
Supplementary File for Paper 13524: Unleashing Hour-Scale Video Training for Long Video-Language Understanding

Anonymous Author(s)

Affiliation

Address

email

A Appendix

This document provides more implementation details of the VideoMarathon dataset construction, additional experimental details, limitation, and broader impacts, organized as follows:

- **Details of VideoMarathon Construction** (Section A.1). We involve the exact prompts and examples for hierarchical video captioning and topic-specific QA generation.
- **More Experimental Details** (Section A.2). We present additional experimental details, including detailed training schedules for Hour-LLaVA and experimental settings of ablation studies in Section 5.3, 5.4, and 5.5.
- **Limitations** (Section A.3). We discuss several limitations of this study.
- **Broader Impacts**(Section A.4). We analyze both potential positive societal impacts and negative societal impacts of this work.

A.1 Details of VideoMarathon Construction

A.1.1 Details of Hierarchical Video Captioning

Prompts. We provide the exact prompts used for hierarchical video captioning, including clip-level captioning (Figure 5), event-level captioning (Figure 7), and global-level captioning (Figure 8). Additionally, the prompt used for event splitting is presented in Figure 6.

Examples. Figure 12 present several examples of hierarchical video captions, including clip-level, event-level, global-level video captions, and event splits. In addition, Figure 13 presents the word cloud of all the global-level video descriptions in the VideoMarathon dataset.

A.1.2 Details of QA Generation

Prompts. We present the exact prompts used for topic-specific QA generation, including the open-ended (OE) QA prompt in Figure 9 and the multiple-choice (MC) QA prompt in Figure 10. Also, the detailed descriptions of the 22 sub-tasks across six core topics are provided in Figure 11. Please refer to the data file for exact QA examples from the VideoMarathon dataset.

A.2 Additional Experimental Details

A.2.1 Detailed Training Schedules for Hour-LLaVA

In Section 4, we briefly introduce the training schedules of Hour-LLaVA. Furthermore, Table 6 presents the detailed training schedules for each training stage of Hour-LLaVA, containing compression details, data usage, and training hyperparameters.

	Image-Language Pretraining		Video-Language Adaptation		Video Instruction Tuning	
	3B	7B	3B	7B	3B	7B
Compression	FPS	1	1	1	1	1
	Spatial Forgetting (SF)	1/4	1/4	1/4	1/4	1/4
	SF Mechanism	Random	Random	Random	Random	Random
	Temporal Forgetting (TF)	-	-	1/4	1/4	1/4
Data	TF Mechanism	-	-	Uniform	Uniform	Uniform
	Data Type	SI	SI	T + SI + MI + SV	T + SI + MI + SV + LV	T + SI + MI + SV + LV
	Data Sources	OV-SI	OV-SI	OV-SI + L.V.	OV-SI + L.V. + V.M.	OV-SI + L.V. + V.M.
	#Samples	3B	3B	0.6M	4.4M	4.4M
Training	Batch Size	128	128	128	128	256
	LR of Vision Encoder	0	0	5×10^{-6}	5×10^{-6}	0
	LR of Projector	0	0	1×10^{-4}	1×10^{-4}	1×10^{-4}
	LR of MemAug	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
	LR of LLM	0	0	2×10^{-5}	1×10^{-5}	2×10^{-5}
	Epoch	1	1	1	1	1

Table 6: **Detailed training schedule for each training stage of Hour-LLaVA**, including compression details, data usage, and training hyperparameters. In *Data* part, we use the following abbreviations: T for text, SI for single-image, MI for multi-image, SV for short-video, LV for long-video, OV for LLaVA-OV-SI [3], L.V. for LLaVA-Video-178K [8], and V.M. for VideoMarathon.

30 A.2.2 Experimental Settings of Ablation Studies.

31 Due to page limitations, we are unable to include all experimental settings for the ablation studies in
32 Section 5. Below, we provide the experimental settings for each ablation study. We conduct all the
33 ablation studies using Hour-LLaVA-3B model.

34 **Ablation Study for VideoMarathon** (Section 5.3). In this section, we investigate how performance
35 changes with different mixtures of VideoMarathon (long-video) and LLaVA-Video-178K (short-
36 video) training data. We construct two subsets from VideoMarathon and LLaVA-Video-178K,
37 matched in both the number of videos and the number of video-language instruction training samples.
38 Each sub-dataset consist of 70K video-language instruction samples over 10K videos. We then
39 train the Hour-LLaVA-3B model on different mixtures of these two subsets (as shown in Figure 3),
40 while also incorporating the same 60K multimodal samples from LLaVA-OV [3] (*i.e.*, 20K text-only,
41 30K single-image, and 10K multi-image samples) to support multimodal learning. In addition, the
42 Hour-LLaVA-3B model is initialized from the checkpoint obtained after image-language pretraining
43 (Stage 1 in Table 6). All training hyperparameters follow the settings used for *video instruction*
44 *tuning* of the 3B model, as detailed in Table 6.

45 **Comparison with Existing Video Token Compression Techniques** (Section 5.4). We compare the
46 proposed MemAug with several existing video token compression techniques, including uniform [2,
47 6], keyframe [4, 5], and user question-guided [1, 4] temporal compression. Also, we implement
48 random temporal compression as a reference baseline. In particular, the implementation details of
49 different temporal compression methods are shown as below:

- 50 • **Random.** We randomly select $\frac{1}{4}$ frames from a video sampled at 1 FPS sampling.
- 51 • **Uniform.** We uniformly samples $\frac{1}{4}$ frames across the temporal dimension from a 1 FPS video.
- 52 • **Keyframe.** Following [4], for each frame, we compute the average cosine similarity with its K
53 nearest temporal neighbors (with $K = 8$). We then retain the $\frac{1}{4}$ frames that are least similar to
54 their neighbors, removing the $\frac{3}{4}$ most redundant frames. In addition, video frame features are
55 extracted by SigLIP vision encoder [7].
- 56 • **User question-guided.** Following [1], we calculate the cosine similarity between the frame
57 embedding \mathbf{v} and question embedding \mathbf{q} . Next, we select $\frac{1}{4}$ frames with the highest similarity
58 scores. In particular, the frame embedding \mathbf{v} is computed as $\mathbf{v} = \frac{1}{N_v} \sum_{i=1}^{N_v} \mathbf{v}_i$, where N_v is the
59 number of tokens per frames and \mathbf{v}_i refers to the i -th visual token vector. The question embedding
60 is calculated as $\mathbf{q} = \frac{1}{N_q} \sum_{i=1}^{N_q} \mathbf{q}_i$, where N_q is the number of tokens in the given question and \mathbf{q}_i
61 denotes the i -th query token vector. Additionally, \mathbf{v}_i is obtained using the SigLIP vision encoder
62 followed by the projector, while \mathbf{q}_i is derived from the Qwen2.5-3B word embedding model.

63 For a fair comparison, all the above temporal compression techniques apply different temporal
64 compression strategies with the same compression ratio of $\frac{1}{4}$ on 1 FPS input videos, along with a
65 random spatial compression with a compression ratio of $\frac{1}{4}$.

66 Following a similar training configuration to Section 5.3, we use a training set composed of 70K video-
67 language instruction samples with a 75%/25% mixture of VideoMarathon and LLaVA-Video-178K,
68 respectively, and involve 60K multimodal samples from LLaVA-OV. All training hyperparameters
69 are aligned with the 3B model’s *video instruction tuning* setup in Table 6.

70 **Ablation Study for Hour-LLaVA** (Section 5.5). In this section, we primarily conduct analysis on
71 two key components in Hour-LLaVA: forgetting mechanisms and memory repository.

72 We ablate **forgetting mechanisms** from both spatial and temporal perspectives.

- 73 • **Spatial Forgetting.** In this setting, we adopt 3B LLaVA-OV-SI data for training. The vision
74 encoder, projector, and LLM decoder of Hour-LLaVA-3B model is initialized by a pretrained
75 LLaVA-OV-SI-3B model as mentioned in Section 5.1. During training, only the parameters of the
76 MemAug module are tuned, while the rest of the model remains frozen. Training hyperparameters
77 follow the settings used for *image-language pretraining* of the 3B model in Table 6.
- 78 • **Temporal Forgetting.** We adopt the same training configuration as in Section 5.4, using a dataset
79 composed of 70K video-language instruction samples with a 75%/25% mixture of VideoMarathon
80 and LLaVA-Video-178K, along with 60K multimodal samples from LLaVA-OV. The Hour-
81 LLaVA-3B model is initialized from the checkpoint obtained after image-language pretraining.
82 All training hyperparameters follow the *video instruction tuning* setup for the 3B model.

83 For **memory repository**, we mainly analyze the impact of the memory repository scale in Figure 4
84 (*middle*). The 100% scale refers to storing full video tokens sampled at 1 FPS. For the 50% and 25%
85 scales, we uniformly sample 50% and 25% of the frame-level features along the temporal dimension,
86 respectively. The <10% scale denotes a lightweight configuration in which only decayed video
87 tokens are replaced in the memory repository, resulting in fewer than 10% of the full video token
88 count. For the training, we follow the same setup as in Section 5.4. All training hyperparameters are
89 consistent with the *video instruction tuning* configuration of the 3B model in Table 6.

90 A.3 Limitations

91 Despite the promising results of Hour-LLaVA, several limitations remain. First, due to the lack of
92 comprehensive evaluation metrics for hour-long video-language understanding, multi-choice QA
93 remains the most practical task for evaluating long video-language models. However, this evaluation
94 format is limited in scope and fails to assess broader capabilities of Video-LMMs. The development
95 of more diverse and holistic benchmarks is therefore essential for advancing this field. Second,
96 Hour-LLaVA is trained on large-scale synthetic instruction-following data, which inevitably contains
97 noise. The current training pipeline does not explicitly consider this issue, and future work can
98 further explore noise-robust training strategies. Third, the current framework is limited to video
99 and language modalities, neglecting audio, a crucial component in many long-form videos such as
100 lectures, interviews, and documentaries. Incorporating audio or additional modalities could further
101 enhance the model’s capacity for comprehensive multimodal understanding.

102 A.4 Broader Impacts

103 This study presents significant advancements in long-form video-language modeling through the in-
104 troduction of the VideoMarathon dataset and the Hour-LLaVA model. Positively, the VideoMarathon
105 dataset paves the way for more sophisticated AI systems capable of handling realistic, long-duration
106 scenarios, which are essential for practical applications in education, security, autonomous driving,
107 and augmented or virtual reality. However, potential negative impacts include the risk of misuse
108 in surveillance, such as continuous monitoring and profiling of individuals in public or private set-
109 tings without consent, and the possibility of misinterpreting nuanced or sensitive content in long
110 videos, which might lead to harmful decisions in critical domains like healthcare or security. These
111 implications highlight the importance of responsible development practices, including robust privacy
112 safeguards and clear ethical guidelines for the deployment of long-form video-language models.

Clip-level Video Captioning

You are given a 30-second video clip. Your task is to generate a detailed, structured description of the video based on six specific perspectives.

Instructions:

Please ensure your descriptions are vivid, precise, and rich in detail. For each of the six topics below, please provide a comprehensive caption. Your goal is to clearly convey the video's visual and contextual content from multiple dimensions:

- **Temporality:** Describe how the scene evolves over time. Highlight key transitions, unfolding actions, or changes from the beginning to the end of the video.
- **Spatiality:** Describe the spatial layout of the scene. Explain how key elements are positioned and oriented, and how they relate to one another within the frame.
- **Object:** Identify key objects in the video, including inanimate items and humans. Describe their appearance, clothing, physical traits, materials, and possible roles.
- **Action:** Describe the main actions taking place. Specify what actions occur, who or what performs them, and the manner or sequence in which they unfold.
- **Scene:** Provide a high-level overview of the setting. Describe the environment, background, and general activity or context presented in the video.
- **Summary:** Offer a brief yet comprehensive summary that contains the core event or purpose of the given video clip.

Output Format (JSON):

Your response should be formatted as a JSON object, where each key corresponds to a topic and each value is the description associated with that topic.

Example Output:

```
```json {
 "Temporality": "The video begins with the group standing idle under the tree and progresses to active conversation and movement, suggesting preparation for an activity.",
 "Spatiality": "The individuals are grouped under a large tree, spaced out in a semicircle. The surrounding area is an open, grassy park with scattered trees in the background.",
 "Object": "The scene includes several casually dressed men wearing t-shirts, shorts, hats, and sunglasses. One man carries a medium-sized cardboard box. The background features natural objects like trees, grass, and park benches.",
 "Action": "The men engage in conversation, gesturing with their hands, and preparing for an activity. One man walks away, possibly to retrieve additional items.",
 "Scene": "The video captures a casual outdoor setting in a park. A small group of men gather under a shady tree, seemingly preparing for a social or recreational activity in a relaxed, natural environment.",
 "Summary": "A group of men gather under a tree in a public park, casually conversing and preparing for an activity. The video captures a moment of calm interaction and coordination in a relaxed outdoor setting."
}
```

Figure 5: The prompt for clip-level video captioning.

## Event Splitting

You will be given a list of video clips, where each clip is defined by a start time, end time, and its video content. Your task is to merge related clips into a small number of coherent events, each represented by a merged time span and an event title.

### ### Instructions:

Consider the context for each video clip and ignore some outliers or unreasonable video content. If the given video clips are already an event, do not merge.

- **Merge Related Clips:** Combine consecutive clips that logically belong to the same event based on content continuity.
- **Preserve Context:** Each merged event should represent a self-contained and contextually consistent unit of action or activity.
- **Filter Outliers:** Ignore clips that are irrelevant, inconsistent with the surrounding context, or clearly out of place.
- **Balance Granularity:** Avoid over-segmenting the video. Aim to minimize the number of events while ensuring that each one captures a distinct scene, set of actions, or objects.
- **Respect Standalone Clips:** If a single clip already represents a complete, meaningful event, retain it without merging.

### ### Output Format (JSON):

Return a JSON object where:

- Each key is a string denoting the merged time span of an event (*e.g.*, "0-40s").
- Each value is a concise event title that summarizes the main content or activity.

### #### Example Output:

```
```json {
  "0-40s": "Introduction to Holiday Desserts",
  "60-110s": "Preparing Ingredients and Tools",
  "120-190s": "Mixing Ingredients",
  "200-290s": "Placing Ingredients into Tools",
  "300-350s": "Baking of Desserts",
  "360-410s": "Presentation and Enjoyment of Desserts",
}
```

Figure 6: The prompt for event splitting.

Event-level Video Captioning

You will be provided with a sequence of video clips depicting a specific event. Each clip includes metadata (start time, end time) and structured descriptions from six perspectives. Your task is to generate a cohesive, natural-language narrative that summarizes the entire event in chronological order. To support your understanding of the broader context, you will also receive brief descriptions of the events immediately before and after the current one. Use this surrounding context to enhance the flow and interpretation, but do not include those descriptions in your output.

Instructions:

1. Maintain Chronological Flow

- The clip-level video captions are **already chronologically ordered**. Your summary must **preserve this timeline** and describe the event as a continuous progression.
- **Avoid explicitly referencing individual clips** (e.g., "*The first clip shows...*" or "*As the clip progresses...*"). Instead, describe the event as a seamless and continuous narrative.
- **Do not assume the first or last frame of any clip represents the beginning or end of the entire event.** Instead, focus on how the event unfolds as a whole.

2. Use Adjacent Event Context Thoughtfully

- To **improve coherence and contextual understanding**, you will receive **brief descriptions of the events that occur immediately before and after the current event**.
- Use this information to **better understand** the current event.
- **IMPORTANT:** Do not include these descriptions in your summary. They are provided solely to help you maintain context and continuity.

3. Preserve Key Details, Filter Outliers

- Retain **all relevant details** from the clip descriptions to ensure accuracy and completeness.
- **Based on the brief description of the current event**, ignore **outliers** or clips with inconsistent, irrelevant, or contradictory content that do not contribute meaningfully to the event-level narrative.

4. Write in a Natural, Engaging Tone

- Your summary should **read like a natural video description**, as if you are directly describing the event rather than summarizing segmented clips.
- **Avoid mechanical phrases** often found in clip descriptions, such as "*The clip begins...*", "*As the clip progresses...*", "*The clip concludes...*", "*The first/last frame of this clip...*", "*The second clip shows...*".
- Instead, **focus on crafting a flowing narrative**. The goal is to help a reader visualize the full event as if watching it unfold.

Output Format (JSON):

The output should be structured as a JSON object.

Example Output:

```
```json {
 "Event-Level Description": "YOUR DESCRIPTION HERE."
} ...
```

### ### Input:

- Brief description of current event: **<EVENT TITLE>**
- The event before the current event: **<PREV EVENT DESCRIPTION>**
- The event after the current event: **<NEXT EVENT DESCRIPTION>**
- Detailed descriptions of all clips in the current event:
  - **<CLIP-LEVEL DESCRIPTION 1>**
  - **<CLIP-LEVEL DESCRIPTION 2>**
  - **...**
  - **<CLIP-LEVEL DESCRIPTION N>**

Figure 7: The prompt for event-level video captioning.

## Global-level Video Captioning

You will be provided with a chronologically ordered sequence of video events, each accompanied by both a brief and a detailed description. Your task is to synthesize these descriptions into a single, cohesive, and vivid narrative that summarizes the entire sequence of events as a unified whole, without breaking the flow into separate segments or referencing individual clips mechanically.

### ### Instructions:

#### #### 1. Event-level Video Descriptions

Each event is described with two levels of granularity:

- **Brief Description:** A concise overview of the event.
- **Detailed Description:** A more in-depth description, including finer details.

#### #### 2. Generate a Unified Narrative

- Write a **single, flowing narrative** that describes the full sequence of events from beginning to end.
- Maintain **chronological order** without explicitly mentioning timestamps or referencing individual events (e.g., "*In the second event...*").
- Ensure the narrative reads as if you are describing a continuous experience, **not a series of separate parts**.

#### #### 3. Focus on Relevance and Consistency

- Incorporate all **meaningful and consistent details** across the event descriptions.
- If a detail appears **inconsistent, irrelevant, or clearly unrelated**, omit it based on your understanding of the overall narrative.

#### #### 4. Use a Natural and Engaging Tone

- Write in a **fluent, descriptive style**, suitable for someone reading or hearing a natural summary of the video.
- Avoid mechanical phrases like: "*The event begins...*", "*As the event progresses...*", "*The first/last event shows...*".
- Instead, **immerse the reader** in the experience, emphasizing continuity, clarity, and engagement.

### ### Output Format (JSON):

The output should be structured as a JSON object.

#### #### Example Output:

```
```json {
  "Global-Level Description": "YOUR DESCRIPTION HERE."
}
```

Input:

Event from <START TIME 1> - <END TIME 1>:

- Brief description: <EVENT TITLE 1>
- Detailed description: <EVENT-LEVEL DESCRIPTION 1>

Event from <START TIME 2> - <END TIME 2>:

- Brief description: <EVENT TITLE 2>
- Detailed description: <EVENT-LEVEL DESCRIPTION 2>

...

Event from <START TIME N> - <END TIME N>:

- Brief description: <EVENT TITLE N>
- Detailed description: <EVENT-LEVEL DESCRIPTION N>

Figure 8: The prompt for global-level video captioning.

Open-Ended QA Generation

You are an intelligent assistant specializing in **open-ended question-answer generation** for video understanding. Please follow the instructions precisely and **use only the information provided** in the input. Do **not** introduce any content unrelated to the described video clips.

You will be provided with:

- A **chronologically ordered sequence** of **<TOPIC>**-based video clip descriptions (each 30 seconds long, with start and end timestamps).
- A **global-level description** summarizing the overall content of the video.

Your task is to generate **open-ended Question-Answer (QA) pairs** from the perspective of **<TOPIC>**. These QA pairs should promote deep understanding and must be **strictly grounded** in the given descriptions. **No hallucination or fabrication is allowed.**

Instructions:

1. <TOPIC>-Based Sub-Tasks

The **<TOPIC>**-based sub-tasks can be categorized into the following sub-tasks:

- **<SUB-TASK 1>**: **<TASK DESCRIPTION 1>**
- **<SUB-TASK 2>**: **<TASK DESCRIPTION 2>**
- (...additional sub-tasks as needed)

2. QA Examples for Each Sub-Task

To guide your generation, here are example QA pairs for each sub-task:

Examples of **<SUB-TASK 1>**:

```
[
  { "question": <DEMO Q1-1>, "answer": <DEMO A1-1> },
  { "question": <DEMO Q1-2>, "answer": <DEMO A1-2> },
  ...
]
```

Examples of **<SUB-TASK 2>**:

```
[
  { "question": <DEMO Q2-1>, "answer": <DEMO A2-1> },
  { "question": <DEMO Q2-2>, "answer": <DEMO A2-2> },
  ...
]
...
```

3. Guidelines for Question-Answer Generation:

- **Focus on <TOPIC>-Relevant Information:** Carefully analyze the descriptions to identify patterns relevant to the **<TOPIC>**.
- **Relevance and Context:** The questions and answers must align with the content and context of the video clips. The generated question-answer pairs should not introduce information that is not present in the description.
- **Balance Diversity and Clarity:** Create a variety of questions that collectively capture a full understanding of the topic.
- **Quantity:** Generate exactly **three** QA pairs for each sub-task.

Output Format (JSON):

The output should be structured as a JSON object.

Example Output:

```
```json
{
 "<Sub-Task 1>": [{"question": "<Question 1-1>", "answer": "<Answer 1-1>"}, ...],
 "<Sub-Task 2>": [{"question": "<Question 2-1>", "answer": "<Answer 2-1>"}, ...],
 ...
}
```

### ### Input:

- **<TOPIC>**-based Descriptions:
  - Clip from **<START TIME 1>** - **<END TIME 1>**: **<TOPIC-SPECIFIC CLIP DESCRIPTION 1>**
  - Clip from **<START TIME 2>** - **<END TIME 2>**: **<TOPIC-SPECIFIC CLIP DESCRIPTION 2>**
  - ...
- Overall Description: **<GLOBAL-LEVEL DESCRIPTION>**

Figure 9: The prompt for open-ended (OE) question-answer generation.



## Multiple-Choice QA Generation

You are an intelligent assistant specializing in **multi-choice question-answer generation** for video understanding. Please follow the instructions precisely and **use only the information provided** in the input. Do **not** introduce any content unrelated to the described video clips.

You will be provided with:

- A **chronologically ordered sequence** of **<TOPIC>**-based video clip descriptions (each 30 seconds long, with start and end timestamps).
- A **global-level description** summarizing the overall content of the video.

Your task is to generate **multiple-choice Question-Option-Answer triplets** from the perspective of **<TOPIC>**. These triplets should promote deep understanding and must be **strictly grounded** in the given descriptions. **No hallucination or fabrication is allowed.**

### ### Instructions:

#### #### 1. <TOPIC>-Based Sub-Tasks

The **<TOPIC>**-based sub-tasks can be categorized into the following sub-tasks:

- **<SUB-TASK 1>**: **<TASK DESCRIPTION 1>**
- **<SUB-TASK 2>**: **<TASK DESCRIPTION 2>**
- (...additional sub-tasks as needed)

#### #### 2. QA Examples for Each Sub-Task

To guide your generation, here are example QA pairs for each sub-task:

Examples of **<SUB-TASK 1>**:

```
[
 { "question": <DEMO Q1-1>, "options": <DEMO OP1-1>, "answer": <DEMO A1-1> },
 { "question": <DEMO Q1-2>, "options": <DEMO OP1-2>, "answer": <DEMO A1-2> },
 ...
]
```

Examples of **<SUB-TASK 2>**:

```
[
 { "question": <DEMO Q2-1>, "options": <DEMO OP2-1>, "answer": <DEMO A2-1> },
 { "question": <DEMO Q2-2>, "options": <DEMO OP2-2>, "answer": <DEMO A2-2> },
 ...
]
```

#### #### 3. Guidelines for Question-Answer Generation:

- **Focus on <TOPIC>-Relevant Information:** Carefully analyze the descriptions to identify patterns relevant to the **<TOPIC>**.
- **Relevance and Context:** The questions, options, and answers must align with the content and context of the video clips. The generated question-option-answer triplets should not introduce information that is not present in the description.
- **Balance Diversity and Clarity:** Create a variety of questions that collectively capture a full understanding of the topic.
- **Quantity:** Generate exactly **three** QA pairs for each sub-task.

### ### Output Format (JSON):

The output should be structured as a JSON object.

#### #### Example Output:

```
```json
{
  "<Sub-Task 1>": [
    { "question": "<Question 1-1>", "options": [...], "answer": "<Answer 1-1>" },
    { "question": "<Question 1-2>", "options": [...], "answer": "<Answer 1-2>" },
    ...
  ],
  "<Sub-Task 2>": [
    { "question": "<Question 2-1>", "options": [...], "answer": "<Answer 2-1>" },
    { "question": "<Question 2-2>", "options": [...], "answer": "<Answer 2-2>" },
    ...
  ],
  ...
}
```

Input:

- **<TOPIC>**-based Descriptions:
 - Clip from **<START TIME 1>** - **<END TIME 1>**: **<TOPIC-SPECIFIC CLIP DESCRIPTION 1>**
 - Clip from **<START TIME 2>** - **<END TIME 2>**: **<TOPIC-SPECIFIC CLIP DESCRIPTION 2>**
 - ...
- Overall Description: **<GLOBAL-LEVEL DESCRIPTION>**

Figure 10: The prompt for multiple-choice (MC) question-answer generation.

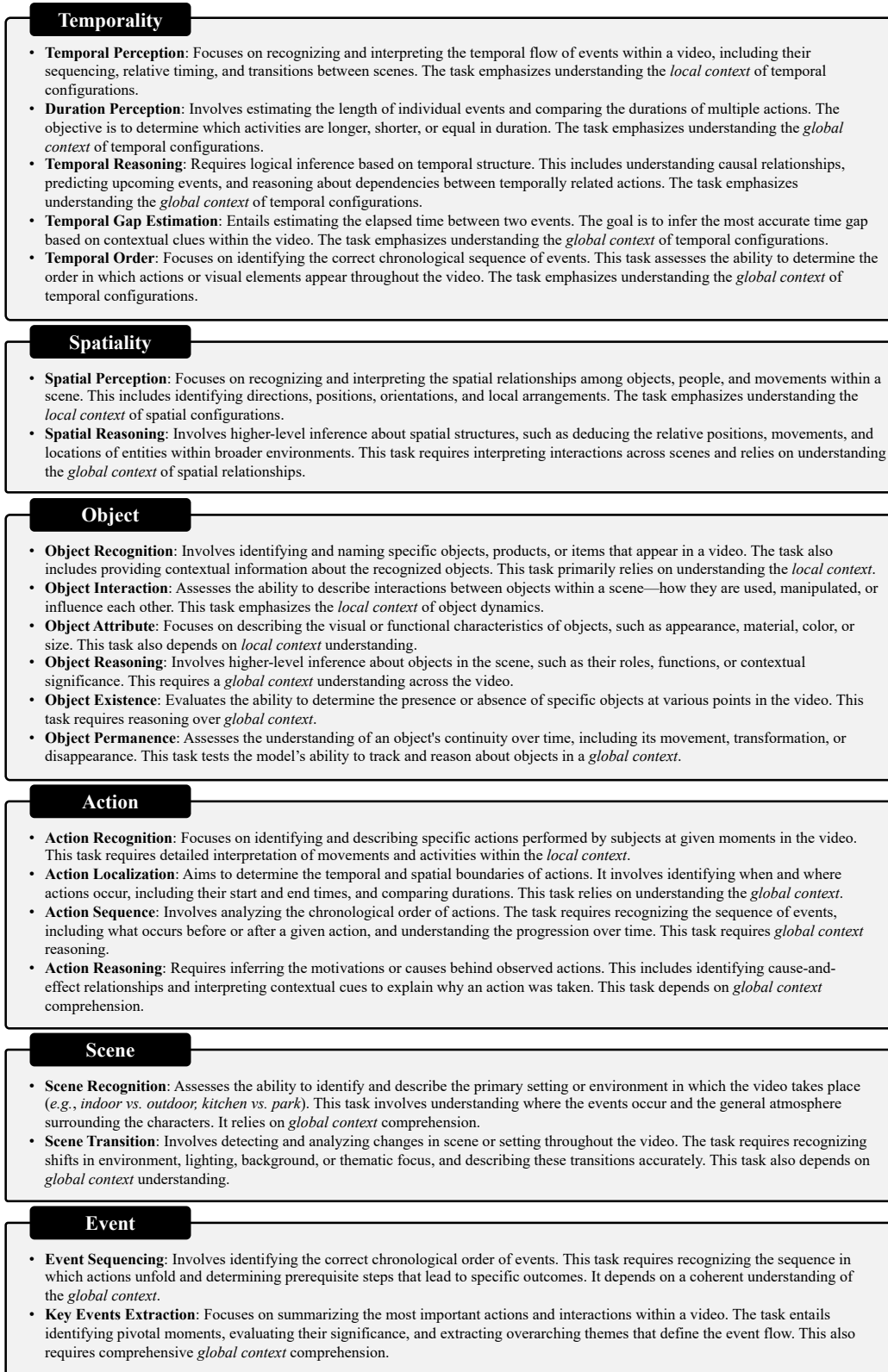


Figure 11: Detailed descriptions of 22 sub-tasks over six fundamental topics (except “overall summarization” task). For the “overall summarization” task, we directly use global-level video descriptions as the corresponding answers, so there is no need to design an additional prompt.



```
"20-50s": {
  "Temporality": "The video progresses from the woman introducing the recipe to showing the ingredients and possibly demonstrating the steps involved.",
  "Spatiality": "The woman is seated in front of the table, which is positioned against a plain wall. The ingredients are arranged neatly on the table.",
  "Object": "The woman wears a black top and a choker necklace with a pendant. On the table, there are three bowls containing sugar, flour, and butter, an egg, and a glass bowl.",
  "Action": "The woman speaks to the camera, gestures with her hands, and points to the ingredients on the table. She appears to be explaining the recipe.",
  "Scene": "The video features a woman sitting in a chair, discussing a recipe while showing various ingredients laid out on a table.",
  "Summary": "This video showcases a woman presenting a homemade chocolate salted caramel tart recipe. She explains the ingredients and their arrangement on the table, providing a step-by-step guide for viewers to follow along."
}
```

(a) Examples of clip-level video captioning.

```
"0-70s": "Introduction and Overview of Chocolate Salted Caramel Tart Recipe",
"60-130s": "Preparing and Kneading the Dough",
"120-190s": "Rolling and Shaping the Tart Crust",
"180-270s": "Creating the Caramel Sauce and Assembling the Tart",
"280-350s": "Finalizing and Presenting the Chocolate Salted Caramel Tart"
```

(b) Examples of event splitting

```
"120-190s": {
  "Brief Description": "Rolling and Shaping the Tart Crust",
  "Detailed Description": "In a cozy kitchen setting, a person meticulously prepares a tart crust. The process begins with a ball of dough on a wooden surface, surrounded by a generous sprinkling of flour. The individual, dressed in a dark-colored shirt, kneads the dough by hand, ensuring it is well-mixed and pliable. Once the dough is ready, they use a traditional rolling pin to flatten it, moving the pin back and forth to achieve an even thickness. The rolling pin's rhythmic motion is a testament to the baker's skill and experience. As the dough is rolled out, it is carefully transferred to a round tart pan, which is lined with parchment paper to prevent sticking. The person then shapes the dough into the pan, pressing it gently into the sides and bottom to create a smooth and even crust. Excess dough is trimmed away, and the final touches are added to ensure the tart base is perfect. The entire process is carried out with a focused and meticulous approach, highlighting the importance of precision in baking. The scene is a blend of traditional techniques and modern kitchen tools, creating a harmonious and engaging visual of the tart crust preparation."
}
```

(c) Examples of event-level video captioning.

The video begins with a warm and inviting introduction to a homemade chocolate salted caramel tart recipe. A woman, dressed in a black top and wearing a choker necklace with a pendant, sits in front of a table adorned with colorful candies spelling out 'THE SWEETEST THINGS.' She speaks directly to the camera, gesturing with her hands in a friendly and engaging manner. The scene transitions to her discussing the recipe, pointing out the ingredients laid out on the table: sugar, flour, butter, an egg, and a glass bowl. The setting appears to be a kitchen or dining area, bathed in natural daylight, creating a cozy atmosphere. The woman stands in front of a wooden table, where she begins to mix the ingredients for the tart. Using a fork, she combines the flour, butter, sugar, and egg methodically, demonstrating each step clearly and encouraging viewers to follow along. As the video progresses, the focus shifts to the preparation and kneading of the dough. The woman, now in a slightly messy kitchen environment, works on a wooden table set with all the necessary ingredients. She starts by mixing the flour and butter with a fork, ensuring a thorough blend. After adding an egg and a small amount of water, she continues to mix the dough until it reaches the right consistency. Her hands, now covered in flour, demonstrate a practiced technique as she kneads the dough on the wooden surface, shaping it into a ball. The scene highlights her attention to cleanliness and precision, setting the stage for the next step in the recipe.

The process of rolling and shaping the tart crust follows. The woman, dressed in a dark-colored shirt, works with the dough on a floured wooden surface. She uses a traditional rolling pin to flatten the dough, moving it back and forth to achieve an even thickness. The dough is then carefully transferred to a round tart pan lined with parchment paper. She presses the dough gently into the sides and bottom of the pan, ensuring a smooth and even crust. Excess dough is trimmed away, and the final touches are added to perfect the tart base. The scene showcases a blend of traditional techniques and modern kitchen tools, emphasizing the importance of precision in baking.

Next, the video transitions to the creation of the caramel sauce and the assembly of the tart. The woman places parchment paper into the tart pan and adds rice to weigh it down, ensuring an even distribution of weight. After removing the parchment paper, she applies an egg wash to the pre-baked tart shell. She then prepares the caramel sauce by heating sugar and water in a pot, stirring the mixture with a wooden spoon. Cream and butter are added to create a rich, smooth caramel sauce, which is spread evenly over the tart shell. Dark chocolate chunks are added for flavor and texture, setting the stage for the final steps in the recipe.

The video concludes with the finalization and presentation of the chocolate salted caramel tart. A close-up shows a bowl of melted chocolate being poured into the tart shell, which is placed on a black slate board. The chocolate is spread evenly using a spoon, and the tart is then moved to a wooden cutting board for presentation. Sea salt is sprinkled on top, adding a touch of contrast and enhancing the tart's flavor. The tart is cut into triangular slices with a steady and deliberate technique, ready to be served. The entire process highlights the simplicity and elegance of the chocolate salted caramel tart, from its preparation to its final presentation, leaving viewers inspired to try the recipe themselves.

(d) Examples of global-level video captioning.

Figure 12: Examples of hierarchical video captioning.

References

- [1] Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, TaoZhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video KV-cache retrieval. In *ICLR*, 2025.
- [2] Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, et al. Token-efficient long video understanding for multimodal llms. *arXiv:2503.04130*, 2025.
- [3] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
- [4] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv:2410.17434*, 2024.
- [5] Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, Bryan A Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. In *CVPR*, 2024.
- [6] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv:2407.15841*, 2024.
- [7] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.
- [8] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv:2410.02713*, 2024.