volume of data consistently increases the controllable scores. Breakdown analysis of multi-grained commands reveals that more challenging commands, e.g. ⟨*add,-,-*⟩, require higher volume of training data samples to get desired performance.

C.3 Data Visualization

We manually collect data samples that support the editing of two types of attributes: *specific* and *abstract*. *Specific* attributes directly appear in the reference caption to support straightforward content editing such as "comfortable" and "fashion". Meantime, *abstract* attributes are more high-level concepts that may consist of multiple specific attributes. For example, "Style", "Target People", and "Time and Seasons" are annotated *abstract* attributes in the dataset. The word clouds (shown in Figure 7) of *specific* and *abstract* attributes show the diversity of the EMMAD-EDIT dataset.

We visualize annotated samples of the EMMAD-EDIT dataset in Figure 8. Besides quadruples *(video, command, reference caption, edited caption)* that support video caption editing, we also provide additional product information such as structured information and the video title. The annotation interface during data construction is shown in Figure 9.

D METRIC DETAILS

To support evaluations in Chinese, we utilize a Chinese GPT-2 [2] to calculate Chinese sentence likelihood and then obtain PPL scores. For EMScore, we replace the core vision-language alignment model EN-CLIP [8] with CN-CLIP [11].

(a) Specific Attribute (b) Abstract Attribute

Figure 7: Word clouds of attributes on the EMMAD-EDIT dataset. The left shows the diversity of annotated specific attributes and the right shows the abstract attributes.

D.1 Position Accuracy Design

We propose a novel *Position Accuracy (Pos-Acc)* metric to measure whether models insert/remove the content in the specified positions under fine-grained editing control. The challenge of calculating Pos-Acc is the misalignment between the reference caption R and the edited caption Y. The two textual sentences have variable lengths and the edited operation may appear in the other positions to maintain the overall fluency. To tackle the above challenge, we propose a novel Dynamic Sequence Aligning (DSA) algorithm to align two variable-length textual sequences based on the absolute positions, inspired by classical Dynamic Time Warping (DTW) [6].

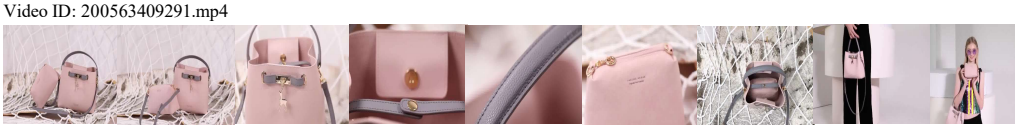
The pseudo-code is presented in Figure 10, which can align two variable-length text sequences in positional indexes, resulting in related spans $\{S_{m_1}, S_{m_2}, \ldots, S_{m_K}\}$ in Y aligned to [MASK] tokens $\{m_1, m_2, \ldots, m_K\}$ in R. We count the percentage of correct samples that insert/remove new content S_{m_K} in given position m_K.

As Figure 11 illustrates, we visualize two aligned cases in English and Chinese respectively obtained by the DSA algorithm.

E CONVERSION FROM INTERFACE SIGNALS TO TRIPLET FORMAT

Though the main focus of this paper lies in exploring the novel VCE task utilizing triplet commands, we also shed light on how to

Video ID: 2005634092911.mp4



Video Title	VH女包2020新款潮流单肩包时尚简约小鹿水桶包休闲女士手提斜挎包 *VH Women's Bag 2020 New Trend Single-shoulder Bag Fashionable and Simple Little Deer Bucket Bag Casual Ladies' Handheld Crossbody Bag*
Structured Info.	品类:斜挎包,单肩包,水桶包,女包;时间季节:2020;新品:新款;风格:时尚,休闲,简约,潮流,欧美时尚;修饰:手提,小鹿;人群:女士;上市时间:2018年春夏; 款式:单肩包;里料材质:织物;背包方式:单肩斜挎手提; 品牌:VANESSA HOGAN;颜色分类:香草白1,黑色,香草白,婴儿粉;皮革材质:牛皮; *Category: Crossbody Bags, Shoulder Bags, Bucket Bags, Women's Bags; Season: 2020; New Product: New Model; Style: Fashionable, Casual, Simple, Trendy, European and American Fashion; Embellishment: Handheld, Little Deer; Target Audience: Women; Launch Date: Spring/Summer 2018; Style: Shoulder Bag; Lining Material: Fabric; Backpack Method: Single Shoulder Crossbody Handheld; Brand: VANESSA HOGAN;*
Command	***<add, -, ->***
Ref Cap	经典水桶包，外型酷似水桶，包身圆润又不失俏皮，别致的子母包设计，凸现自我的创意个性与个人色彩。 *The classic bucket bag has a cool bucket-shaped appearance, with a rounded and nifty body. The unique design of the parent-child bag highlights the creative personality and individual style.*
Edited Cap	经典水桶包，**子母包设计，再现繁盛时代的小鹿包。**外形酷似水桶**外型**，包身圆润又不失俏皮**的造型**，别致的子母包设计，**追求简约实用的时尚**，凸现自我的创意个性与个人色彩。 *The classic bucket bag **with a parent-child bag design brings back the prosperous era of little deer bags.** Its cool bucket-shaped appearance, rounded and nifty body, and unique design **pursue minimalist and practical fashion,** highlighting the creative personality and individual style.*
Command	***<del, -, "实用, 俏皮">*** *<del, -, 'nifty, practical'>*
Ref Cap	经典水桶包，子母包设计，再现繁盛时代的小鹿包。外形酷似水桶外型，包身圆润又不失俏皮的造型，别致的子母包设计，追求简约实用的时尚，凸现自我的创意个性与个人色彩。 *The classic bucket bag with a parent-child bag design brings back the prosperous era of little deer bags. Its cool bucket-shaped appearance, rounded and nifty body, and unique design pursue minimalist and practical fashion, highlighting the creative personality and individual style.*
Edited Cap	经典水桶包，子母包设计，再现繁盛时代的小鹿包。外形酷似水桶~~(外型)~~，包身圆润~~(又不失俏皮的造型)~~，别致的子母包设计，追求简约~~(实用的)~~、时尚，凸现自我的创意个性与个人色彩。 *The classic bucket bag with a parent-child bag design brings back the prosperous era of little deer bags. Its cool bucket-shaped appearance, rounded ~~(and nifty)~~ body, and unique design pursue minimalist ~~(and practical)~~ fashion, highlighting the creative personality and individual style.*
Command	***<add, -, "风格">*** *<add, -, 'style'>*
Ref Cap	经典水桶包，子母包设计，再现繁盛时代的小鹿包。外形酷似水桶，包身圆润又不失俏皮，别致的子母包设计，追求实用，凸现自我的创意个性与个人色彩。 *The classic bucket bag with a parent-child bag design brings back the prosperous era of little deer bags. Its cool bucket-shaped appearance, rounded and nifty body, and unique design pursue practical usage, highlighting the creative personality and individual style.*
Edited Cap	经典水桶包，子母包设计，再现繁盛时代的小鹿包。外形酷似水桶**外型**，包身圆润又不失俏皮**的造型**，别致的子母包设计，**追求简约实用的时尚**，凸现自我的创意个性与个人色彩。 *The classic bucket bag with a parent-child bag design brings back the prosperous era of little deer bags. Its cool bucket-shaped appearance, rounded and nifty body, and unique design pursue **minimalist and practical fashion**, highlighting the creative personality and individual style.*

Figure 8: Data samples of annotated EMMAD-EDIT dataset.

convert the prevalent interface signals, i.e. natural language and handwriting editing trajectories, to the triplet formatted controls.

E.1 From Natural Language to Triplets

To convert natural language signals, we can adopt fuzzy matching to recognize add or delete operations and use text parser tools to get specific semantic roles of a sentence. We can also leverage the ability of LLMs such as ChatGPT and LLaMA-2, which already have outstanding language understanding and summarization capabilities. Specifically, we can design suitable prompts to convert text sentences into triplets to meet the pre-defined requirements.

E.2 From Triplets to Natural Language

The triplet command in our annotated datasets can be easily converted to natural languages to support more diverse scenarios and applications. For example, using natural language to explore the capability of LLMs in the VCE task (A.2). Specifically, we design a series of templates shown in Table 3 to convert the triplet command to natural language. We will also release the two benchmark datasets with user commands both in the triplet and natural language format to benefit the community.

E.3 From Handwriting-revision Traces to Triplets

The recent advancement of GPT-4Vision (GPT-4V) has shown its powerful capability of Visual Referring Prompting [12]. GPT-4V can well understand visual pointers (such as circles, arrows or traces) directly drawn on images, therefore, revealing a novel human-model interaction method called "visual referring prompting". Combined with its accurate OCR capability, GPT-4V can serve as an ideal tool to input the handwriting editing traces as an image and convert it to the triplet control output, as illustrated in Figure 12.

F FUTURE DIRECTIONS

In this paper, we make the first attempt to propose the novel Video Caption Editing (VCE) task and collect two benchmark datasets

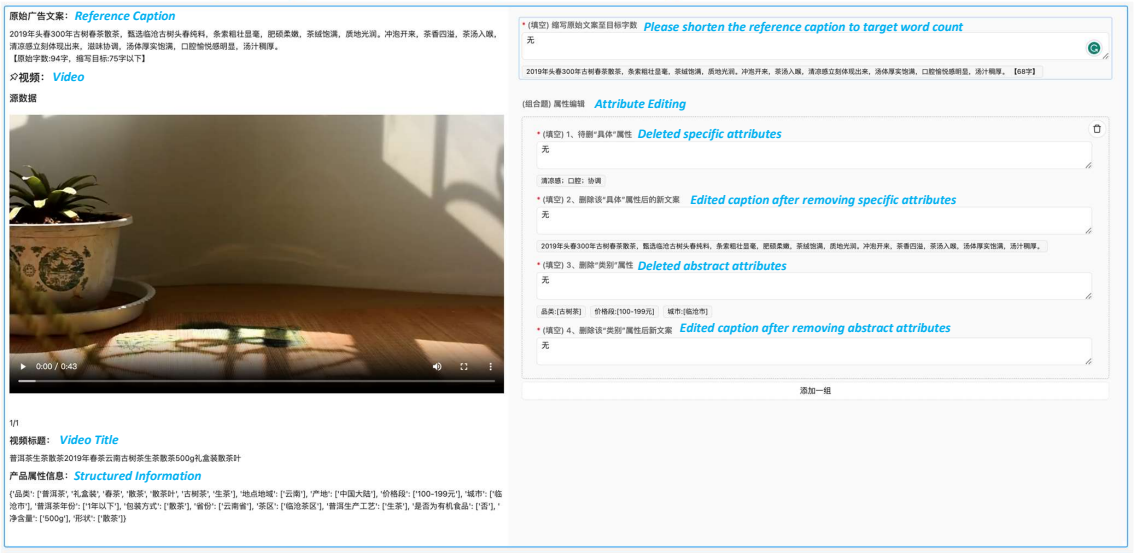


Figure 9: Annotation interface for constructing the EMMAD-EDIT dataset.

Command	Conversion Template
⟨*del*, - , *attr* ⟩	Delete contents about '{}' from the reference description.
⟨*add*, *pos*, *attr*⟩	Add contents about '{}' at [POS].
⟨*add*, - , *attr*⟩	Add contents about '{}' to expand the reference description.
⟨*add*, *pos*, - ⟩	Add video-related contents at [POS].
⟨*del*, *pos*, - ⟩	Contents have been deleted at [POS], please make the rest sentence fluent and coherent.
⟨*del*, - , - ⟩	Shorten the reference description.
⟨*del*, - , *attr*⟩	Expand the reference description.

Table 3: Defined templates that conveniently convert triplet commands to natural language format. The '{}' represents the placeholder of specific attributes.

to support it. We further propose a unified framework OPA for VCE and compare it with a ChatGPT pipeline. Based on the task, dataset and method foundation, we aim to make a good start for the community. There are various interesting aspects worth exploring in the future:

- **A versatile system for video captioning and editing.** The dense annotation of the quadruple *(video, command, reference caption, edited caption)* in our dataset has the potential to support building a versatile system encompassing conventional video captioning, controllable video caption and video caption editing. For initialization, conventional video captioning can produce a generated description for a given video. subsequently, video caption editing can be utilized to update and revise the original description. When omitting the reference caption, the rest annotation *(video, command, edited caption)* can be adjusted to achieve controllable video captioning.
- **A more robust system for poor reference caption.** In the VCE task, the reference caption can be the edited caption from the last round to fulfill multi-round editing. Furthermore, it can also be extended with human-written drafts or machine-generated captions. Under these circumstances, the robustness of the VCE system to low-quality reference captions should be taken into consideration.
- **The abstract-attribute subset of EMMAD-EDIT is underexplored.** The abstract-attribute subset (details in Sec 4.2.) involves abstract attributes that do not directly appear in the reference caption. It requires models to understand and reason the video content at a higher semantic level. In the experiments, we only assess the performances of OPA and ChatGPT pipeline on the easier *specific* subset, whose performances are not yet so desirable, the *abstract* subset therefore remains a challenge for future exploration.
- **Serve as a touchstone for video large language models (VidLLMs).** The recent emergence of VidLLMs such as GPT4-Vision [7], VideoChat [4], and VideoChatGPT [5] has ignited sparks for a generalist video assistant. However, existing research also points out their limitations for long video understanding [3], visual hallucination [10] and multi-modal instruction following capability [13]. The VCE benchmark can serve as a new multi-modal evaluation for assessing both long video understanding and multi-grained text editing abilities.

```python
# R is the word sequence of Reference Caption
# Y is the word sequence of Edited Caption
def DynamicSeqAlign(R, Y):
    # Initialize DTW distance matrix
    DTW = {}
    for i in range(len(R)):
        DTW[(i, -1)] = float('inf')
    for j in range(len(Y)):
        DTW[(-1, j)] = float('inf')
    DTW[(-1, -1)] = 0
    # Initialize the aligned path matrix
    PATH = {}
    # Dynamic programming to align R and Y
    for i in range(len(R)):
        for j in range(len(Y)):
            # get distance between word i and word j
            dist = get_dist(R[i], Y[j])
            # get the min distance from last time step
            min_dist = min(DTW[(i-1, j)], DTW[(i, j-1)
                ], DTW[(i-1, j-1)])
            # update DTW matrix for current time step
            DTW[(i, j)] = dist + min_dist
            # record the related distance path
            PATH = update(PATH)
    # get the minimum distance path aligning R and Y
    best_path = trace(PATH)
    # filter repetitive words in Y
    align_path = filter(best_path)
    return align_path

# get distance between two words from R and Y
def get_dist(R_word, Y_word):
    if R_word == Y_word:        # same words
        return 0
    elif R_word == '[MASK]':    # ([MASK], Y_word)
        return 100
    else:                       # unmatched words
        return float('inf')
```

Figure 10: Pseudo-code for Dynamic Sequence Aligning in a Python-like style.

REFERENCES

[1] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[2] Zeyao Du. 2019. GPT2-Chinese: Tools for training GPT2 model in Chinese language. https://github.com/Morizeyao/GPT2-Chinese.

[3] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. 2023. Chat-UniVi: Unified Visual Representation Empowers Large Language Models with Image and Video Understanding. *arXiv preprint arXiv:2311.08046* (2023).

[4] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023).

[5] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. *ArXiv* abs/2306.05424 (2023). https://api.semanticscholar.org/CorpusID:259108333

[6] Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion* (2007), 69–84.

[7] OpenAI. 2023. GPT-4V(ision) System Card. https://cdn.openai.com/papers/GPTV_System_Card.pdf.

[8] 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. In *ICML*.

[9] Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255* (2019).

[10] Yin Shukang, Fu Chaoyou, Zhao Sirui, Xu Tong, Wang Hao, Sui Dianbo, Shen Yunhang, Li Ke, Sun Xing, and Chen Enhong. 2023. Woodpecker: Hallucination Correction for Multimodal Large Language Models. *arXiv preprint arXiv:2310.16045* (2023).

[11] An Yang, Junshu Pan, Junyang Lin, Rui Men, Yichang Zhang, Jingren Zhou, and Chang Zhou. 2022. Chinese CLIP: Contrastive Vision-Language Pretraining in Chinese. *arXiv preprint arXiv:2211.01335* (2022).

[12] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision). arXiv:2309.17421 [cs.CV]

[13] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023. mPLUG-Owl2: Revolutionizing Multi-modal Large Language Model with Modality Collaboration. arXiv:2311.04257 [cs.CL]

Command: *<add, pos, attr>*		
EN Case	Ref Cap	"A woman gives a demonstration [MASK-1] to come to her [MASK-2] sessions [MASK-3] ."
	Edited Cap	"A woman is giving a demonstration and invitation to come to her gong therapy sessions ."
	Pos Align	([MASK-1], "and invitation", ✓); ([MASK-2], "gong therapy", ✓); ([MASK-3], None, ✗)
CN Case	Ref Cap	"此款皮鞋精选 [MASK-1] 头层牛皮，采用擦色 [MASK-2] 打磨工艺；鞋帮选用松紧套脚，穿着上更加的便捷。"
	Edited Cap	"此款 高帮皮鞋精选 优质 头层牛皮，采用擦色 手工 打磨工艺；鞋帮选用 的是 松紧套脚，穿着上更加的便捷。"
	Pos Align	([MASK-1], "优质", ✓)； ([MASK-2], "手工", ✓)

Figure 11: Cases of aligned reference captions and edited captions by the proposed Dynamic Sequence Aligning Algorithm.

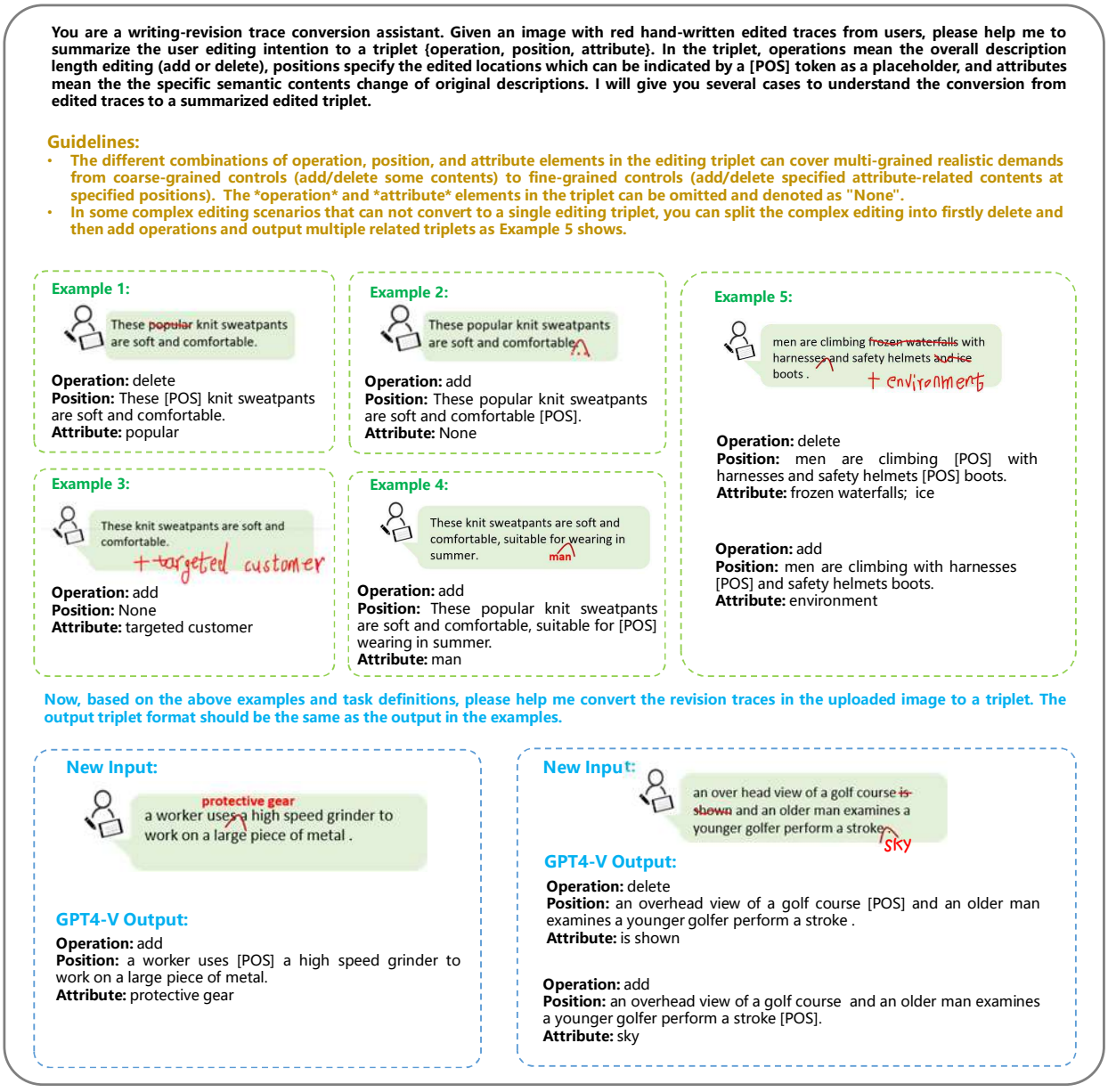


Figure 12: Designed prompts and output instances of GPT-4V to convert the human-writing editing trace as an image to the triplet format control.