# What Happens When: Learning Temporal Orders of Events in Videos

## Supplementary Material

**Note**: We use blue color to refer to figures, tables, section numbers, and citations **in the main paper** (*e.g.*, [10]). We use steel blue color to refer to figures, tables, section numbers, and citations **in this supplementary material**.

## 1. Further Discussion on the Impact of Shuffled Video Frames in Previous Video Understanding Benchmarks

The ability to comprehend event sequences in videos is crucial for VLMMs to achieve human-like understanding of real-world visual scenarios. Recent benchmarks evaluate the temporal understanding of VLMMs through question-answering tasks [11, 17, 18, 19, 24, 29, 38, 44], primarily focusing on temporal relationships and causal reasoning.

However, our preliminary experiments reveal that state-of-the-art (SoTA) VLMMs [15] demonstrate notably strong performance even when video frames are randomly shuffled (Fig.1-(a)). This indicates that existing benchmarks often enable correct answers without the necessity of genuine temporal comprehension. In particular, as illustrated in Fig.1, both 7B and 72B variants of LLaVA-One-Vision [15] exhibit similar robustness when temporal frame order is disrupted, as supported by Fig.1. Specifically, Tab.1 shows that across nine benchmarks, both model scales retain over 82% of their original accuracy ($\rho$) even with temporally shuffled input. This consistency across model scales further supports the observation that current benchmarks inadequately assess genuine temporal understanding capabilities. We hypothesize that models primarily rely on their *prior knowledge* of common event scenarios, inferring plausible contexts from isolated frames rather than explicitly analyzing temporal relationships depicted in the videos [7, 47]. Consequently, VLMMs may bypass true temporal reasoning by using common-sense shortcuts.

## 2. Comprehensive Comparison with Existing Video Understanding Benchmarks

We compare detailed properties of existing benchmarks and highlight differences with VECTOR. Table 2 provides a statistical comparison of our proposed VECTOR benchmark against various existing video understanding benchmarks. Unlike most prior benchmarks, VECTOR explicitly evaluates temporal order understanding at both event and pattern levels. It includes 31,200 videos covering 4,800 questions, each video containing 4–9 clearly defined events. The videos have an average duration of 64 seconds, ranging from 4 to 100 seconds, providing concise yet temporally structured sequences. In contrast, previous benchmarks primarily rely on multiple-choice (MCQ) or binary answers and tend to evaluate only event-level comprehension. VECTOR uniquely integrates both event and higher-level temporal reasoning tasks, specifically targeting temporal order understanding, thus offering a comprehensive diagnostic tool for video multimodal models.

Detailed descriptions of each general video benchmark and further comparisons to the proposed VECTOR benchmark are provided below.

**TempCompass.** TempCompass [24] evaluates temporal perception in Video Large Multimodal Models (VLMMs) across aspects such as action, speed, direction, attribute change, and event order. It introduces conflicting video pairs—where static content remains unchanged while temporal aspects vary—to mitigate single-frame bias and reliance on language priors. TempCompass also expands evaluation formats beyond multiple-choice QA, incorporating caption matching and generation to assess models' generalization across different response styles. Although TempCompass covers diverse temporal phenomena, it is focused on short videos that contains less than two events. Also, many of its questions could still be answered by analyzing individual or pairwise frames rather than reasoning over multiple successive events.

**MLVU.** MLVU [57] evaluates long video understanding across various tasks and genres. One of MLVU's key tasks involves embedding short 'probe' events into a lengthy background video, necessitating that models first locate these target clips amid large amounts of noisy frames, and then interpret their order. This setup does incorporate an element of temporal reasoning; however, it often centers on searching through excessive context to spot the relevant information, rather than systematically examining the temporal sequence of every event in the video. In contrast, VECTOR focuses specifically on multi-event *ordering*, using intentionally concatenated short clips to create a series of distinct events. Furthermore, VECTOR not only evaluates models on full-sequence ordering but also introduces various tasks such as sub-sequence ordering with detailed metrics over exact match that improves understanding of model's failure cases, enabling more comprehensive assessment for global temporal understanding.

**TemporalBench.** TemporalBench [4] is a benchmark for evaluating fine-grained temporal understanding in videos. It contatins approximately 10K QA pairs derived from human annotations, covering temporal reasoning skilss such as action frequency, motion magnitude, and event order. The benchmark supports both video QA and captioning across short and long videos, thereby offering a comprehensive testbed for assessing multimodal video models. The key difference between TemporalBench and our VECTOR lies in their evaluation scope. TemporalBench primarily measures whether models capture temporal dynamics across entire individual video clips, relying on a binary caption selection and a single, generic caption generation ("Please generate a caption for the following video"). In contrast, VECTOR introduces a multi-layered evaluation framework with diverse tasks explicitly designed to assess a model's understanding of temporal relationships among multiple events and patterns.

**EgoSchema.** EgoSchema [29] is a diagnostic benchmark designed to assess long-form video-language understanding in

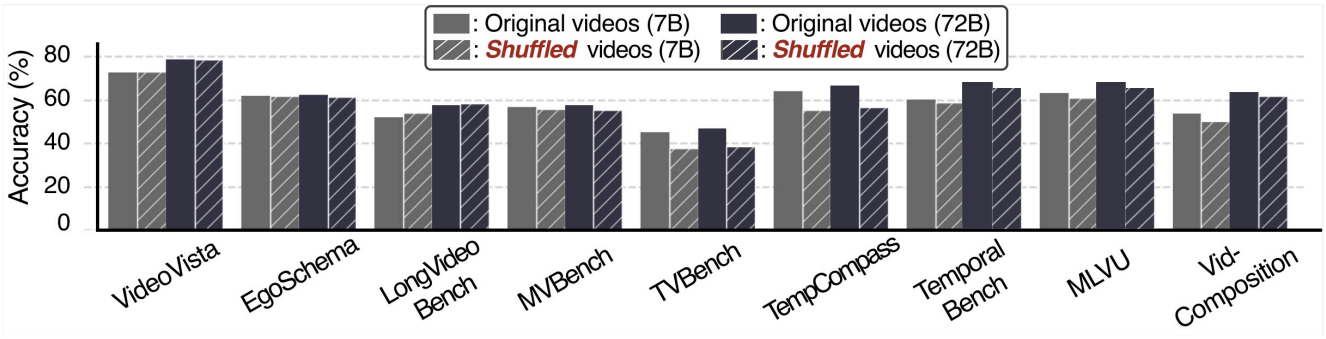



Figure 1. **Accuracy of VLMMs on original vs. frame-shuffled videos across different benchmarks.** We evaluate the accuracy of LLaVA-OV 7B and 72B [15] across nine video understanding benchmarks. We observe that there's no significant performance difference between evaluations using original videos and frame-shuffled videos across all benchmarks.

| Models | $\rho$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | MVBench | EgoSchema | LongVideoBench | TempCompass | TVBench | VideoVista | TemporalBench | MLVU |
| LLaVA-OV (7B) | 98.59 | 99.36 | 104.4 | 86.35 | 83.68 | 100.1 | 97.25 | 96.56 |
| LLaVA-OV (72B) | 96.01 | 98.77 | 101.4 | 85.57 | 82.71 | 100.1 | 96.22 | 95.58 |

Table 1. **Performance ratio ($\rho$) of VLMMs: Comparison of model performance on original *vs*. frame-shuffled videos across different benchmarks.** We measure performance ratio ($\rho$) of LLaVA-OV 7B and 72B on seven video understanding benchmarks. $\rho$ represents the ratio of benchmark performance on frame-shuffled videos compared to their performance on original videos. We observe no significant performance difference between evaluations using original videos and frame-shuffled videos across all benchmarks.

VLMMs. It introduces the notion of temporal certificate length, quantifying the intrinsic temporal difficulty of video understanding tasks. It consists of over 5000 multiple-choice QAs curated to require reasoning over long, diverse egocentric videos, with questions demanding extended temporal comprehension beyond short-term cues. While EgoSchema focuses on long-form video comprehension and memory, often posing questions about events that are inherently difficult to segment and thus more easily answered using prior knowledge, VECTOR is specifically designed to prevent reliance on such priors, ensuring a more effective evaluation of temporal understanding.

**MVBench.** MVBench [17] evaluates temporal perception and reasoning in video large multimodal models (VLMMs) by adapting static image tasks into 20 video-based tasks (*e.g.*, action sequence, moving direction). It covers action recognition, object tracking, scene transitions, and reasoning, requiring models to process temporal changes rather than single frames. Using 11 public datasets, MVBench automatically generates multiple-choice QAs with LLMs, ensuring broad and diverse evaluation. While MVBench assesses general video understanding, VECTOR focuses on event sequencing and temporal order comprehension. Unlike multiple-choice QAs, VECTOR requires structured event predictions, making it more resistant to shortcuts and prior-knowledge reliance.

**VideoVista.** VideoVista [19] is a large-scale, versatile benchmark designed to comprehensively evaluate video understanding and reasoning capabilities of VLMMs. It consists of 25,000 questions spanning multiple categories, covering both short and long videos (ranging from a few seconds to over 10 minutes). VideoVista includes various tasks related to temporal reasoning, such as event sequence prediction and action localization. However, its primary focus is on diverse aspects of video comprehension rather than specifically evaluating the fine-grained understanding of multi-event temporal order. In contrast, VECTOR specifically targets a model's ability to comprehend and reason about event sequences within videos.

**TVBench.** TVBench [11] is a multiple-choice video-language benchmark designed to evaluate the temporal understanding capabilities of VLMMs. It focuses on assessing event sequencing, temporal localization, and the ability to distinguish fine-grained temporal relationships in video data. Unlike many existing video QA benchmarks that contain spatial and textual biases, TVBench ensures that solving the tasks requires genuine temporal understanding by carefully designing questions and answer choices. The benchmark includes a diverse set of tasks, such as action sequencing, object movement tracking, and scene transitions, sourced from various real-world and synthetic datasets. Compared to TVBench, which primarily serves as a diagnostic benchmark for analyzing the limitations of current VLMMs in handling temporal dependencies, VECTOR introduces a novel dataset specifically designed to evaluate temporal order comprehension.

**LongVideoBench.** LongVideoBench [44] is a benchmark designed to evaluate long-context multimodal understanding in video-language models. It assesses whether models can retrieve and reason about specific details within hour-long videos using referring reasoning tasks, which require models to locate rele-

| Benchmark | Answer Type | #Events | #Videos | Duration (Avg./Max) | #QA | Difficulty Level | Pattern Level | Main Goal |
|---|---|---|---|---|---|---|---|---|
| VideoVista | MCQ | 2-10 | 3,402 | 131s / 919s | 25,000 | O | X | long video understanding |
| EgoSchema | MCQ | - | 5,031 | 180s / 180s | 5,031 | X | X | long egocentric video understanding |
| LongVideoBench | MCQ | - | 3,763 | 477s / 60m | 6,678 | O | X | long video understanding |
| MVBench | MCQ | - | 3,655 | 18s / 176s | 4,000 | O | X | comprehensive video understanding |
| TVBench | MCQ | 2-13 | 2,217 | 21s / 116s | 2,525 | X | X | video temporal understanding |
| TempCompass | MCQ/Open | 1-2 | 410 | 11s / 35s | 7,540 | X | X | short video temporal understanding |
| TemporalBench | Binary | 4-8 | 3,753 | 46s / 20m | 9,867 | O | X | video caption matching |
| MLVU | MCQ/Open | 4 | 1,112 | 755s / 133m | 2,174 | X | X | long video understanding |
| VidComposition | MCQ | 3-8 | 982 | 26s / 139s | 1,706 | X | X | compiled video understanding |
| VECTOR | List/MCQ | 4-9 | 31,200 | 64s / 100s | 4,800 | O | O | temporal order understanding |

Table 2. **Statistics of VECTOR compared with previous benchmarks.** Unlike other benchmarks, VECTOR considers pattern-level order. For comprehensive benchmarks, we count # Events for the benchmarks with event ordering task.

vant moments and answer complex multimodal questions. Unlike LongVideoBench, which focuses on long video comprehension and multimodal reasoning, VECTOR is specifically designed to assess event sequencing and temporal order understanding in shorter multi-event videos. VECTOR evaluates whether models can correctly order events based on temporal cues, while LongVideoBench includes some temporal reasoning tasks that can often be solved using prior knowledge rather than requiring models to derive the actual event sequence from visual evidence.

**VidComposition.** VidComposition [38] is a benchmark evaluating fine-grained video composition understanding of Multimodal Large Language Models (MLLMs). It comprises 982 videos and 1,706 multiple-choice questions, emphasizing compositional elements like camera movement, angles, shot size, narrative structure, character actions, and emotions. Compared to other general video benchmarks, VidComposition focuses on nuanced interpretation of complex visual contexts. Evaluations of 33 MLLMs highlight substantial gaps compared to human performance, revealing areas for improvement.

Unlike VidComposition, which assesses visual composition and narrative interpretation, VECTOR specifically targets models' ability to explicitly reason about temporal order across discrete video events. While VidComposition emphasizes cinematographic and emotional context, VECTOR uses controlled synthetic videos to deliberately disrupt common-sense priors, explicitly evaluating temporal reasoning independent of prior knowledge.

# 3. Experimental Details For Diagnosing the Prior-Knowledge Shortcut

To investigate whether VLMMs rely more on prior knowledge than true temporal understanding, we conduct an empirical study using two densely captioned video datasets: MECD [7] and HiREST [49]. Both datasets provide fine-grained captions aligned with specific video segments, enabling precise temporal analysis. We preprocess these datasets to construct an evaluation set that assesses whether models can accurately infer event order based on visual content rather than relying on common-sense priors.

To construct the evaluation set, we select non-overlapping captions from both datasets to ensure that each caption represents a distinct event. To minimize the possibility of models inferring event order without relying on visual content, we refine the captions using GPT-4o. Specifically, we eliminate references such as "that man" or "then", which could implicitly encode temporal relationships between events.

Using this refined dataset, we conduct an evaluation where models are presented with a set of event descriptions labeled with option letters (A, B, C, D) and tasked with reordering them into their correct chronological sequence based on the video input. The dataset consists of 534 samples from MECD and 296 samples from HiREST, where each sample contains multiple captions describing different events in the video. Each sample includes 3, 4, 5, or 6 distinct events.

# 4. Building VECTOR Benchmark

## 4.1. Data Construction for Understanding Temporal Order of Events

We constructed VECTOR rule-based to allow the scalable evaluation. For event-level temporal order understanding tasks, we ensure that each multi-event video $V$ comprises distinct events. We first sample subset of event categories $C^e$ which is feed to the model to list the recognized events from all 700 Kinetics action classes [37]. We then randomly sample $N_e$ distinct events and their corresponding videos from the validation split of Kinetics-700 dataset. We made sure that all $N_e$ event types are distinct to avoid ordering confusion. We define two difficulty levels by varying the sequence length $N_e$, $N_e = 4$ for L1 and $N_e = 8$ for L2. Resulting event sequencing task consists of 3k questions across three tasks, with each task evaluated at two task difficulty levels (L1 and L2):

- Event sequencing: 0.6k questions (0.3k per difficulty level), using 2.5k videos.
- Relative event sequencing: 0.6k questions (0.3k per difficulty level), using 2.5k videos.
- Event position identification: 1.8k questions (0.6k for each single, double, and triple event identification), using 8.1k videos.

### 4.2. Data Construction for Understanding Temporal Order of Patterns

For pattern-level temporal order understanding tasks, we carefully construct multi-event videos $V$ by selecting events that align with specific task requirements. We sample events from our semantic groups $\mathcal{G}$, which is a set of semantic superclasses of Kinetics action classes [37], using videos from the validation split of Kinetics-700 dataset. The dataset consists of 1.8k questions across two reasoning tasks:

- Discordant semantic-group position identification: 0.6k questions (0.3k per difficulty level), using 3.1k videos.
- Discordant event position identification: 1.2k questions (0.3k for each of four event patterns), using 8.6k videos.

**Construction for semantic group $\mathcal{G}$.** We organize the 700 action classes from the Kinetics dataset into semantic groups $\mathcal{G}$ through an iterative refinement process involving human-LLM collaboration (Claude 3.5 [2]). Initially, Claude 3.5 generated 50 preliminary groups, followed by three rounds of refinement: (1) expert review by a vision-language researcher, (2) reorganization based on expert judgment, and (3) LLM validation and refinement of the revised groupings.

Rare cases that do not fit into clear categories are consolidated into a 'Miscellaneous Activities' group, which is excluded from our data construction to maintain clarity in group separation. Through this three-iteration refinement, we categorize the 700 classes into 50 distinct semantic groups. Figure 3 visualizes the top 20 groups for clarity, with 'Others' representing the aggregate of the remaining 30. The values in parentheses indicate the number of video instances within each group.

**Discordant semantic-group position identification.** We randomly choose two semantic groups from $\mathcal{G}$: a dominant group $g^d$ and a discordant group. For the dominant group $g^d$, we sample $N_e - 1$ different events and their corresponding videos from Kinetics validation split dataset. We then sample one event from the discordant group and position it at the $k$-th position in the sequence of events, as illustrated in the Task 4 of Fig. 5. The VLMM's task is to identify the position $k$ where the event's semantic group $g_{e_k}$ differs from the dominant group $g^d$, while all other positions contain events that belong to the dominant group ($g_{e_i} = g^d$ for $i \neq k$). For this discordant semantic-group position identification task, we collected 300 multi-event videos with corresponding question-answer pairs.

**Discordant event position identification.** For discordant event position identification, we first define four types of event patterns: $s_1 s_2 s_1 s_2 s_1 s_2$, $s_1 s_2 s_1 s_2 s_1 s_2 s_1 s_2$, $s_1 s_2 s_3 s_1 s_2 s_3$, and $s_1 s_2 s_3 s_1 s_2 s_3 s_1 s_2 s_3$, where each $s_i$ represents events from different classes. For each pattern, the task is to identify a position where a randomly injected discordant event $x$ ($x \neq s_i$) disrupts the repeating pattern. We collected 300 multi-event videos with corresponding question-answer pairs for each pattern type.

### 4.3. Detailed Evaluation Prompts for Each Task

Figures 4 and 5 illustrate the complete set of task prompts from our VECTOR benchmark. Each task is accompanied by answer prompts to guide VLMMs in generating appropriately formatted responses.

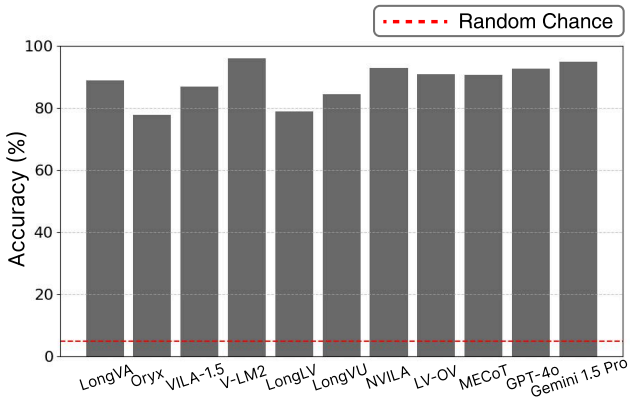



Figure 2. **Single-event recognition accuracy on various VLMMs**. We conduct experiment on 11 tested VLMMs, including our proposed MECoT, on single event recognition. The VLMM is provided with 20 candidate event categories, and asked to choose the most likely event that is actually happening in the video. We report 'Accuracy' on 2400 videos sampled from the Kinetics-700 validation set. The VLMM could correctly recognize each events separately, achieving 90% accuracy on average. 'Random Chance' represents the accuracy of randomly selecting the correct answer.

## 5. Single Event Recognition Capability in Various VLMMs

Before evaluating VLMMs' ability to understand multiple events in videos, we first assess their single event recognition capabilities. This preliminary experiment ensures that models can accurately recognize individual events.

In this task, each VLMM is tasked with identifying the single major action in the video from a set of 20 candidate action categories $\mathcal{C}^e$. For each VLMM, we provide video clips along with 20 action category labels - the correct class of the video clip plus 19 randomly selected classes - to evaluate their classification accuracy. We set $|\mathcal{C}^e| = 20$ to evaluate the single-event recognition capabilities required by our VECTOR benchmark while covering diverse action categories. For this experiment, we randomly sampled 2,400 videos from the Kinetics-700 validation set. As shown in Fig. 2, the VLMMs achieve mean accuracy rates above 90 %, indicating strong proficiency in single-event recognition.

## 6. Details about MECoT

### 6.1. Constructing Multi-Event Video Description Dataset

Existing instruction-tuning datasets [32, 37] often compress multi-event scenarios into single comprehensive summary, losing the ordered information of fine-grained events. For instance, a video containing multiple distinct actions (entering a room, sitting, and typing) may be simplified as 'person working at a desk.', removing key temporal dependencies of actions. To address this limitation,

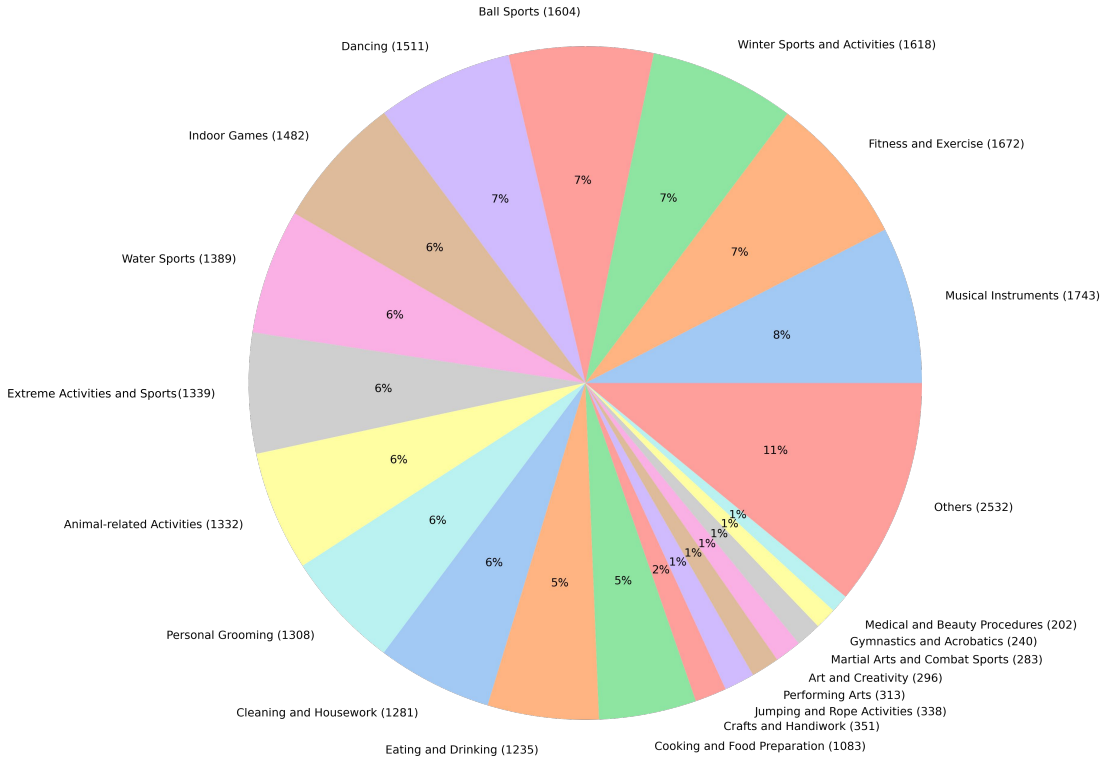



Figure 3. **Top 20 subset of semantic group $\mathcal{G}$.** We visualize the top 20 semantic groups for visualization clarity, used in constructing multi-event videos. These groups are derived from Kinetics-700 classes [37] through an iterative refinement process. Initially, we use Claude 3.5 [2] to create 50 preliminary groups, followed by three rounds of revision involving expert review by a vision-language researcher and subsequent refinement using Claude 3.5. The values in parentheses indicate the number of video instances within each event group.

we construct a multi-event description dataset using a two-stage approach, as illustrated in Fig. 4-(a).

We first sample 56k short videos from the Kinetics-700 dataset [37] in training split, then create synthetic video sequences by temporally combining these clips into sets of 3, 4, 5, 6, and 8 segments. To generate multi-event video descriptions for these synthetic sequences, we follow a two-stage process. First, a LLaVA-OneVision-7B [15] produces detailed individual descriptions for each video segment. These individual descriptions are then integrated into a coherent narrative with GPT-4o-mini [31]. This approach results in 120k temporally enriched descriptions for our synthetic videos. These descriptions are then used to fine-tune LLaVA-OneVision-7B, forming the foundation of MECoT's base model, $\mathcal{M}$, which serves as the backbone for multi-event temporal reasoning.

## 6.2. Chain-of-Thought Reasoning

Although multi-event instruction fine-tuning helps temporal understanding, *explicitly* articulating the reasoning process is essential for recognizing and interpreting events in multi-event videos [45, 53]. Hence, MECoT uses a CoT inference strategy [1, 53, 54], as shown in Fig. 4-(b). We obtain our MECoT by fine-tuning a pre-trained VLMM (7B) [15] with the above synthetically generated data, and adopting the chain-of-thought in the inference stage. Starting from our fine-tuned foundation model, $\mathcal{M}$, CoT explicitly structures the reasoning process by guiding the model through sequential event analysis step by step.

Specifically, given a multi-event video $V$, the fine-tuned $\mathcal{M}$ first generates a chronological *video context* $d$ using a generation prompt $p_{\text{gen}}$:

$$d = \mathcal{M}(p_{\text{gen}}, V). \tag{1}$$

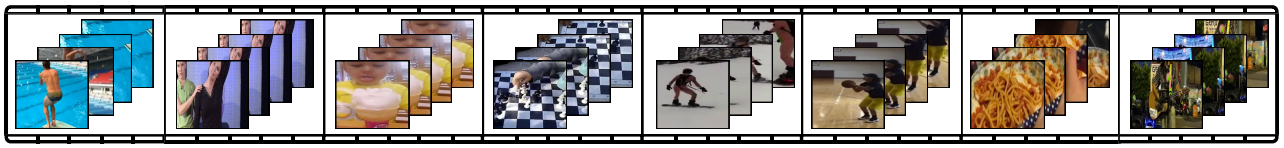
Then, the query prompt $p_{\text{query}}$ is concatenated with the generated context to predict $y$:

$$y = \mathcal{M}\left(p_{\text{query}} || d, V\right). \tag{2}$$

This two-step process helps the model capture the temporal structure of multi-event videos more effectively. As demonstrated in Tab. 6, MECoT shows significantly improved capabilities in temporal reasoning and event sequence understanding compared to the baseline model.

Understanding temporal order of **'events'**

A video of multiple events



*Time*

**Task 1: Event sequencing**

Question: The given video consists of 8 distinct clips, each featuring a different human activity. Your task is to analyze the video and ***identify the actions present in each clip in sequential order***. You must use only the predefined action classes listed below, ensuring they align with the activities described in the video.
The available action classes are:
{'A': 'playing basketball', 'B': 'eating spaghetti', 'C': 'diving', 'D': 'climbing ladder', 'E': 'stretching', 'F': 'dancing macarena',
'G': 'playing chess', 'H': 'playing guitar', 'I': 'ice climbing', 'J': 'skateboarding', 'K': 'snowkiting', 'L': 'playing ukulele', 'M': 'snorkeling',
'N': 'parasailing', 'O': 'ice fishing', 'P': 'eating icecream', 'Q': 'snowboarding', 'R': 'riding a bike', 'S': 'longboarding', 'T': 'busking'}.
Your answer should contain exactly 8 different actions in the order they appear in the video. The response format must be a Python list containing only the corresponding action labels (e.g., ['C', 'H', 'M', 'N', 'A', 'E', 'D', 'G']). The list must have exactly 8 different elements.

Answer: ['C', 'E', 'P', 'G', 'Q', 'A', 'B', 'T']

**Task 2: Relative event sequencing**

Question: The given video consists of 8 distinct clips, each featuring a different human activity. Your task is to ***identify the actions present in the video between the action of eating icecream and busking.*** You must use only the predefined action classes listed below, ensuring they align with the activities described in the video.
The available action classes are:
{'A': 'playing basketball', 'B': 'eating spaghetti', 'C': 'diving', 'D': 'climbing ladder', 'E': 'stretching', 'F': 'dancing macarena',
'G': 'playing chess', 'H': 'playing guitar', 'I': 'ice climbing', 'J': 'skateboarding', 'K': 'snowkiting', 'L': 'playing ukulele', 'M': 'snorkeling',
'N': 'parasailing', 'O': 'ice fishing', 'P': 'eating icecream', 'Q': 'snowboarding', 'R': 'riding a bike', 'S': 'longboarding', 'T': 'busking'}.
Your answer should contain exactly 4 different actions in the order they appear in the video. The response format must be a Python list format, containing only the corresponding action labels (e.g., ['C', 'A', 'B', 'E']). Do not include the two mentioned actions, eating icecream and busking in your answer list. The list must have exactly 4 element(s).

Answer: ['G', 'Q', 'A', 'B']

**Task 3: Event position identification**

Question: You will watch a video that contains 8 combinations of video clips about human activities, each showing different human activities. In ***what order*** is the [***eating ice cream***] action performed in the video? Answer ONLY the exact one position number in the integer format ranging from 1 to 8. (e.g., 1)

Answer: 3

Question: You will watch a video that contains 8 combinations of video clips about human activities, each showing different human activities. In ***what order*** are the [***diving and playing basketball***] actions performed in the video? Answer ONLY the exact two position numbers in list format as integers ranging from 1 to 8. (e.g., [1, 3])

Answer: [1, 6]

Question: You will watch a video that contains 8 combinations of video clips about human activities, each showing different human activities. In ***what order*** are the [***stretching, playing chess and eating spaghetti***] actions performed in the video? Answer ONLY the exact three position numbers in list format as integers ranging from 1 to 8. (e.g., [1, 2, 3])

Answer: [2, 4, 7]

Figure 4. **Prompt details employed for understanding temporal order of events.** We decompose the event-level temporal order understanding task into three sub-tasks: event sequencing task, relative event sequencing task and event position identification task. We divide event position identification task into three variations (1 to 3) according to the number of events to be identified.


Figure 5. **Prompt details employed for understanding temporal order of patterns.** We decompose the pattern-level temporal order understanding into two sub-tasks: discordant semantic-group position identification task and discordant event position identification task. We divide discordant event position identification task into four variations ($s_1s_2s_1s_2s_1s_2$ + x, $s_1s_2s_3s_1s_2s_3$ + x, $s_1s_2s_1s_2s_1s_2s_1s_2$ + x, $s_1s_2s_3s_1s_2s_3s_1s_2s_3$ + x) according to the temporal patterns in video composition.