arXiv:2505.01583v1 [cs.CV] 2 May 2025

TEMPURA: Temporal Event Masked Prediction and Understanding for Reasoning in Action

Jen-Hao Cheng [1], Vivian Wang [1], Huayu Wang [1], Huapeng Zhou [1], Yi-Hao Peng [2], Hou-I Liu [3], Hsiang-Wei Huang [1], Kuang-Ming Chen [1], Cheng-Yen Yang [1], Wenhao Chai a[1], Yi-Ling Chen [4], Vibhav Vineet [4], Qin Cai, Jenq-Neng Hwang [1]

[1] University of Washington
[2] Carnegie Mellon University
[3] National Yang Ming Chiao Tung University
[4] Microsoft

`Link:` `Project Page` | `VER Data` | `Code & Model`

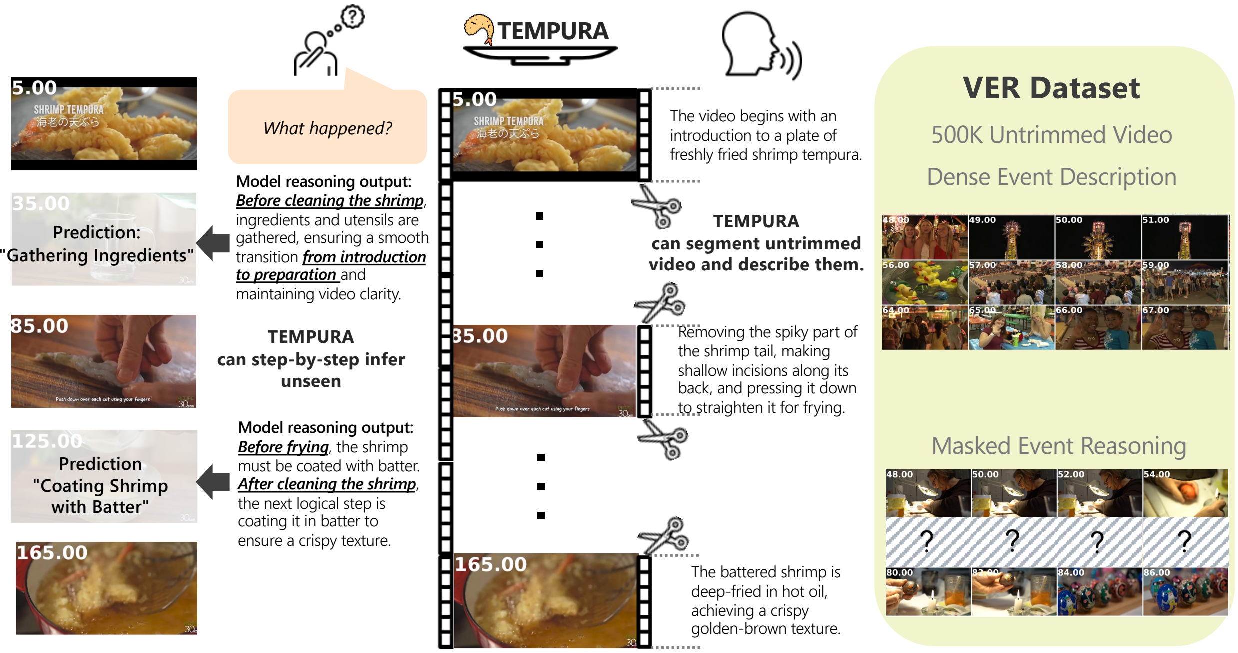

5.00
35.00
Prediction:
"Gathering Ingredients"
85.00
125.00
Prediction
"Coating Shrimp with Batter"
165.00
What happened?
Model reasoning output:
Before cleaning the shrimp, ingredients and utensils are gathered, ensuring a smooth transition from introduction to preparation and maintaining video clarity.
TEMPURA
can step-by-step infer unseen
Model reasoning output:
Before frying, the shrimp must be coated with batter. After cleaning the shrimp, the next logical step is coating it in batter to ensure a crispy texture.
TEMPURA
The video begins with an introduction to a plate of freshly fried shrimp tempura.
TEMPURA
can segment untrimmed video and describe them.
Removing the spiky part of the shrimp tail, making shallow incisions along its back, and pressing it down to straighten it for frying.
The battered shrimp is deep-fried in hot oil, achieving a crispy golden-brown texture.
VER Dataset
500K Untrimmed Video
Dense Event Description
Masked Event Reasoning
?
?
?
?

Fig. 1: Our model, TEMPURA, is trained using a two-stage process for video understanding. The model first infers event structures and causal relationships by filling in missing details and reasoning about event sequences (e.g., recognizing that shrimp must be battered before frying). Second, it is learned to partition video into non-overlapping events and describe them in details. To achieve TEMPURA, we propose a new large-scale dataset consisting of 500k videos with dense event captions.

Understanding causal event relationships and achieving fine-grained temporal grounding in videos remain challenging for vision-language models. Existing methods either compress video tokens to reduce temporal resolution, or treat videos as unsegmented streams, which obscures fine-grained event boundaries and limits the modeling of causal dependencies. We propose TEMPURA (Temporal Event Masked Prediction and Understanding for Reasoning in Action),

a two-stage training framework that enhances video temporal understanding. TEMPURA first applies masked event prediction reasoning to reconstruct missing events and generate step-by-step causal explanations from dense event annotations, drawing inspiration from effective infilling techniques. TEMPURA then learns to perform video segmentation and dense captioning to decompose videos into non-overlapping events with detailed, timestamp-aligned descriptions. We train TEMPURA on VER, a large-scale dataset curated by us that comprises 1M training instances and 500K videos with temporally aligned event descriptions and structured reasoning steps. Experiments on temporal grounding and highlight detection benchmarks demonstrate that TEMPURA outperforms strong baseline models, confirming that integrating causal reasoning with fine-grained temporal segmentation leads to improved video understanding.

1. Introduction

Recent video Large Multi-modal Models (LMMs) [1, 21, 26, 27] have extended Large Language Models (LLMs) with video understanding capabilities. However, understanding and reasoning over the temporal relationships in long videos remains challenging for current models, particularly when analyzing events over time. Recent methods compress video tokens by consolidating key features from adjacent frames [18, 40, 45], which reduces computational and memory costs but leads to fine-grained temporal information loss. Some other works construct synthetic datasets and develop training pipelines to improve temporal reasoning. For example, LLaVA-Video [55] curates large-scale, high-quality video data for video-language instruction fine-tuning, and TPO [24] uses contrast training pairs with preference learning to steer models toward contextually appropriate responses. However, these approaches still struggle to capture fine-grained event dependencies and achieve long-video temporal understanding.

To address these limitations, we introduce **TEMPURA** (***T**emporal **E**vent **M**asked **P**rediction and **U**nderstanding for **R**easoning in **A**ction*), a two-stage training pipeline that unifies dense event segmentation with masked event prediction to build robust video temporal understanding LMMs. In the first stage, TEMPURA enhances video reasoning by teaching the model to infer missing events and generate step-by-step causal explanations. Drawing inspiration from the Fill-in-the-Middle (FIM) paradigm [2, 39], our training pipeline masks segments of dense video captions and leverages a strong LLM to predict pseudo-events and associated reasoning steps. This training objective maximizes the likelihood of reconstructing both the absent event and its causal narrative from the surrounding context, thereby aligning vision-based inference with language-based reasoning. The second stage focuses on video segmentation and dense captioning, where the model learns to partition untrimmed videos into non-overlapping events with precise start and end timestamps, each enriched with detailed descriptions. This stage eliminates the need for auxiliary temporal encoders by directly grounding each event in its corresponding video segment.

To support TEMPURA's training pipeline, we introduce VER, a large-scale dataset constructed through a multi-step event annotation pipeline (see Figure 4). The pipeline begins by filtering dynamic content from YT-1B [50] and categorizing videos into 10 common categories using Llama-3-72B [7] while discarding videos dominated by interviews, lectures, or speeches. We then applied GPT-4o [16] to segment each video by sampling frames at 1 FPS and arranging them into chronological frame sequence image, which facilitates accurate event boundary detection and dense caption generation. A temporal coherence check further refines the data by filtering out events lacking causal relevance, and a masked event prediction subset reinforces the training signal for temporal inference. The resulting

Table 1: **Video Dataset Characteristics Comparison across mainstream benchmarks.**

Dataset	Video Hours	Events per Video	Events per Minute	Coverage	Event Details	Temporal Reasoning
Youcook2 [58]	175	7.7	1.5	Sparse	✓	×
Charades [9]	476	6.8	2.3	Sparse	✓	×
ActivityNet Captions [19]	849	3.6	2.0	Sparse	×	×
Finevideos [8]	3,425	-	-	Dense	✓	×
ViTT [15]	541	7.1	1.5	Sparse	✓	×
Moment-10M [30]	7,260	**22.5**	<u>3.3</u>	Dense	✓	×
VER (Ours)	**18,329**	<u>10.5</u>	**6.0**	Dense	✓	✓

VER dataset comprises 500K untrimmed videos spanning a total duration of 18K hours, providing dense, timestamp-aligned event captions and structured reasoning that capture fine-grained temporal dynamics across diverse video types.

Our experiments demonstrate the effectiveness of TEMPURA in video temporal understanding tasks. On the Charades-STA benchmark [9], TEMPURA achieves a mIoU of 39.2, outperforming the baseline by 6.3 points. On the QVHighlights dataset [20], it attains a HIT@1 score of 51.7, surpassing the baseline by 6.9 points. Ablation studies reveal that sequentially applying masked event prediction followed by dense video captioning is crucial for unlocking fine-grained temporal reasoning, thereby enhancing the model's performance in video understanding.

In summary, TEMPURA advances video understanding by integrating dense video captioning with structured causal reasoning to capture fine-grained temporal dynamics in long videos. By decomposing videos into non-overlapping events with precise timestamps and enabling the model to infer missing events through masked prediction, TEMPURA goes beyond holistic processing to achieve robust temporal grounding and causal inference. Our contributions are twofold:

- We develop TEMPURA, a novel training pipeline that leverages masked event prediction to reconstruct missing events with step-by-step causal explanations, and then refines temporal grounding via dense event segmentation and captioning.
- We curate VER, a large-scale dataset of 500K videos spanning 18K hours, annotated with diverse, timestamp-aligned event captions and structured reasoning across 10 common video categories.

2. Related Work

2.1. Video Large Multi-modal Models

Researchers have developed video Large Multi-modal Models (LMMs) that address a broad range of video understanding tasks and its application [56, 57]. Many models integrate vision foundation models [34, 52] with Large Language Models [3, 7, 33] to enhance video question answering. Several approaches [4, 17, 40, 40, 41, 46, 48] rely on token merging strategies to fuse visual tokens to enable long video question answering. Models such as LLaVA-OneVision [21] and LLaVA-Next-Interleave [22], which extend the LLaVA architecture [26] with a simple projector design, demonstrate strong performance across both image and video question answering. The Video-LLaMA series [6, 53] further incorporates an audio modality, supporting more fine-grained multi-modal video comprehension. Recent works [11, 13, 36] reveal, however, that many LMMs still struggle with temporal reasoning. The limited ability to capture the order of events arises from a shortage of temporally structured video training data and training methods that overlook time causality. To enhance temporal reasoning and understanding in LMMs, we propose masked temporal event learning in our training pipeline, which strengthens models' ability to predict event order in videos.

2.2. LLM with Reasoning

Recent advancements in LLMs have significantly improved their reasoning capabilities, enabling them to handle complex multi-step problems across various domains. Latest models like DeepSeek-R1 [10] involve reinforcement learning during the training process and achieve state-of-the-art performance across various LLM evaluation benchmarks with its strong reasoning ability. On the other hand, parallel efforts in LMMs have similarly advanced image-based reasoning, as demonstrated by studies training the multi-modal models to generate step-by-step solutions for math problems [54] or perform chain-of-thought reasoning for object localization or visual reasoning [29, 38, 42]. This emphasis on step-by-step reasoning in static domains naturally aligns with approaches like LLaVA-CoT [47], which leverages four sequential stages during model inference to guide models in systematically breaking down problems and delivering more accurate responses. However, despite these advances, the application of such reasoning capabilities to the video domain, particularly for temporal understanding across dynamic sequences, remains largely unexplored, with few works developing large multi-modal models to address these challenges.

2.3. Temporal Understanding with LMMs

Temporal understanding in videos is essential for comprehending event relationships and causal dependencies, enabling models to interpret actions, anticipate future occurrences, and infer missing visual events. Video LMMs have been developed to facilitate temporal grounding through timestamp-based event localization and video captioning. Models such as TimeMarker [5], VTimeLLM [13], and Momentor [31] enhance video comprehension through adaptive token compression, segment-level event alignment, and fine-grained moment localization. Additionally, Trace [11], TimeSuite [51], and TimeChat [37] introduce refined temporal modeling techniques, incorporating structured temporal embeddings and improved event localization. However, these works primarily focus on timestamp retrieval and event segmentation, lacking the ability to infer missing events and reason about causal dependencies between actions. In this work, we address these limitations by incorporating masked event prediction and structured temporal reasoning, enhancing the coherence of event transitions and improving long video comprehension, thereby advance the fine-grained temporal reasoning ability of video LMMs.

3. Method

Understanding and reasoning about video content require the ability to segment a video into meaningful events, establish their temporal order, and infer relationships among them. We define **video reasoning** as the capability to: 1) Comprehend video progression by identifying distinct events and their temporal boundaries, and 2) Analyze event relationships to infer missing or implicit information based on context and logical flow.

To develop a video LMMs with robust video reasoning capabilities, we propose a structured training framework comprising two key stages: Masked Event Prediction Reasoning and Video Segmentation and Dense Captioning. The first stage enables the model to infer missing events and reason about causality within the video context, while the second stage focuses on enhancing the

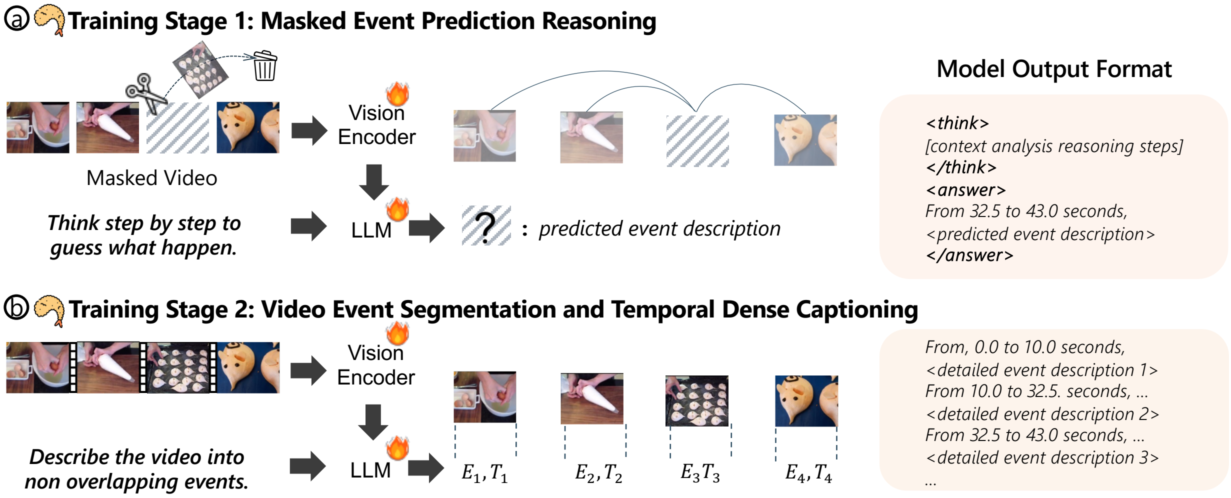

ⓐ Training Stage 1: Masked Event Prediction Reasoning
Masked Video
Vision Encoder
LLM
Think step by step to guess what happen.
?
: predicted event description
Model Output Format
<think>
[context analysis reasoning steps]
</think>
<answer>
From 32.5 to 43.0 seconds,
<predicted event description>
</answer>
ⓑ Training Stage 2: Video Event Segmentation and Temporal Dense Captioning
Vision Encoder
LLM
Describe the video into non overlapping events.
E_1, T_1
E_2, T_2
$E_3 T_3$
E_4, T_4
From, 0.0 to 10.0 seconds,
<detailed event description 1>
From 10.0 to 32.5. seconds, ...
<detailed event description 2>
From 32.5 to 43.0 seconds, ...
<detailed event description 3>
...

Fig. 2: Overview of TEMPURA's two-stage training pipeline. **(a) Masked Event Prediction Reasoning:** The model infers missing events by analyzing the masked video context, generating both a textual description and step-by-step causal explanations. **(b) Video Event Segmentation and Temporal Dense Captioning:** The model partitions an untrimmed video into non-overlapping events, each aligned with precise start/end timestamps and enriched with detailed captions, thereby reinforcing a structured understanding of temporal progressions.

video LMM's ability to decompose a video into temporally grounded event sequences. Together, these stages equip the video LMM with a structured understanding of video narratives, improving its generalization to downstream tasks such as temporal grounding and highlight detection.

3.1. Masked Event Prediction

To enhance the video LMM's ability to reason from video input, we introduce **Masked Event Prediction**, a novel training stage that aims to enhance the model's understanding of event logical flow, causality, and inductive reasoning with that of a language model. Inspired by *Fill-in-the-Middle (FIM)* [2, 39], which is widely used in code and text infilling tasks, we extend this concept to the video domain. FIM typically trains a model to predict missing content based on preceding and succeeding contexts. Similarly, we formulate a video event infilling task where the video LMM learns to reconstruct masked video events through inferred text description.

To enable this capability, we leverage the strong reasoning ability of LLMs to generate pseudo-events and reasoning steps based on our dense video caption data, detailed in Section 4. Specifically, we prompt the LLM to infer and construct plausible intermediate events that are masked within a video sequence, ensuring logical consistency with the surrounding context. As shown in Figure 2a, we apply segment-level masking to dense video captions and use the LLM to produce pseudo-events with step-by-step reasoning explanations for the missing segments. These generated pseudo-events and reasoning steps serve as supervised fine-tuning data for the video LMM, enabling it to align its video-based reasoning capability with the strong contextual understanding of LLMs. By training the video LMM on this curated data, we reinforce its ability to infer missing content and establish logical event progression solely from video input.

Formally, given a masked video input, V_{masked}, the training objective is to maximize the likelihood of predicting the pseudo-event, $\tilde{E}$, along with intermediate reasoning steps, R, in a predefined structured format:

$$\max_{\theta} \mathbb{E}_{V_{\text{masked}}} \left[P_{\theta}(\tilde{E}, R \mid V_{\text{masked}}) \right]$$

This stage bridges the gap between vision and language-based reasoning by aligning the strong logical filling ability of the LLM with the video understanding of the video LMM. Making the model more effective on downstream tasks that require complex video comprehension.

3.2. Video Segmentation and Dense Captioning

Dense video captioning [13, 14, 51] is a crucial task for fine-grained video understanding. The resulting video events, grounded with timestamps, provide the necessary context for a language model to establish relationships between events, assisting it in extracting facts and reasoning in response to queries.

In the second training stage, Video Event Segmentation and Temporal Dense Captioning, we teach the model to break down a video into non-overlapping events and describe each event in detail. As illustrated in Figure 2b, we develop the video LMM's temporal awareness by learning to segment a video into non-overlapping events, each defined by its start and end timestamps.

We design an instruction, I, to guide the video LMM in transforming a video input, V, into a structured event sequence, $\{E_i \mid 1 \le i \le N\}$, where each event is represented by its timestamp and caption, $E = (T, C)$. Unlike Trace [11], which utilizes extra encoders to model time and saliency scores, we eliminate these components and instead train the model to ground all video segments using their enclosing timestamps. This is achieved by leveraging dense video captions from our VER dataset, which consists of 500K annotated videos. This design choice reduces the need for additional parameters, making the video LMM more versatile for various tasks while ensuring that it learns the structural and temporal progression of videos in this initial training stage.

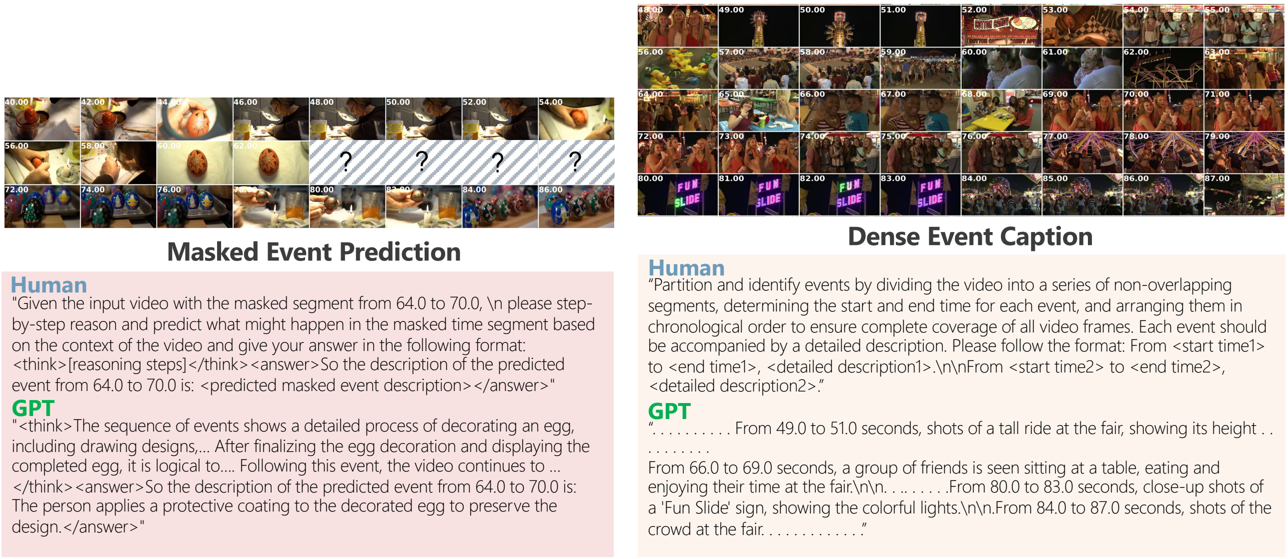
4. VER Data Pipeline

Our TEMPURA training pipeline equips video LMM with three key capabilities: (1) segmenting an untrimmed video into non-overlapping events while ensuring full video coverage, (2) generating detailed descriptions for each segmented event, and (3) building a strong understanding of event logical flow, allowing the model to infer missing events in masked video segments based on contextual cues.

Existing datasets, as summarized in Table 1, lack large-scale timestamp-aligned dense event captions [8, 9, 15, 30, 58] and dense video coverage, where all events comprehensively describe the entire video [8, 30]. To support TEMPURA training, we construct Video Event Reasoning (VER), a large-scale dataset consisting of 500K untrimmed videos spanning a total duration of 18K hours. Our dataset provides non-overlapping video events with corresponding detailed descriptions. Compared to existing datasets, VER offers longer video hours, a diverse range of video types, and fine-grained event segmentation and captions. Additionally, our TEMPURA masked event prediction training leverages temporal event reasoning data generated from our dense event captions.

4.1. Dataset Construction

Figure 4 presents our VER data pipeline. Our video data is filtered from YT-1B [50]. Firstly, we remove static videos following the method in [8] to ensure a richer temporal structure. Next, we categorize



Masked Event Prediction

Human

"Given the input video with the masked segment from 64.0 to 70.0, \n please step-by-step reason and predict what might happen in the masked time segment based on the context of the video and give your answer in the following format: <think>[reasoning steps]</think><answer>So the description of the predicted event from 64.0 to 70.0 is: <predicted masked event description></answer>"

GPT

"<think>The sequence of events shows a detailed process of decorating an egg, including drawing designs,... After finalizing the egg decoration and displaying the completed egg, it is logical to.... Following this event, the video continues to ... </think><answer>So the description of the predicted event from 64.0 to 70.0 is: The person applies a protective coating to the decorated egg to preserve the design.</answer>"

Dense Event Caption

Human

"Partition and identify events by dividing the video into a series of non-overlapping segments, determining the start and end time for each event, and arranging them in chronological order to ensure complete coverage of all video frames. Each event should be accompanied by a detailed description. Please follow the format: From <start time1> to <end time1>, <detailed description1>.\n\nFrom <start time2> to <end time2>, <detailed description2>."

GPT

". . . . . From 49.0 to 51.0 seconds, shots of a tall ride at the fair, showing its height . . . .
From 66.0 to 69.0 seconds, a group of friends is seen sitting at a table, eating and enjoying their time at the fair.\n\n. . .. . .From 80.0 to 83.0 seconds, close-up shots of a 'Fun Slide' sign, showing the colorful lights.\n\n.From 84.0 to 87.0 seconds, shots of the crowd at the fair. . . . ."

Fig. 3: Structured Training Data for Masked Event Prediction and Dense Event Caption

videos into 10 of our predefined common video categories using Llama-3-72B [7] to classify based on video captions.

To define event boundaries, we apply GPT-4o [16] by sampling the video at 1 FPS and arranging the frames into frame sequence images. Each frame is indexed with a marker at the top-left corner, and frame sequence images are ordered chronologically. We then ensure event time boundaries: (1) do not overlap, (2) cover the entire video, and (3) fall within the video length range. Once event boundaries are established, GPT-4o is further utilized to generate detailed event descriptions, compiling them into a structured narrative describing the video's progression and event sequences.

After filtering and alignment, we retain 500K videos with dense event captions. Each annotated video contains a series of events, where each event includes an event ID, description, and start and end timestamps. Figure 3 showcases an example of a video-dense event caption in our fine-tuning format.

4.2. Masked Event Prediction

To enhance video LMM's temporal reasoning and event inference, we leverage strong LLMs for causal understanding and masked event prediction. Specifically, we randomly mask an event from the dense event caption and employ GPT-4o to analyze the structured captions and predict the missing event within the masked time window. To ensure that masked events are logically inferable, we filter out videos with uncorrelated event captions using GPT-4o. We achieve this by prompting GPT-4o to determine whether a causal relationship exists between event captions, applying step-by-step reasoning to arrive at a binary decision. During training, we align LMM's reasoning capabilities with LLM event inference by fine-tuning on these structured reasoning processes, as shown in Figure 3. We provide additional dataset statistics, annotation details, and more data examples in the supplementary material.

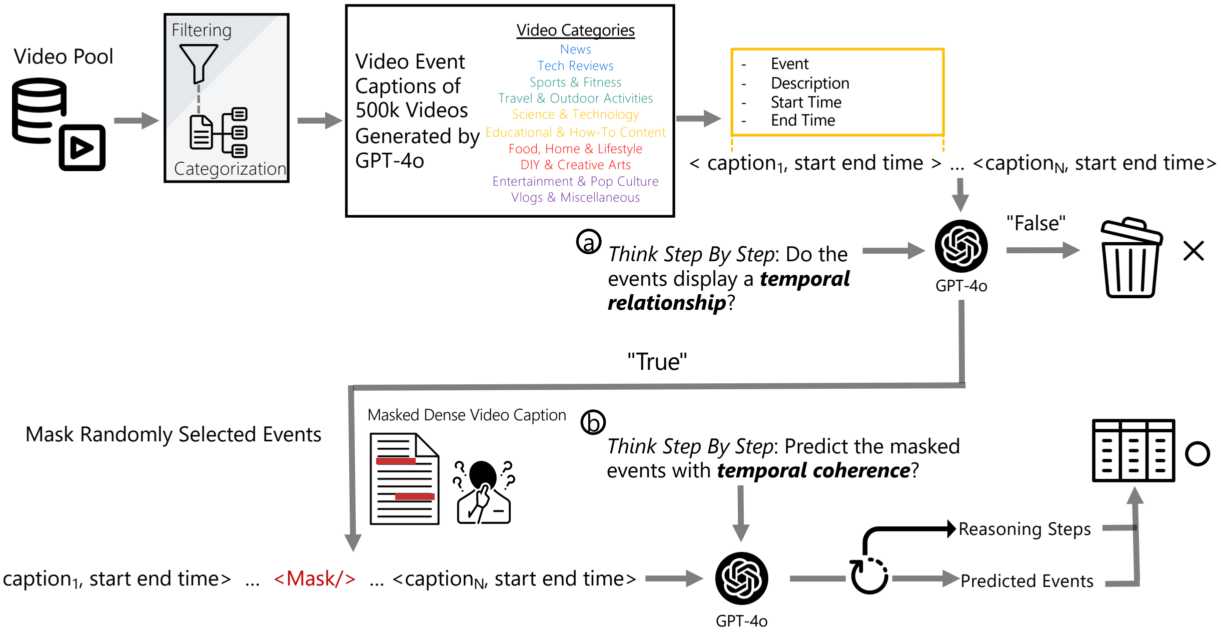

Video Pool
Filtering
Categorization
Video Event Captions of 500k Videos Generated by GPT-4o
Video Categories
News
Tech Reviews
Sports & Fitness
Travel & Outdoor Activities
Science & Technology
Educational & How-To Content
Food, Home & Lifestyle
DIY & Creative Arts
Entertainment & Pop Culture
Vlogs & Miscellaneous
- Event
- Description
- Start Time
- End Time
< caption$_1$, start end time > ... <caption$_N$, start end time>
ⓐ *Think Step By Step*: Do the events display a ***temporal relationship***?
GPT-4o
"False"
✕
"True"
Mask Randomly Selected Events
Masked Dense Video Caption
ⓑ *Think Step By Step*: Predict the masked events with ***temporal coherence***?
○
Reasoning Steps
Predicted Events
<caption$_1$, start end time> ... <Mask/> ... <caption$_N$, start end time>
GPT-4o

Fig. 4: VER Data Pipeline: The pipeline begins by filtering and categorizing a large video pool. GPT-4o then generates event captions with start/end times, followed by a temporal coherence check that discards invalid events. For valid events, a subset is masked to form a fill-in-the-blank task, and GPT-4o infers the missing segments—ultimately creating a dataset for video temporal understanding.

5. Experiments

5.1. Implementation

We adopted Qwen2.5-VL [1] as our base model and conduct training on our collected data. Additionally, we train our model using DeepSpeed Zero2 [35], with the global batch size is set to 64. To fine-tune the LLM and MLP adapter, we use a learning rate of 1×10^{-5}, while the vision encoder is trained with a lower learning rate of 2×10^{-6}. We observed that the original temporal encoding scheme of Qwen2.5-VL tends to misalign for fine-grained temporal grounding, especially in longer videos (see the supplementary material for more examples). To overcome this issue, we introduced two key modifications. First, we overlay **visual timestamps** on the upper left corner of each sampled video frame to explicitly mark the temporal context. Second, we adjusted the temporal encoding in M-RoPE by assigning a fixed position ID to every sampled frame, ensuring that the model reliably associates each frame with its corresponding timestamp. We conducted training on 8 NVIDIA H100 GPUs for 1 epoch in each training stage. More training details can be found in the supplementary material.

5.2. Video Temporal Understanding

Our evaluation focuses on video temporal understanding benchmarks, where the goal is to accurately localize temporal events within videos based on textual queries. In the following, we detail our evaluation of two specific tasks: Video Temporal Grounding and Highlight Detection.

Table 2: **Video Temporal Grounding** on Charades-STA and **Highlight Detection** on QVHighlight. The top half reports models fine-tuned with the benchmark training sets while the bottom half shows zero-shot performance. FT denotes fine-tuned models.

Method	**LLM Size**	**Charades-STA**				**QVHighlight**	
		mIoU	**R@1 (IoU=0.3)**	**R@1 (IoU=0.5)**	**R@1 (IoU=0.7)**	**mAP**	**HIT@1**
QD-DETR (FT) [28]	-	-	-	57.3	32.6	38.9	64.2
UnLoc-L (FT) [49]	-	-	-	60.8	38.4	-	-
HawkEye (FT) [44]	7B	49.3	72.5	58.3	28.8	-	-
TimeChat (FT) [37]	7B	-	-	46.7	23.7	21.7	37.9
VideoChat-T (FT) [51]	7B	-	79.4	67.1	43.0	27.0	55.3
MovieChat [40]	7B	-	8.8	2.9	1.3	11.7	16.1
GroundingGPT [25]	7B	-	-	29.6	11.9	-	-
VTimeLLM [13]	7B	31.2	51.0	27.5	11.4	-	-
HawkEye [44]	7B	33.7	50.6	31.4	14.5	-	-
TimeChat [37]	7B	-	-	32.2	13.4	14.5	23.9
Trace [11]	7B	-	-	40.3	19.4	26.8	42.7
ChatVTG [32]	7B	-	52.7	33.0	15.9	-	-
VideoChat2 [23]	7B	34.9	9.6	3.4	1.4	13.4	18.6
Momentor [31]	7B	28.5	42.6	26.6	11.6	7.6	-
Grounded-VideoLLM [43]	4B	36.8	54.2	36.4	19.7	36.8	46.2
Qwen-VL-2.5 [1]	3B	33.1	52.4	34.3	12.5	42.1	44.8
TEMPURA (Ours)	**3B**	**39.2** (+6.3)	63.8 (+11.4)	39.3 (+5.0)	15.0 (+2.5)	**48.3** (+6.2)	**51.7** (+6.9)

Video Temporal Grounding. Video temporal grounding aims to localize specific moments in a video based on a language query. We evaluate our model on Charades-STA [9] using mean Intersection over Union (mIoU) and Recall@1 at different IoU thresholds following previous work [43], assessing both temporal localization accuracy and recall. Our model is capable of providing more granular descriptions of videos, including more detailed content descriptions and greater sensitivity to temporal intervals. In the example shown in Figure 5, given the same prompt, our model not only expresses the same meaning but also provides more detailed grounding and descriptions for each step. We also analyzed performance on Youcook2[58]. Compared to the baseline model's average of 15.53 events per video, our model achieves 27.49 events, demonstrating significantly more detailed temporal understanding and description capabilities.

Highlight Detection. The goal of highlight detection is to identify relevant time windows within a video and predict saliency scores based on a given language query. We evaluate our model on QVHighlights [20], reporting mean Average Precision (mAP) and HIT@1 as evaluation metrics. HIT@1 measures whether the highest-ranked retrieved time window aligns with the ground truth. Unlike video temporal grounding, which focuses on localizing a single moment, highlight detection aims to retrieve all relevant time windows corresponding to the query.

Table 2 shows that TEMPURA improves the baseline model by 6.3 mIoU, and either matches or exceeds the state-of-the-art in video temporal grounding—all without any target-task fine-tuning and with a smaller model size. In contrast to previous approaches that rely on various forms of instruction tuning data for video temporal grounding [13, 25, 37, 43, 44], our method trains the model to segment a video into a series of events, infer their relationships, and describe them in detail. As a result, TEMPURA not only eliminates the need for extra components such as time prediction models [11], temporal encoding tokens [43], and video-specific vision encoders [37], but also outperforms methods like [32] that are optimized for generating dense captions and extracting time windows from model

Table 3: TEMPURA Training Stages **S1**: Masked Event Prediction. **S2** Event Segmentation and Temporal Captioning.

Training Stages	mIoU	R@1 (IoU=0.3)	R@1 (IoU=0.5)
S2	38.4	59.1	32.8
S2 → S1	34.0	55.6	32.6
S1 → S2	**39.2**	**63.9**	**39.3**

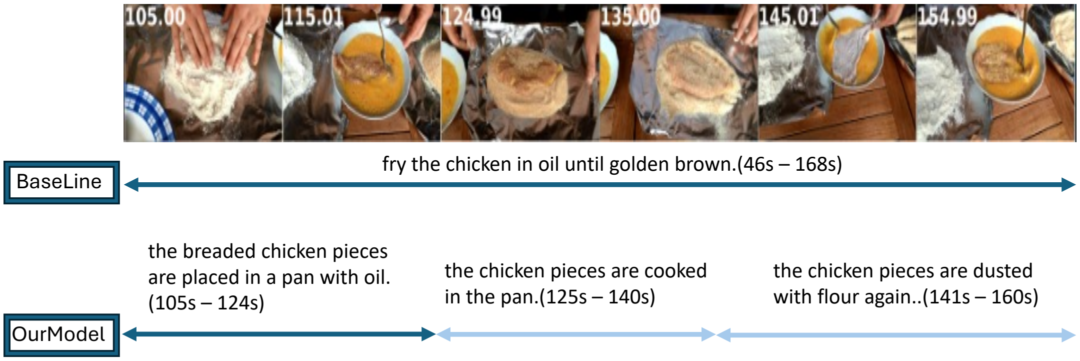

105.00
115.01
124.99
135.00
145.01
154.99
BaseLine
fry the chicken in oil until golden brown.(46s – 168s)
the breaded chicken pieces are placed in a pan with oil. (105s – 124s)
the chicken pieces are cooked in the pan.(125s – 140s)
the chicken pieces are dusted with flour again..(141s – 160s)
OurModel

Fig. 5: Our model can segment videos into more fine-grained events, capturing subtle transitions and short-duration activities. In contrast, the baseline model (QwenVL2.5) tends to generate coarser segments. This difference suggests that our approach is more adept at recognizing and differentiating fine-grained patterns within video sequences, leading to detailed and structured event representation.

outputs. TEMPURA also enhances the performance in highlight detection by 6.9 HIT@1 over the baseline model and surpasses other methods. The superior performance of our model in two tasks demonstrates that the model's learned fine-grained temporal understanding ability trained with our TEMPURA pipeline and the data curated in VER can be easily adapted in downstream video temporal understanding tasks without fine-tuning on the benchmark datasets.

5.3. Ablation Study

To study the effectiveness of each component in TEMPURA, we split our ablation analysis into three parts and report mIoU and R@1 (IoU=0.5) on Charades-STA.

TEMPURA Training Stage. TEMPURA uses masked event prediction as the first training stage, and video dense captioning as the second training stage. As shown in Table 3, we found that using mask event prediction as the pre-training stage before dense captioning will enhance the model's temporal understanding of videos. On the contrary, training the model first on dense captioning and continuing to fine-tune the model on the masked event prediction tasks would not improve the model to follow temporal grounding instructions since the model was not explicitly trained to segment video into fine-grained events. Nonetheless, we can still observe the model starts to extract facts around the masked video time windows and generate longer reasoning steps to predict plausible infill as shown in the supplementary material. We compare our model's generated masked event prediction and reasoning steps with the baseline model in the supplementary material.

Table 4: Temporal Encoding Scheme. We found adding visual timestamp on sampled video frames provide the most accurate and robust way to encode time. **V.T.** means adding visual timestamp to the images. **T.M.** means using temporal MRoPE for the encoding. **T.I.** means appending time instruction in the prompt.

V.T.	T.M.	T.I.	mIoU	R@1 (IoU=0.3)	R@1 (IoU=0.5)
✓	✓		25.6	33.1	16.7
✓	✓	✓	26.7 (+1.1)	39.0 (+5.9)	22.1 (+4.4)
✓			38.4	59.1	32.8
✓		✓	38.9 (+0.5)	64.2 (+5.1)	38.5 (+5.7)

Table 5: Dynamic Scene and Relevant Segment Filtering. **D.S.**: Dynamic Scene Filtering, **T.R.**: Relevant Segment Filtering

Data Filtering	mIoU	R@1 (IoU=0.3)	R@1 (IoU=0.5)
No D.S.	34.8	51.3	27.4
D.S.	38.7 (+3.9)	63.7 (+12.4)	38.3 (+10.9)
No T.R.	33.0	47.3	28.3
T.R.	37.5 (+4.5)	57.1 (+9.8)	34.8 (+6.5)

Temporal Encoding Scheme. In Table 4, we compared three different temporal encoding schemes during model training: using absolute temporal encoding in M-RoPE, appending time instruction ahead of user queries, and adding visual timestamps. We found that adding visual timestamps provides the best-grounded captions with the timestamps. Since our model was pre-trained with a large amount of OCR data, and the LLM is good at understanding structured information, overlaying the visual timestamp on each sampled video frame will naturally allow the model to understand videos' progression. In addition, we show that the baseline model Qwen2.5-VL tends to misalign the description with the correct timestamps. On the contrary, our model's temporal grounding and captioning are robust when the video gets longer.

Dynamic Scene and Relevant Event Filtering. Our TEMPURA model is learned to partition the videos into non-overlapping segments and describe the segments focusing on the video progression. We found that fine-tuning the base model without filtering out static scenes would weaken the model's grounding and captioning ability. Static scenes contain redundant video frames and fewer semantics, and training on such video-text pairs will make the model leans to generate shorter descriptions. During the masked event prediction stage, it is crucial to filter out videos with non-relevant events. Since we train the model to predict possible events in masked time windows, the model learns to build casual bidirectional thinking around past and future video content. Training with masked event data generated from videos with non-relevant events weakens the model's temporal understanding as shown in Table 5.

6. Conclusions

In this work, we present TEMPURA, a two-stage training framework to enhance video LMM's temporal understanding by intergrating coarse visual extraction with deep causal reasoning. Furthermore, we proposed VER, a large-scale video event reasoning dataset that aims to enhance the temporal understanding ability of video LMMs. After trained on VER, our model substantially outperforms strong baseline model Qwen2.5-VL on multiple temporal understanding benchmarks for temporal grounding and highlight detection tasks. Meanwhile, our ablation studies reveal that the integration of masked event prediction and follow-up fine-grained segmentation further improve video LMM's performance on video temporal understanding.

Acknowledgments

We gratefully acknowledge Ms. Grace Chao for sponsoring the GPU training hours that supported this work.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

[2] Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle, 2022. URL https://arxiv.org/abs/2207.14255.

[3] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, et al. Internlm2 technical report, 2024.

[4] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. *arXiv preprint arXiv:2410.03051*, 2024.

[5] Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability, 2024. URL https://arxiv.org/abs/2411.18211.

[6] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. https://huggingface.co/datasets/HuggingFaceFV/finevideo, 2024.

[9] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5277–5285, 2017. doi: 10.1109/ICCV.2017.563.

[10] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[11] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Qingbin Liu, and Xi Chen. Trace: Temporal grounding video llm via causal event modeling, 2024. URL https://arxiv.org/abs/2410.05643.

[12] Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, and Yanning Chen. Liger kernel: Efficient triton kernels for llm training. *arXiv preprint arXiv:2410.10989*, 2024. URL https://arxiv.org/abs/2410.10989.

[13] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280, 2024.

[14] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *ECCV*, 2024.

[15] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. In *AACL-IJCNLP 2020*, 2020.

[16] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[17] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024.

[18] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024.

[19] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

[20] Jie Lei, Tamara L. Berg, and Mohit Bansal. Qvhighlights: detecting moments and highlights in videos via natural language queries. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

[21] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[22] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

[23] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[24] Rui Li, Xiaohan Wang, Yuhui Zhang, Zeyu Wang, and Serena Yeung-Levy. Temporal preference optimization for long-form video understanding. *arXiv preprint arXiv:2501.13919*, 2025.

[25] Zhaowei Li, Qi Xu, Dong Zhang, Hang Song, Yiqing Cai, Qi Qi, Ran Zhou, Junting Pan, Zefeng Li, Van Tu Vu, et al. Groundinggpt: Language enhanced multi-modal grounding model. *arXiv preprint arXiv:2401.06071*, 2024.

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[27] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.

[28] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23023–23033, 2023.

[29] Yi-Hao Peng, Faria Huq, Yue Jiang, Jason Wu, Xin Yue Li, Jeffrey P Bigham, and Amy Pavel. Dreamstruct: Understanding slides and user interfaces via synthetic data generation. In *European Conference on Computer Vision*, pages 466–485. Springer, 2024.

[30] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning, 2024.

[31] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: advancing video large language model with fine-grained temporal reasoning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[32] Mengxue Qu, Xiaodong Chen, Wu Liu, Alicia Li, and Yao Zhao. Chatvtg: Video temporal grounding via chat with video dialogue large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1847–1856, 2024.

[33] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[35] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986.

[36] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024.

[37] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding, 2024. URL https://arxiv.org/abs/2312.02051.

[38] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning, 2024. URL https://arxiv.org/abs/2403.16999.

[39] Tianxiao Shen, Hao Peng, Ruoqi Shen, Yao Fu, Zaid Harchaoui, and Yejin Choi. Film: Fill-in language models for any-order generation, 2023. URL https://arxiv.org/abs/2310.09930.

[40] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[41] Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*, 2024.

[42] Yael Vinker, Tamar Rott Shaham, Kristine Zheng, Alex Zhao, Judith E Fan, and Antonio Torralba. Sketchagent: Language-driven sequential sketch generation. *arXiv preprint arXiv:2411.17673*, 2024.

[43] Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models, 2024. URL https://arxiv.org/abs/2410.03290.

[44] Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. Hawkeye: Training video-text llms for grounding text in videos. *arXiv preprint arXiv:2403.10228*, 2024.

[45] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024.

[46] Wenhao Wu. Freeva: Offline mllm as training-free video assistant. *arXiv preprint arXiv:2405.07798*, 2024.

[47] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2025. URL https://arxiv.org/abs/2411.10440.

[48] 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 preprint arXiv:2407.15841*, 2024.

[49] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13623–13633, 2023.

[50] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022.

[51] Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. Timesuite: Improving mllms for long video understanding via grounded tuning, 2024. URL `https://arxiv.org/abs/2410.19702`.

[52] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[53] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[54] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, Peng Gao, Chunyuan Li, and Hongsheng Li. Mavis: Mathematical visual instruction tuning with an automatic data engine, 2024. URL `https://arxiv.org/abs/2407.08739`.

[55] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024. URL `https://arxiv.org/abs/2410.02713`.

[56] Zhonghan Zhao, Wenhao Chai, Xuan Wang, Boyi Li, Shengyu Hao, Shidong Cao, Tian Ye, and Gaoang Wang. See and think: Embodied agent in virtual environment. In *European Conference on Computer Vision*, pages 187–204. Springer, 2024.

[57] Zhonghan Zhao, Kewei Chen, Dongxu Guo, Wenhao Chai, Tian Ye, Yanting Zhang, and Gaoang Wang. Hierarchical auto-organizing system for open-ended multi-agent navigation. *arXiv preprint arXiv:2403.08282*, 2024.

[58] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17344`.

Supplementary Material

The supplementary material is structured as follows:

- VER data creation pipeline and statistics in Section S1.
- Implementation details in Section S2.
- Qualitative analysis in Section S3.

S1. VER Data Creation Pipeline and Statistics

After data filtering, we uniformly sampled video frames and arranged them into frame sequence images. The example input to GPT4-o is shown on the right of Figure S1. We added the timestamp in each image, and combined multiple frame sequence images to obtain a grid-formatted composite image as the input. These inputs were first temporally segmented by GPT4-o, and we consider these segments as events. Based on the generated time segments, GPT4-o then generated the descriptions of these events separately. Next, we used these temporally aligned event descriptions to construct our masked event prediction data. To create this dataset, we first need to filter out data with weak event correlations. The bottom half of Figure S1 shows an example of the data pipeline described in Figure 4. By inputting the event information into the model in text form, we have it determine whether each correlation is logically valid. After filtering, we obtained 200K reasoning data from 500K dense video captioning data.

Our dataset contains videos across 10 domains like travel, DIY, tech reviews, etc. (Figure S2). The average duration and the length of the caption also varied between different domains (Figure S3). Figure S4 presents the percentage of videos with temporal relevance in each category.

S2. Implementation Details

We fully fine-tuned the 3B model from Qwen2.5-VL checkpoint on two tasks, masked event prediction and event segmentation and temporal captioning, in two sequential stages:

- **Stage 1**: We trained the model using **masked event prediction** with supervised fine-tuning (SFT) on our VER dataset.
- **Stage 2**: We fine-tuned the checkpoints from Stage 1 on the **event segmentation and temporal captioning** task, utilizing our VER dataset to enhance temporal event understanding.

Our training configuration includes:

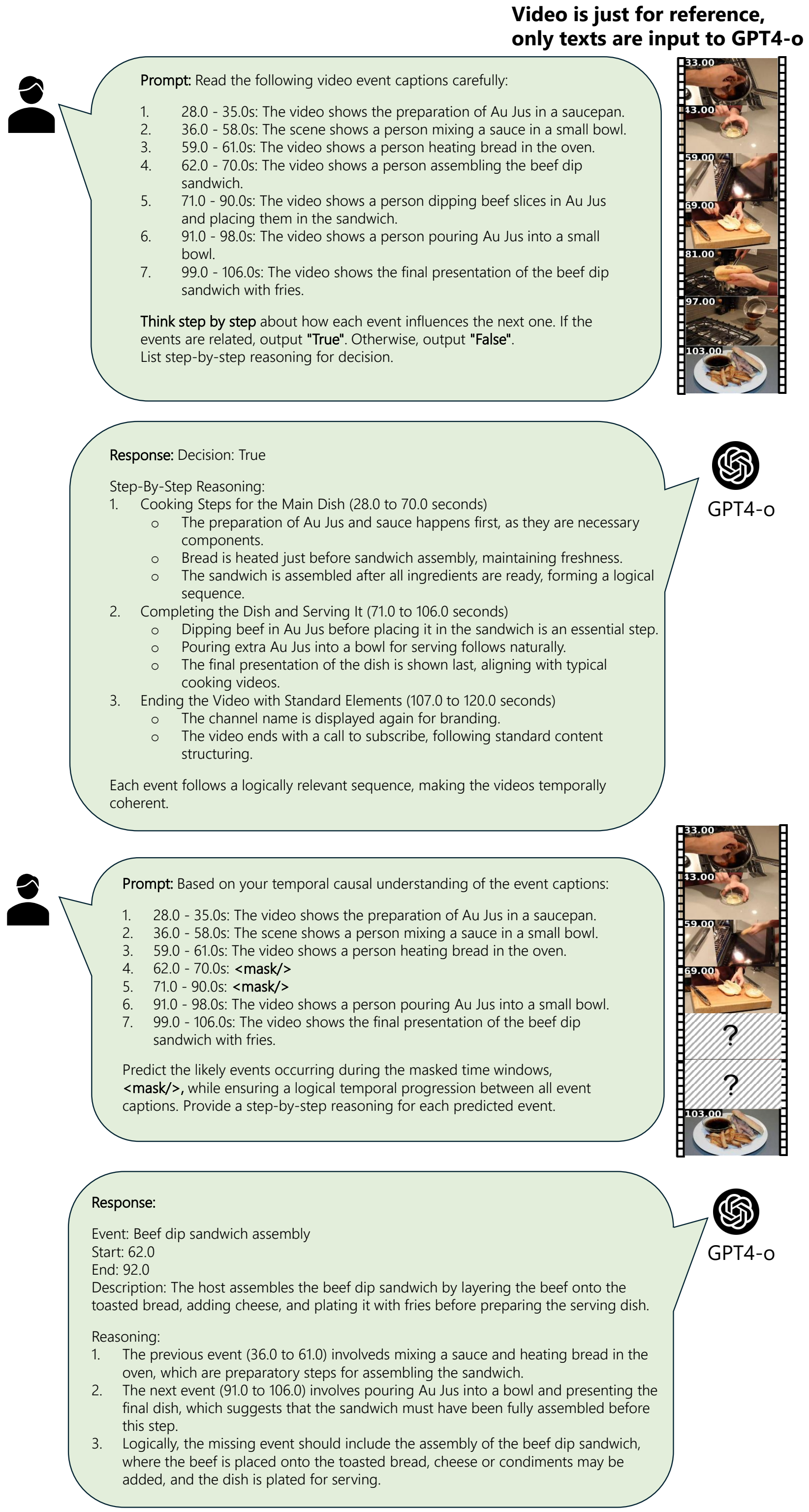
- Global batch size: 64, with 2 samples per device across 8 devices, resulting in gradient accumulation steps of 4
- Learning rates: 1×10^{-5} for the LLM and MLP adapter; 2×10^{-6} for the vision encoder
- Weight decay: 0.1 to prevent overfitting
- Warm-up ratio: 0.03 in the cosine learning rate schedule
- Gradient checkpoint: Enabled to reduce memory consumption
- Liger kernel integration [12]: Significantly reduced memory overhead during full fine-tuning, making it feasible to process long video input frames efficiently.

During training, we adopted a uniform sampling rate at 1 frame per second (FPS) and fixed every sampled frame to 320×180 pixels.

S3. Qualitative Analysis

Figure S5 and S6 compare the performance of Qwen2.5-VL-3B (our baseline), Grounded-Video-LLM-Phi, VideoQA, and TEMPURA (our proposed model) on long video temporal grounding tasks. The red text highlights errors in timestamp predictions when other models segment videos into fine-grained events and identify their start and end times. While other models often struggle, especially toward the end of long videos, TEMPURA consistently segments events accurately and assigns precise timestamps. For instance, the green text shows that TEMPURA correctly identifies a person filling and wrapping spring rolls from 161.00 to 183.00 seconds, followed by placing them into oil for frying from 185.00 to 205.00 seconds. Additionally, TEMPURA has better performance in producing more fine-grained event captions, as shown by the larger number of event captions produced by our model.

Figure S7 compares the performance of Qwen2.5-VL-3B (our baseline), Grounded-Video-LLM-Phi, VideoQA, and TEMPURA (our proposed model) on fine-grained video segmentation. The red highlights indicate errors in timestamp predictions and the failure of other models to produce detailed event captions, even in shorter videos. In contrast, TEMPURA demonstrates better performance, as indicated by the green text, by producing more accurate timestamps, fine-grained events, and descriptive event captions.



Video is just for reference, only texts are input to GPT4-o

Prompt: Read the following video event captions carefully:

1. 28.0 - 35.0s: The video shows the preparation of Au Jus in a saucepan.
2. 36.0 - 58.0s: The scene shows a person mixing a sauce in a small bowl.
3. 59.0 - 61.0s: The video shows a person heating bread in the oven.
4. 62.0 - 70.0s: The video shows a person assembling the beef dip sandwich.
5. 71.0 - 90.0s: The video shows a person dipping beef slices in Au Jus and placing them in the sandwich.
6. 91.0 - 98.0s: The video shows a person pouring Au Jus into a small bowl.
7. 99.0 - 106.0s: The video shows the final presentation of the beef dip sandwich with fries.

Think step by step about how each event influences the next one. If the events are related, output **"True"**. Otherwise, output **"False"**.
List step-by-step reasoning for decision.

Response: Decision: True

Step-By-Step Reasoning:

1. Cooking Steps for the Main Dish (28.0 to 70.0 seconds)
   - The preparation of Au Jus and sauce happens first, as they are necessary components.
   - Bread is heated just before sandwich assembly, maintaining freshness.
   - The sandwich is assembled after all ingredients are ready, forming a logical sequence.
2. Completing the Dish and Serving It (71.0 to 106.0 seconds)
   - Dipping beef in Au Jus before placing it in the sandwich is an essential step.
   - Pouring extra Au Jus into a bowl for serving follows naturally.
   - The final presentation of the dish is shown last, aligning with typical cooking videos.
3. Ending the Video with Standard Elements (107.0 to 120.0 seconds)
   - The channel name is displayed again for branding.
   - The video ends with a call to subscribe, following standard content structuring.

Each event follows a logically relevant sequence, making the videos temporally coherent.

Prompt: Based on your temporal causal understanding of the event captions:

1. 28.0 - 35.0s: The video shows the preparation of Au Jus in a saucepan.
2. 36.0 - 58.0s: The scene shows a person mixing a sauce in a small bowl.
3. 59.0 - 61.0s: The video shows a person heating bread in the oven.
4. 62.0 - 70.0s: **<mask/>**
5. 71.0 - 90.0s: **<mask/>**
6. 91.0 - 98.0s: The video shows a person pouring Au Jus into a small bowl.
7. 99.0 - 106.0s: The video shows the final presentation of the beef dip sandwich with fries.

Predict the likely events occurring during the masked time windows, **<mask/>,** while ensuring a logical temporal progression between all event captions. Provide a step-by-step reasoning for each predicted event.

Response:

Event: Beef dip sandwich assembly
Start: 62.0
End: 92.0
Description: The host assembles the beef dip sandwich by layering the beef onto the toasted bread, adding cheese, and plating it with fries before preparing the serving dish.

Reasoning:

1. The previous event (36.0 to 61.0) involveds mixing a sauce and heating bread in the oven, which are preparatory steps for assembling the sandwich.
2. The next event (91.0 to 106.0) involves pouring Au Jus into a bowl and presenting the final dish, which suggests that the sandwich must have been fully assembled before this step.
3. Logically, the missing event should include the assembly of the beef dip sandwich, where the beef is placed onto the toasted bread, cheese or condiments may be added, and the dish is plated for serving.

Fig. S1: Our mask event prediction data example and generation process.

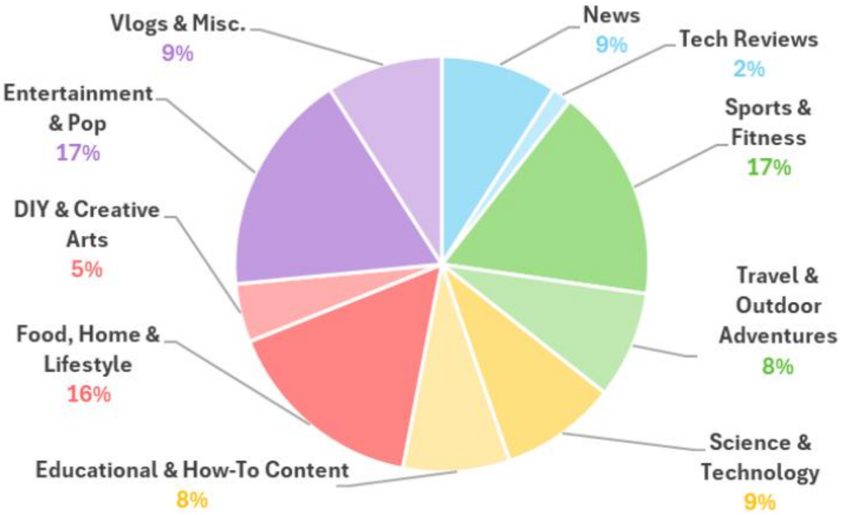


Fig. S2: Video Class Distribution

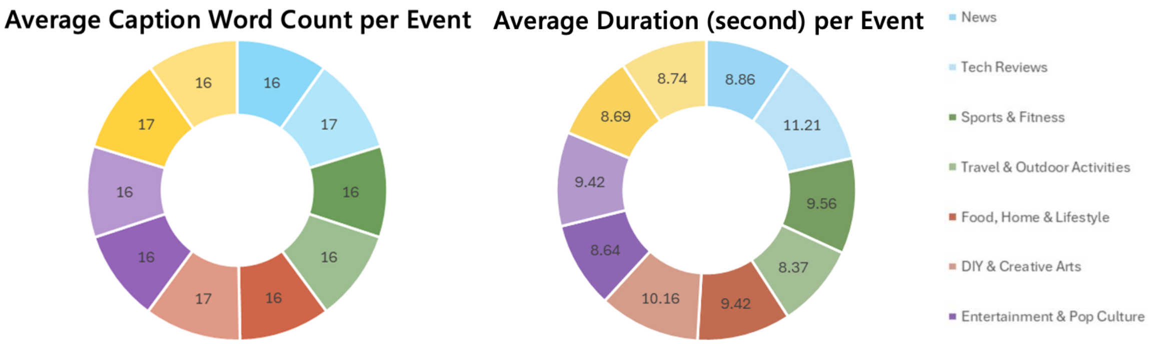


Fig. S3: Event Caption Distribution

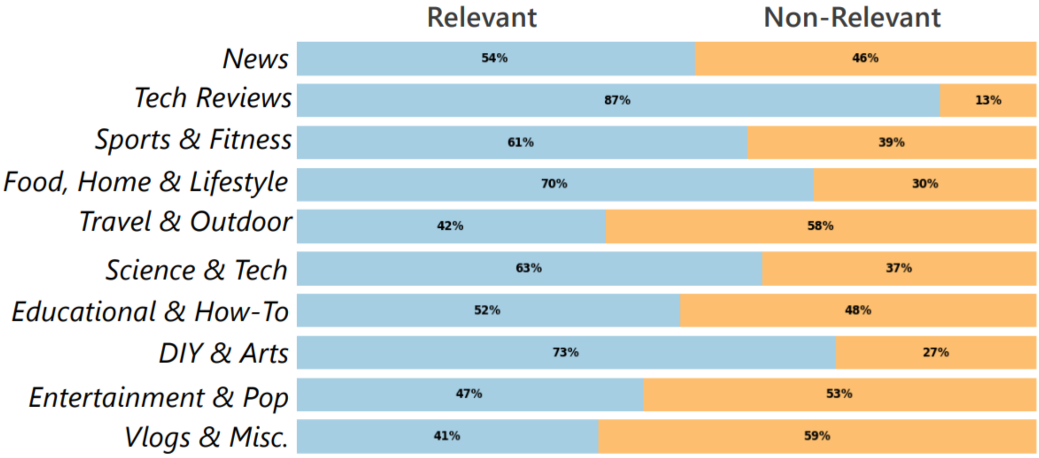
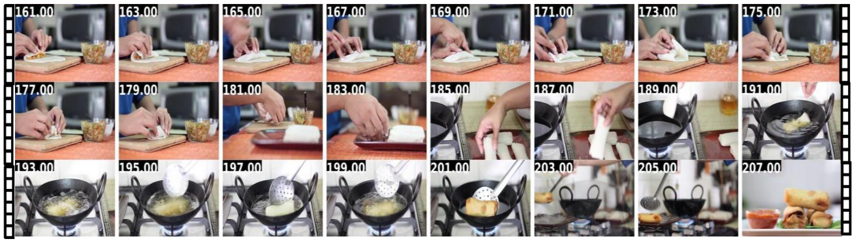


Fig. S4: Video Frame Temporal Relevance Distribution



Segment the video into a sequence of non-overlapping events by identifying the start and end times of each event. Ensure that all video frames are covered chronologically. Each event should be described in detail. Please follow the format below:

"From <start time1> to <end time1>, <detailed description1>."
"From <start time2> to <end time2>, <detailed description2>."
(Continue for all identified events)

1. From 14.0 to 31.0 seconds, mix corn flour and plain flour in a bowl.
2. From 32.0 to 65.0 seconds, add water and salt to the mixture and stir until smooth.
3. From 78.0 to 119.0 seconds, add oil, ginger, green bell peppers, onions, carrots, and cabbage to the pan and stir.
4. From 120.0 to 125.0 seconds, add salt and black pepper to taste.
5. From 126.0 to 165.0 seconds, place the filling on the wrapper and roll it up.
6. From 166.0 to 185.0 seconds, fry the spring rolls in oil until golden brown.

Qwen2.5-VL-3B

1. From 0 to 13 seconds, the video starts with a list of ingredients for making spring rolls.
2. From 13 to 31 seconds, the woman adds the ingredients to a bowl and mixes them together.
3. From 31 to 57 seconds, she adds the mixture to a pan and cooks it on the stove.
4. From 57 to 84 seconds, the woman adds oil to the pan and cooks the mixture.
5. From 84 to 102 seconds, she adds the spring onions to the pan.
6. From 102 to 120 seconds, she adds the garlic to the pan.
7. From 120 to 141 seconds, she adds the carrots and the green chili pepper to the pan.
8. From 141 to 161 seconds, she adds the cabbage to the pan.
9. From 161 to 181 seconds, she adds the salt and pepper to the pan.
10. From 181 to 194 seconds, she rolls the spring rolls up and places them on a plate.
11. From 194 to 223 seconds, the woman adds the spring onions to the spring rolls.
12. From 223 to 245 seconds, she rolls the spring rolls up and places them on the plate.
13. From 245 to 263 seconds, the woman cooks the spring rolls in the pan.
14. From 263 to 299 seconds, the woman takes the spring rolls out of the pan and places them on a plate.

Grounded Video LLM-Meta-Llama-3-8B-Instruct

Fig. S5: Comparison of **long video temporal grounding responses** on a cooking tutorial video, generated by baseline models: Qwen2.5-VL-3B and Grounded Video LLM-Meta-Llama-3-8B-Instruct. Each of the baseline models are prompted to segment videos and describe each of the video segments in detail with correct start and end times. The text highlighted in red indicates incorrect determination of start and end times for frame descriptions.

Grounded-Video-LLM-Phi

1. From 0 seconds, intro.
2. From 18 seconds, ingredients.
3. From 35 seconds, mixing the ingredients.
4. From 57 seconds, cooking the pancake.
5. From 111 seconds, sauteing the vegetables.
6. From 164 seconds, making the sauce.
7. From 184 seconds, rolling the pancake.
8. From 207 seconds, adding the vegetables.
9. From 245 seconds, frying the spring roll.
10. From 283 seconds, outro.

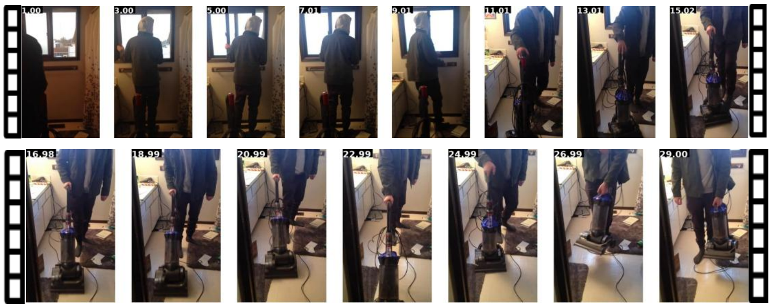
VideoQA

1. From 0 seconds, intro
2. From 18 seconds, ingredients.
3. From 37 seconds, mixing ingredients.
4. From 58 seconds, cooking pancake.
5. From 112 seconds, sauteing vegetables.
6. From 166 seconds, making sauce.
7. From 184 seconds, rolling out dough.
8. From 208 seconds, filling dough.
9. From 255 seconds, frying spring roll.
10. From 283 seconds, finished result.

TEMPURA

1. From 0.5 to 4.5 seconds, the video begins with an introduction to the recipe for vegetable spring rolls by Archana's Kitchen.
2. From 5.5 to 12.5 seconds, the ingredients for the spring roll wrapper are listed: 1/2 cup plain flour, 1/4 cup cornflour, 3/4 cup water or more to make a smooth batter, and a pinch of salt.
3. From 13.5 to 20.5 seconds, the ingredients are measured and mixed in a bowl.
4. From 21.5 to 27.5 seconds, water is added to the mixture and stirred until it forms a smooth batter.
5. From 28.5 to 39.5 seconds, the batter is left to rest while other ingredients are prepared.
6. From 40.5 to 50.5 seconds, oil is heated in a pan and the batter is poured into the pan to form a pancake-like shape.
7. From 51.5 to 69.5 seconds, the pancake is flipped over to cook the other side.
8. From 70.5 to 80.5 seconds, the ingredients for the spring rolls are listed: 10 to 12 spring roll wrappers, shredded cabbage, shredded carrots, chopped green bell peppers, grated ginger, salt, black pepper, flour paste, and oil for deep frying.
9. From 81.5 to 111.5 seconds, the vegetables are cooked in a pot with oil, ginger, and other seasonings.
10. From 112.5 to 121.5 seconds, the cooked vegetables are mixed with flour paste to create a dough-like consistency.
11. From 122.5 to 131.5 seconds, the dough is prepared for rolling into spring rolls.
12. From 132.5 to 183.5 seconds, the dough is rolled into spring rolls and filled with the cooked vegetable mixture.
13. From 184.5 to 204.5 seconds, the filled spring rolls are deep-fried until golden brown.
14. From 205.5 to 218.5 seconds, the finished spring rolls are served with dipping sauce.

Fig. S6: Comparison of **long video temporal grounding responses** on a cooking tutorial video, generated by TEMPURA (our model) and baseline models: Grounded-Video-LLM-Phi and VideoQA. Each of the models are prompted to segment videos and describe each of the video segments in detail with correct start and end times. The text highlighted in red indicates incorrect determination of start/end times and video segment descriptions. The text in green indicates correct determination of start/end times and and video segment descriptions.



Segment the video into a sequence of non-overlapping fine-grained events by identifying the start and end times of each event in seconds. Ensure that all video frames are covered chronologically. Each event should be described in detail, capturing the specific actions taking place, the objects involved, and any scene transitions. Follow the format below:
Format:
"From <start time1> to <end time1>, <detailed description 1>."
"From <start time2> to <end time2>, <detailed description 2>."
(Continue for all identified events)

Qwen2.5-VL-3B

1. From 0.0 to 3.8 seconds, a person is standing in front of a window, looking outside.

Grounded-Video-LLM-Meta-Llama-3-8B-Instruct

1. From 0 to 46 seconds, a man is looking out of a window.
2. From 46 to 94 seconds, he turns around and looks back in the window.
3. From 94 to 112 seconds, he bends down and picks up a vacuum cleaner.
4. From 112 to 299 seconds, he vacuums the floor.

Grounded-Video-LLM-Phi

1. From 0 seconds, intro.
2. From 18 seconds, showing vacuum.
3. From 88 seconds, explaining vacuum.
4. From 139 seconds, demonstrating vacuum.
5. From 269 seconds, outro.

VideoQA

1. From 0 seconds, intro.
2. From 19 seconds, showing the vacuum.
3. From 88 seconds, explaining the vacuum.
4. From 145 seconds, demonstrating the vacuum.
5. From 250 seconds, outro.

TEMPURA

1. From 0.5 to 6.5 seconds, a person wearing a hoodie is seen standing near a window, adjusting the blinds.
2. From 7.5 to 13.5 seconds, the person moves closer to the vacuum cleaner, preparing to use it.
3. From 14.5 to 20.5 seconds, the person starts using the vacuum cleaner, moving it around the room.
4. From 21.5 to 27.5 seconds, the person continues to use the vacuum cleaner, cleaning different areas of the room.
5. From 28.5 to 29.5 seconds, the person finishes using the vacuum cleaner and stands near it.

Fig. S7: Comparing TEMPURA (our model) and other baseline model (Qwen2-5-VL-3B, Grounded-Video-LLM-Meta-Llama-3-8B-Instruct, Grounded-Video-LLM-Phi, and VideoQA) abilities to generate detailed descriptions on **fine-grained events on short videos**. Each model is prompted to segment the video into fine-grained events and describe the events in detail with correct start/end timestamps. Red text indicates incorrect responses with incorrect start/end timestamps and/or poor descriptions of the event segment.