TemporalVLM: Video LLMs for Temporal Reasoning in Long Videos

Fawad J. Fateh[†] **Umer Ahmed**[†] **Hamza Khan** **M. Zeeshan Zia** **Quoc-Huy Tran**
Retrocausal, Inc.
Redmond, WA
www.retrocausal.ai

Abstract

We introduce TemporalVLM, a video large language model (video LLM) for temporal reasoning and fine-grained understanding in long videos. Our approach includes a visual encoder for mapping a long-term video into features which are time-aware and contain both local and global cues. It first divides an input video into short-term clips, which are jointly encoded with timestamps and fused across overlapping temporal windows into time-sensitive local features. Next, the local features are passed through a bidirectional long short-term memory (BiLSTM) module for global feature aggregation. Moreover, to facilitate the evaluation of TemporalVLM, we present a large-scale long video dataset of industry assembly processes, namely IndustryASM, consisting of videos recorded on factory floors with actions and timestamps annotated by industrial engineers for time and motion studies and temporal action segmentation evaluation. Finally, extensive experiments show that TemporalVLM outperforms previous methods across temporal reasoning and fine-grained understanding tasks, i.e., dense video captioning, temporal video grounding, video highlight detection, and temporal action segmentation.

1 Introduction

Video temporal reasoning represents the process of reasoning about time and its passage in videos, with a focus on how events or actions happen and relate to each other in terms of time. Several video understanding applications require the ability of temporal reasoning, e.g., dense video captioning (Wang et al., 2021; Zhu et al., 2022; Yang et al., 2023) and temporal action segmentation (Li et al., 2020; Yi et al., 2021; Hyder et al., 2024; Lu and Elhamifar, 2024). Considerable efforts (Yang et al., 2023; Luo et al., 2023a; Moon et al., 2023; Zhong et al., 2023; Lu and Elhamifar, 2024; Lin et al., 2024) have been invested in developing models for solving individual tasks. These models often have different architectures. Thus, it is favorable to design a unified model for handling various tasks.

The past few years have witnessed the impressive comprehension and generation capability of LLMs (Achiam et al., 2023; Chiang et al., 2023; Taori et al., 2023; OpenAI, 2024; Touvron et al., 2023a,b), which have emerged as a universal agent for performing various tasks. Video LLMs (Li et al., 2023b; Liu et al., 2023; Luo et al., 2023b; Maaz et al., 2023; Zhang et al., 2023; Jin et al., 2024; Song et al., 2024) which incorporate video encoders with LLMs have been introduced. These methods often represent a video by a fixed number of tokens, yielding reduced performance with long videos, and encode frames and timestamps separately and hence struggle with temporal reasoning tasks. Please refer to the inferior results of these methods in Tab. 1. Recently, a few works (Qian et al., 2024; Huang et al., 2024a,b; Ren et al., 2024) have developed video LLMs with temporal reasoning abilities, e.g., TimeChat (Ren et al., 2024) proposes to vary the number of tokens based on the video length and jointly encode frames and timestamps. The above methods usually treat the entire video as a single clip (Qian et al., 2024; Huang et al., 2024a,b; Ren et al., 2024) and aggregate tokens via pooling operation (Huang et al., 2024b) and query aggregation (Ren et al., 2024), struggling to capture fine-grained details in long videos.

We propose TemporalVLM, a video LLM for temporal reasoning and fine-grained understanding in long videos. Fig. 1 shows the comparisons with prior works. Our model includes two architectural contributions. Firstly, we divide a long-term input video into multiple short-term clips and introduce a time-sensitive clip encoder for extracting fused time-aware local features from each clip. Secondly, we adopt a BiLSTM module which takes all the lo-

[†] indicates joint first author.
{fawad,umer,hamza,zeeshan,huy}@retrocausal.ai.

Findings of the Association for Computational Linguistics: ACL 2026, pages 1427–1447

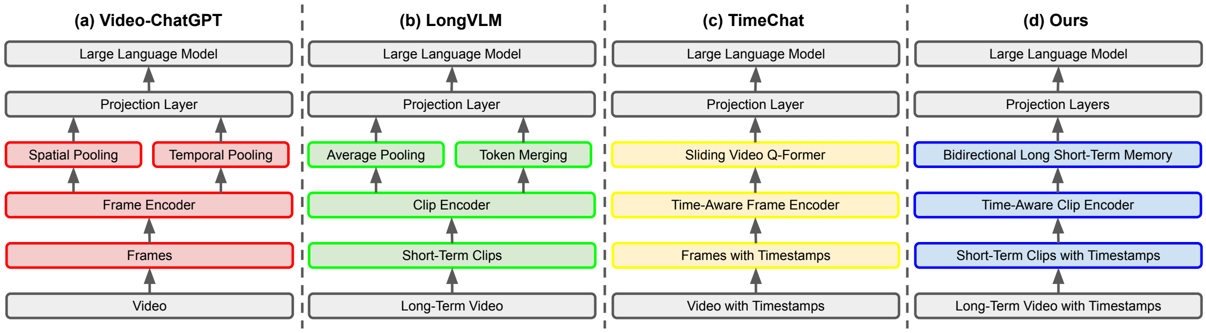


Figure 1: Video LLMs are often not time-sensitive (a, b), consider an input video as a single clip (a, c), and apply pooling (a, b) or query aggregation (c) for aggregating global features. Our model (d) includes a time-aware clip encoder for extracting time-aware fine-grained cues and a BiLSTM for capturing long-range temporal dependencies.

cal features as inputs and computes global features from multiple clips. Our features not only are time-sensitive but also contain both local fine-grained and global semantic information, which are crucial for temporal reasoning in long videos. Moreover, to further evaluate our model, we present IndustryASM, a large-scale long video dataset of industry assembly processes for temporal action segmentation benchmarking and time and motion studies. Our IndustryASM dataset comprises of 4851 videos with an average video duration of 105 seconds. It covers in total 47 diverse industry assembly tasks and includes timestamp and action labels. We convert the labels into chat samples with manually written instructions. Lastly, extensive experiments demonstrate that TemporalVLM achieves superior results over previous methods.

In summary, our contributions include:

- We develop the first time-aware coarse-to-fine encoder, including a time-aware clip encoder (i.e., overlapping sliding video Q-Former) and a BiLSTM. By leveraging the modules, we tackle both fine-grained understanding and temporal reasoning in long videos.
- We present IndustryASM, a large-scale long video dataset of manufacturing assembly procedures. IndustryASM can be downloaded at `https://retrocausal.ai/research/`.
- TemporalVLM outperforms prior works on temporal reasoning and fine-grained understanding. To our best knowledge, this is the first work to blend LSTMs into video LLMs.

2 Related Work

Video Large Language Models. Video LLMs (Li et al., 2023b; Liu et al., 2023; Luo et al., 2023b; Maaz et al., 2023; Zhang et al., 2023; Jin et al., 2024; Song et al., 2024) typically include a pre-trained visual encoder to extract visual features, a projection layer to map visual features into the text latent space of LLMs, and a pre-trained LLM for generating responses. They mostly differ in the visual encoder. VideoChat (Li et al., 2023b) extracts frame features via a visual transformer (Sharir et al., 2021) and employs a query transformer (Q-Former) (Li et al., 2023a) to aggregate frame features into video features, while a video Q-Former is further included for temporal modeling in Video-LLaMA (Zhang et al., 2023). The above methods usually map the video into a fixed number of tokens, yielding degrading performance with long videos, while encoding frames and timestamps separately and hence struggling with temporal reasoning. To address these drawbacks, methods with temporal reasoning capabilities are introduced, e.g., TimeChat (Ren et al., 2024) presents a sliding video Q-Former to handle various video lengths and a time-aware frame encoder to jointly encode frames and timestamps. These methods often consider the video as a single clip (Luo et al., 2023b; Maaz et al., 2023; Li et al., 2023b; Zhang et al., 2023) and aggregate tokens via pooling (Luo et al., 2023b; Maaz et al., 2023) and query aggregation (Li et al., 2023b; Zhang et al., 2023), overlooking fine-grained details. In this work, we divide the video into multiple clips and propose a time-aware clip encoder for capturing fused local features. Also, we integrate a BiLSTM module for aggregating global features.

Video Temporal Reasoning. Temporal reasoning plays an important role in video understanding tasks, e.g., dense video captioning (Wang et al., 2021; Zhu et al., 2022; Yang et al., 2023), temporal video grounding (Wang et al., 2022; Luo et al., 2023a), video highlight detection (Lei et al., 2021;

Moon et al., 2023), and temporal action segmentation (Li et al., 2020; Yi et al., 2021; Mahmood et al., 2026; Lu and Elhamifar, 2024). Prior works (Yang et al., 2023; Luo et al., 2023a; Moon et al., 2023; Lu and Elhamifar, 2024) often focus on designing separate models for tackling individual tasks, while LLM-based models (Qian et al., 2024; Huang et al., 2024a,b; Ren et al., 2024) capable of handling multiple tasks have emerged recently. Our TemporalVLM model belongs to the second group.

Long Video Understanding. Challenges in long video understanding include complex spatial-temporal relationships and redundant information. Long video understanding methods have been developed via efficient architectures (Donahue et al., 2015; Kondratyuk et al., 2021), temporal pooling/aggregation (Sener et al., 2020; Wu and Krahenbuhl, 2021), and clip selection (Ghodrati et al., 2021; Gowda et al., 2021). For vision-language understanding tasks, methods based on temporal alignment (Han et al., 2022; Buch et al., 2022) and memory (Wu et al., 2022; Zhao et al., 2023) have been introduced. Recently, LongVLM (Weng et al., 2024) divides the video into clips and employs a merging module for extracting local features and pooling operation for computing global features. It does not utilize timestamps, which are crucial for temporal reasoning. Our TemporalVLM model explicitly utilizes timestamps via a time-aware clip encoder. Moreover, we employ a learnable BiLSTM module for aggregating global features.

Procedural Activity Datasets. Existing datasets usually focus on cooking activities, e.g., Epic-Kitchens (Damen et al., 2018), are curated from online sources and hence produced with multiple shots, e.g., COIN (Tang et al., 2019), and are recorded with toy objects and lab environments, e.g., Assembly101 (Sener et al., 2022). Our IndustryASM dataset focuses on manufacturing assembly processes, is recorded on factory floors, and is labeled by industrial engineers, thereby capturing procedural activities in realistic and practical environments that current datasets have not addressed.

3 TemporalVLM

An overview of TemporalVLM is shown in Fig. 2.

3.1 Short-Term Temporal Reasoning

Video LLMs often consider an input video as a single clip and sample a fixed number of frames N_f^v from the video, e.g., $N_f^v = 96$ frames in TimeChat (Ren et al., 2024). Moreover, prior works usually use query aggregation, e.g., TimeChat (Ren et al., 2024), or pooling operation, e.g., Video-ChatGPT (Maaz et al., 2023), to aggregate video tokens. Thus, they perform well with short videos but miss out fine-grained information for long videos.

We introduce a time-aware clip encoder for extracting fused local fine-grained cues within a clip. We first divide the long-term input video into $C = 6$ short-term clips and sample $N_f^c = 96$ from each clip. Sampled frames of a clip along with timestamps are then passed to our time-aware clip encoder for jointly encoding frame contents and timestamps, yielding fused time-aware local features. Particularly, the time-aware frame encoder uses a pre-trained image encoder (Sun et al., 2023) to obtain frame features, which are jointly encoded with timestamps via an image Q-Former (Dai et al., 2023), yielding time-aware frame features $\mathbf{f}_t$.

Overlapping Sliding Video Q-Former. Frame features $\mathbf{f}_t$ are fed to a video Q-Former (Ren et al., 2024) across **overlapping** windows $\mathbf{W}_i$ of size q = 32 frames and overlap o = 16 frames. Window-wise outputs $\mathbf{V}_i$ are concatenated into $\mathbf{S}$ as:

$$\mathbf{V}_i = \text{Video Q-Former}(\mathbf{W}_i), \quad (1)$$

$$\mathbf{S} = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_W], \quad (2)$$

with W denoting the number of overlapping windows within a clip. We propose a **fusion** module to align diverse temporal cues from redundant boundary tokens in $\mathbf{S}$. Specifically, we apply multi-headed self-attention on $\mathbf{S}$, yielding $\mathbf{C}$ which fuses local contexts across multiple windows into a single context-aware embedding as:

$$\mathbf{C}^{(h)} = \text{SoftMax}\left(\frac{\mathbf{Q}^{(h)}\mathbf{K}^{(h)}}{\sqrt{d}}\right)\mathbf{V}^{(h)}, \quad (3)$$

$$\mathbf{C} = [\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \ldots, \mathbf{C}^{(H)}]\mathbf{W}^O. \quad (4)$$

Here, we first project $\mathbf{S}$ into $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ for each attention head h. At each head h, $\mathbf{C}^{(h)}$ represents the dot product attention between queries $\mathbf{Q}^{(h)}$ and keys $\mathbf{K}^{(h)}$ scaled over dimension d followed by softmax weighted aggregation over $\mathbf{V}^{(h)}$. $\mathbf{C}$ contains aggregated outputs from all heads with final projection $\mathbf{W}^O$ applied. Using overlapping windows leads to spatially redundant time-aware tokens in $\mathbf{S}$, each with different local window contexts. This allows $\mathbf{C}$ to yield rich time-aware clip-level information by leveraging diverse temporal views from across overlapping windows. TimeChat's (Ren

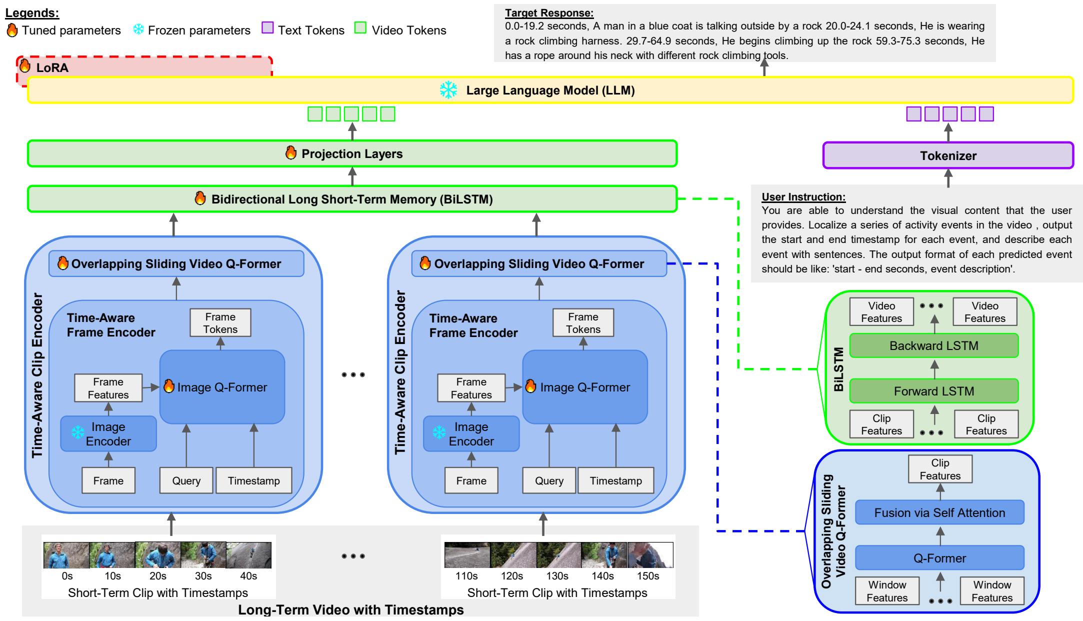


Figure 2: TemporalVLM includes two novel components: a time-aware clip encoder for extracting time-aware fine-grained cues and a BiLSTM module for capturing long-range temporal dependencies.

et al., 2024) video Q-Former uses non-overlapping windows and does not perform fusion, yielding inferior results as shown in Tab. 4.

3.2 Long-Term Temporal Reasoning

Our time-sensitive clip encoder is applied on each short-term clip to obtain fused time-aware local features, which capture useful fine-grained cues for temporal reasoning within the clip. However, they are not effective for temporal reasoning over the long-term video, which requires the ability to capture long-range temporal dependencies.

Bidirectional Long Short-Term Memory. We introduce a BiLSTM module for computing global features across multiple clips. We first concatenate time-aware local features extracted from clips in the temporal order the clips appear in the video. We then pass the sequence of local features to a BiLSTM for aggregating global features. Our BiLSTM follows a standard architecture, including two LSTM networks: one processes the sequence in the original order (forward) and another processes the sequence in the reverse order (backward) as:

$$\mathbf{h}_t^f = \mathrm{LSTM}(\mathbf{h}_{t-1}^f, \mathbf{c}_t), \tag{5}$$

$$\mathbf{h}_t^b = \mathrm{LSTM}(\mathbf{h}_{t+1}^b, \mathbf{c}_t), \tag{6}$$

where $\mathbf{h}_t^f$ and $\mathbf{h}_t^b$ denote the hidden states at time step t of the forward and backward LSTMs respectively and $\mathbf{c}_t$ is the input at time step t. The final output $\mathbf{h}_t$ at time step t is obtained by concatenating the outputs of the forward and backward LSTMs as $\mathbf{h}_t = [\mathbf{h}_t^f, \mathbf{h}_t^b]$. Please refer to the textbook (Learning, 2016) for a detailed description. Our BiLSTM module utilizes both past and future information and is capable of capturing long-range temporal relationships in both forward and backward directions. As observed in Tab. 5, BiLSTM outperforms various alternatives, including average pooling, linear layer, LSTM, and transformer (Vaswani et al., 2017). Unlike LongVLM (Weng et al., 2024), which applies pooling to obtain global features, our BiLSTM module has learnable parameters and achieves better results. Advanced recurring models, e.g., state space (Gu et al., 2021) and Mamba (Gu and Dao, 2024), may further improve the performance, which remains our future work.

The video tokens output by our BiLSTM module are of size $(C \times N_f^c,\ 2 \times N_V)$, while the LLM requires an input size of $(N_f^c,\ N_{LLM})$. N_V and N_{LLM} are the dimensions of the video tokens and the LLM latent space respectively, and 2 is to account for the outputs of both forward and backward passes of BiLSTM. Thus, we pass the video tokens output by BiLSTM through projection layers to match the input dimensions required by the LLM.

3.3 Large Language Model

The LLM takes as input the video tokens $\mathbf{X}^v$, query tokens $\mathbf{X}^q$, and generates responses $\mathbf{X}^r$ to users. Video LLMs typically employ a two-stage training: leveraging large-scale image/video-text pairs for vision-language alignment to pre-train the model, and utilizing instruction data to fine-tune the pre-trained model. Following TimeChat (Ren et al., 2024), we use the checkpoint of the LLaMA-2 7B model (Touvron et al., 2023b) and perform instruction tuning only. We optimize the below loss:

$$\mathcal{L} = -\log P_{\psi}(\mathbf{X}^r|\mathbf{X}^v, \mathbf{X}^q) \tag{7}$$

$$= -\sum_{i=1}^{L} \log P_{\psi}(x_i|\mathbf{X}^v, \mathbf{X}^q, \mathbf{X}^r_{<i}). \tag{8}$$

Here, ψ is the learnable parameters of TemporalVLM, L is the response length, x_i is the current predicted token, and $\mathbf{X}^r_{<i}$ is the prior tokens appearing before x_i in the response.

4 IndustryASM

4.1 Dataset Statistics

IndustryASM includes 4851 videos with each lasting 105 seconds on average, yielding 142 hours as the total dataset duration. In addition, it consists of 47 various industry assembly procedures with each having 12 steps on average. Example videos are shown in Fig. 3. Due to space limits, we provide additional details in our supplementary material.

4.2 Annotation Process

Our videos are annotated by industrial engineers into framewise action segments. The action labels contain the same step naming conventions as used on factory floors during manufacturing processes. To ensure quality, each video is annotated by two labelers, i.e., one obtains the labels for the video, while another checks the labels. In cases of conflicts, both labelers discuss to address them. Overall, around 8% of the videos have conflicts and require fixing, yielding an agreement rate of 92% between labelers. Given the framewise annotations we first convert them to timestamps by using the frame rate of the video. We then manually write the instructions designed for the action segmentation task to ensure quality and generate the answers by using the timestamps and action labels. Examples are included in our supplementary material.

5 Experiments

Tasks and Datasets. We evaluate TemporalVLM on dense video captioning, temporal video grounding, video highlight detection, and temporal action segmentation. For fair comparisons with TimeChat (Ren et al., 2024), we fine-tune our model on a subset[1] of TimeIT (Ren et al., 2024) and Valley (Luo et al., 2023b). The combined dataset contains 142K long videos with diverse activities. We follow TimeChat to use 6 instructions (generated by GPT-4 (Achiam et al., 2023)) per task and convert video annotations into chat samples. We evaluate on YouCook2 (Zhou et al., 2018) for dense video captioning, Charades-STA (Gao et al., 2017) for temporal video grounding, QVHighlights (Lei et al., 2021) for video highlight detection, and IndustryASM for temporal action segmentation. Note that only Tab. 3 uses **our** dataset, all the remaining evaluations use **existing** datasets. Following prior works, we report the results from a single run.

Dataset Splits For dense video captioning, YouCook2 (Zhou et al., 2018) is divided into 1333 videos for fine-tuning out of a total of 1790 videos, while 457 videos are used for evaluation. For the temporal video grounding task, we divide the 16124 video-query pairs of Charades-STA (Gao et al., 2017) into 12404 fine-tuning pairs and 3720 evaluation pairs. For video highlight detection, 6858 videos are used for fine-tuning and 1463 are used for evaluation from the QVHighlights (Lei et al., 2021) dataset. These are the default splits provided by each dataset. Finally, we divide our IndustryASM dataset into 3896 videos for fine-tuning and 955 videos for evaluation.

Implementation Details. We follow TimeChat (Ren et al., 2024) to use LLaMA-2 7B (Touvron et al., 2023b) as the LLM, and ViT-G/14 from EVA-CLIP (Sun et al., 2023) as the image encoder. We take InstructBLIP's (Dai et al., 2023) checkpoints to initialize the image Q-Former, and Video-LLaMA's (Zhang et al., 2023) checkpoints to initialize the video Q-Former. We employ pyTorch's (Imambi et al., 2021) multiheaded attention in the fusion module. The BiLSTM module includes forward and backward LSTMs with one hidden layer each. The BiLSTM and projection layers are initialized randomly, while the LLM and image encoder remain frozen throughout fine-tuning. The LLM is fine-tuned using LoRA (Hu et al., 2023). We fine-tune our model on an 8-A100 (80 GB) machine. Please see our supplementary material for more details.

Competing Methods. We compare our model against VideoChat-Embed (Li et al., 2023b), Video-LLaMA (Zhang et al., 2023), Valley (Luo

[1]We could not download YT-Temporal (Zellers et al., 2022) due to its large size and the restricted number of downloads.

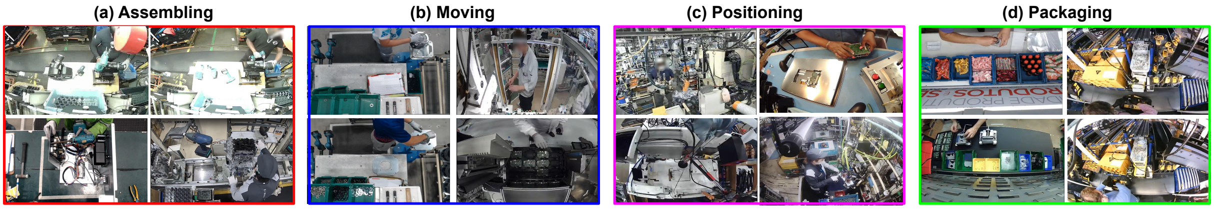


Figure 3: Example IndustryASM videos with different camera viewpoints, actors, backgrounds, and activities.

et al., 2023b), TimeChat (Ren et al., 2024), and LongVLM (Weng et al., 2024). Since LongVLM is not trained on TimeIT (Ren et al., 2024) and Valley (Luo et al., 2023b) and it is a multi-model pipeline (TimeChat and TemporalVLM are end-to-end), we replace our time-aware clip encoder and BiLSTM with their local and global feature extraction modules respectively while keeping the same image encoder and LLM, yielding our reimplemented LongVLM for evaluation. Nevertheless, we compare with the original LongVLM on general video understanding in our supplementary material.

Evaluation Metrics. We use the below metrics:

- Dense video captioning: We use SODA_c (Fujita et al., 2020) for story evaluation, F1 Score for event localization, and CIDEr (Vedantam et al., 2015) for measuring caption quality.
- Temporal video grounding: We use $R@1_{(IoU=\alpha)}$, which is the percentage of predictions with an intersection over union greater than α compared to the ground truth.
- Video highlight detection: We use mean average precision (mAP) and HIT@1 (IoU= 0.75) to evaluate saliency scores of key moments.
- Temporal action segmentation: We use frame-wise accuracy, F1 scores at overlapping thresholds $\{0.1, 0.25, 0.5\}$, and Edit distance.

5.1 Zero-Shot Results

Dense Video Captioning Results. Tab. 1 shows that TemporalVLM achieves the best results across all metrics. For example, we outperform TimeChat (Ren et al., 2024) by **+0.3** on CIDEr and **+0.5** on F1 Score. This is likely because TimeChat treats the entire video as a single clip and aggregates video tokens via query aggregation, overlooking fine-grained information. In contrast, our extracted features, which capture both local fine-grained cues from each clip and global temporal dependencies across multiple clips, are effective for caption generation and event localization. Our approach also obtains superior results than LongVLM (Weng et al., 2024) across SODA_c (**+0.4**), CIDEr (**+1.2**), and F1 Score (**+0.8**), which confirms the importance of our time-aware clip encoder and BiLSTM for dense video captioning.

Video Highlight Detection Results. From Tab. 1, TemporalVLM obtains the best performance on both mAP and HIT@1. It surpasses TimeChat (Ren et al., 2024) by **+1.9** in mAP and **+7.4** in HIT@1, while outperforming LongVLM (Weng et al., 2024) by **+5.3** at mAP and by **+16.3** at HIT@1. These results suggest the effectiveness of our model for video highlight detection. By aggregating global features from time-aware local features, it identifies the most relevant timestamps and assigns them saliency scores based on their relative importance within the video for a given query. Though LongVLM extracts multi-level features, the lack of a time-aware encoder hinders its results.

Temporal Video Grounding Results. It is evident from Tab. 1 that TemporalVLM achieves the best performance across $R@1_{(IoU=0.5)}$ and $R@1_{(IoU=0.7)}$. It outperforms TimeChat (Ren et al., 2024) by **+2.2** and **+0.9** on $R@1_{(IoU=0.5)}$ and $R@1_{(IoU=0.7)}$ respectively, and LongVLM (Weng et al., 2024) by **+16.2** and **+7.1** on $R@1_{(IoU=0.5)}$ and $R@1_{(IoU=0.7)}$ respectively. The results validate the benefits of our time-aware and multi-level features for temporal video grounding. Our clip encoder extracts fine-grained cues for temporal reasoning within each clip via fusion, which enables our BiLSTM to capture long-range temporal relationships across clips to detect the given event effectively. The lack of a time-aware encoder affects LongVLM, while TimeChat suffers from a lack of fine-grained cues.

5.2 Supervised Results

Dense Video Captioning, Video Highlight Detection, and Temporal Video Grounding Results. Tab. 2 presents the results in the supervised setting. TemporalVLM performs the best across all tasks and metrics. As compared to the zero-shot results in Tab. 1, we see an increase in performance, e.g.,

Model	Dense Captioning YouCook2			Highlight Detection QVHighlights		Temporal Grounding Charades-STA	
	SODA_c	**CIDEr**	**F1**	**mAP**	**HIT@1**	**R@1 (IoU=0.5)**	**R@1 (IoU=0.7)**
Valley (Luo et al., 2023b)	0.1	0.0	1.5	10.9	15.2	4.7	1.6
Video-LLaMA (Zhang et al., 2023)	0.0	0.0	0.1	11.3	15.6	2.7	1.2
VideoChat-Embed (Li et al., 2023b)	0.2	0.6	3.4	13.1	18.1	3.2	1.4
LongVLM (Weng et al., 2024)	<u>0.8</u>	2.5	12.3	11.1	15.0	13.9	6.1
TimeChat (Ren et al., 2024)	**1.2**	<u>3.4</u>	<u>12.6</u>	<u>14.5</u>	<u>23.9</u>	<u>27.9</u>	<u>12.3</u>
TemporalVLM	**1.2**	**3.7**	**13.1**	**16.4**	**31.3**	**30.1**	**13.2**

Table 1: Zero-short results. We test on YouCook2 for dense video captioning, QVHighlights for video highlight detection, Charades-STA for temporal video grounding. Best results are in **bold**, second best ones are <u>underlined</u>.

+24.3 on R@1$_{(\text{IoU}=0.5)}$ in temporal video grounding and **+9.5** on CIDEr in dense video captioning. TemporalVLM also outperforms more recent long video understanding models, such as Gen-S (Yao et al., 2025), by **+15.7** and **+13.8** at R@1$_{(\text{IoU}=0.5)}$ and R@1$_{(\text{IoU}=0.7)}$, respectively. Furthermore, TemporalVLM performs better than newer temporal reasoning methods, i.e., NumPro-FT (Wu et al., 2025) and LLAVA-ST (Li et al., 2025). Our model shows an improvement of **+12.4** and **+8.5** compared to NumPro-FT and **+9.6** and **+5.6** compared to LLAVA-ST at R@1$_{(\text{IoU}=0.5)}$ and R@1$_{(\text{IoU}=0.7)}$, respectively. However, we acknowledge that task-specific models, e.g., MMN (Wang et al., 2022) and QD-DETR (Moon et al., 2023), may outperform generalist models, e.g., TimeChat (Ren et al., 2024) and TemporalVLM, due to their task-specific designs. Due to space limits, we provide a comparison with task-specific models in our supplementary material.

Temporal Action Segmentation Results. We now evaluate on our IndustryASM dataset. Since the activities in IndustryASM significantly differ from those in TimeIT (Ren et al., 2024) and Valley (Luo et al., 2023b), we fine-tune all models on IndustryASM before evaluation. Tab. 3 presents the results. TemporalVLM performs the best across all metrics. For example, it outperforms TimeChat (Ren et al., 2024) by **+23.5** and **+6.9** on Edit and Acc respectively, and LongVLM (Weng et al., 2024) by **+8.9** and **+5.3** on Edit and Acc respectively. The results confirm the benefits of our time-aware and multi-level features.

Multi-Clip Encoder vs. Time-Aware Encoder. Multi-clip encoders in LongVLM (Weng et al., 2024) and TemporalVLM are crucial to capturing fine-grained cues, leading to superior results over TimeChat (Ren et al., 2024) on temporal action segmentation in Tab. 3. In contrast, time-aware encoders in TimeChat and TemporalVLM are important to event localization and temporal modeling, yielding better results than LongVLM on dense video captioning, video highlight detection, and temporal video grounding in Tabs. 1 and 2. TemporalVLM is both multi-clip and time-aware.

5.3 Ablation Results

We fine-tune several variants of TemporalVLM on YouCook2 (Zhou et al., 2018) to study our overlapping sliding video Q-Former and BiLSTM.

Impacts of Overlapping Sliding Video Q-Former. Tab. 4 demonstrates the importance of fusing encoded frames across overlapping temporal windows in our clip encoder. Our model with fusion across overlapping windows achieves the best results, whereas using non-overlapping windows and removing the fusion module as in TimeChat's (Ren et al., 2024) design lead to performance drops. The results validate our design.

Impacts of BiLSTM. To study the effects of BiLSTM, we replace it with various alternatives and report the results in Tab. 5. From Tab. 5, BiLSTM outperforms average pooling, which is not trained and employed in prior works (Maaz et al., 2023; Weng et al., 2024), and traditional LSTM, which only performs the forward pass and hence relies only on past cues. In contrast, BiLSTM is learnable and conducts both forward and backward passes to utilize both past and future cues. Moreover, we replace BiLSTM with linear layer to examine the impacts of video Q-Formers (Zhang et al., 2023; Ren et al., 2024) alone, leading to worse results. This highlights the importance of BiLSTM even when video Q-Formers are being used. Finally, BiLSTM outperforms transformer (Vaswani et al., 2017), which we will discuss below.

(Bi)LSTM vs. Transformer. The permutation-invariant nature of the self-attention mechanism in transformer (Vaswani et al., 2017) leads to loss in temporal information of the input sequence.

Model	Dense Captioning YouCook2			Highlight Detection QVHighlights		Temporal Grounding Charades-STA	
	SODA_c	CIDEr	F1	mAP	HIT@1	R@1 (IoU=0.5)	R@1 (IoU=0.7)
LongVLM (Weng et al., 2024)	2.3	8.1	16.9	16.0	22.5	27.2	11.9
TimeChat (Ren et al., 2024)	<u>*3.1*</u>	<u>*10.3*</u>	<u>*19.5*</u>	<u>*21.7*</u>	<u>*37.9*</u>	<u>*46.7*</u>	<u>*23.7*</u>
Gen-S (Yao et al., 2025)	–	–	–	–	–	38.7	15.2
NumPro-FT (Wu et al., 2025)	–	–	–	–	–	42.0	20.6
LLAVA-ST (Li et al., 2025)	–	–	–	–	–	44.8	23.4
TemporalVLM	**3.4**	**13.2**	**20.0**	**25.1**	**43.0**	**54.4**	**29.0**

Table 2: Supervised results. We first fine-tune all models on TimeIT and Valley, and then perform task-specific fine-tuning and evaluation on YouCook2 for dense video captioning, QVHighlights for video highlight detection, and Charades-STA for temporal video grounding. Best results are in **bold**, while second best ones are <u>*underlined*</u>.

Model	F1@{10,25,50}	Edit	Acc
LongVLM (Weng et al., 2024)	{<u>*17.1*</u>, <u>*13.3*</u>, <u>*7.3*</u>}	<u>*47.3*</u>	<u>*47.6*</u>
TimeChat (Ren et al., 2024)	{11.1, 8.5, 4.4}	32.7	46.0
TemporalVLM	{**22.3**, **18.3**, **11.1**}	**56.2**	**52.9**

Table 3: Temporal action segmentation results in supervised setting on IndustryASM. All models are fine-tuned on IndustryASM before evaluation. Best results are in **bold**, while second best ones are <u>*underlined*</u>.

Model	SODA	CIDEr	F1
No Overlap + No Fusion	2.3	11.0	17.3
No Overlap + Fusion	2.0	9.5	16.2
Overlap + No Fusion	<u>*2.8*</u>	<u>*11.7*</u>	<u>*18.9*</u>
Overlap + Fusion	**3.4**	**13.2**	**20.0**

Table 4: Effects of overlapping sliding video Q-Former. Best results are in **bold**, second best are <u>*underlined*</u>.

Model	# Layers	SODA_c	CIDEr	F1
Average Pooling	1	1.7	6.0	16.1
Linear Layer	1	2.4	<u>*9.5*</u>	17.8
Transformer	2	2.3	6.8	15.6
LSTM	1	2.8	9.4	18.2
LSTM	2	<u>*2.9*</u>	9.3	<u>*18.5*</u>
BiLSTM	2	**3.4**	**13.2**	**20.0**

Table 5: Effects of BiLSTM module. Best results are in **bold**, while second best ones are <u>*underlined*</u>.

Though using positional encoding alleviates this issue, it is inevitable that some temporal information loss occurs (Zeng et al., 2023). In contrast, (Bi)LSTM processes data one by one and preserves the sequence order, making them better suited for temporal modeling. Similar to (Zeng et al., 2023), we also observe in Tab. 5 that linear layer outperforms transformer (Vaswani et al., 2017).

5.4 Comparisons with Specialized Models

Tab. 6 shows that specialized models such as Vid2Seq (Yang et al., 2023) and QD-DETR (Moon et al., 2023), whose advantages come from task-specific training data and model designs, outperform generalist models such as TimeChat (Ren et al., 2024) and our TemporalVLM model. In particular, the use of 4 loss functions for 200 epochs during training and the utilization of saliency tokens for saliency prediction in QD-DETR lead to superior performance in video highlight detection tasks. In contrast, our TemporalVLM model is trained on a simpler language modeling loss for a smaller number of epochs. Nevertheless, generalist models exhibit better generalization across zero-shot, multi-task, and multi-domain settings.

5.5 Qualitative Results

Dense Video Captioning Results. Fig. 4 shows example results in the zero-shot setting on a YouCook2 (Zhou et al., 2018) video. TimeChat (Ren et al., 2024) fails to predict any correct timestamps and tends to hallucinate. LongVLM (Weng et al., 2024) yields better results but still hallucinates objects not present in the video and produces inaccurate captions. In contrast, TemporalVLM produces much more accurate timestamps and captions. Also, it is the only model that provides captions till the end of the video, showing the effectiveness for long video understanding.

Temporal Action Segmentation Results. Fig. 5 presents example results in the supervised setting on an IndustryASM video. TimeChat (Ren et al., 2024) fails to segment the video and hallucinates actions not present in the video. LongVLM (Weng et al., 2024) fares better in terms of the actions detected in the video but struggles with predicting the action segments and hallucinations further into the video. In contrast, TemporalVLM predicts the action segments notably closer to the ground truth.

Model	Dense Captioning YouCook2			Highlight Detection QVHighlights		Temporal Grounding Charades-STA	
	SODA_c	CIDEr	F1	mAP	HIT@1	R@1 (IoU=0.5)	R@1 (IoU=0.7)
Vid2Seq (Yang et al., 2023)	**7.9**	**47.1**	**27.3**	–	–	–	–
QD-DETR (Moon et al., 2023)	–	–	–	<u>*38.9*</u>	<u>*62.4*</u>	–	–
QD-DETR w/ Audio (Moon et al., 2023)	–	–	–	**39.0**	**62.9**	–	–
MMN (Wang et al., 2022)	–	–	–	–	–	50.5	<u>*29.7*</u>
VDI (Luo et al., 2023a)	–	–	–	–	–	<u>*52.3*</u>	**31.4**
Timechat (Ren et al., 2024)	3.1	10.3	19.5	21.7	37.9	46.7	23.7
TemporalVLM	<u>*3.4*</u>	<u>*13.2*</u>	<u>*20.0*</u>	25.1	43.0	**54.4**	29.0

Table 6: Comparisons with specialized models. Best results are in **bold**, while second best ones are <u>*underlined*</u>.

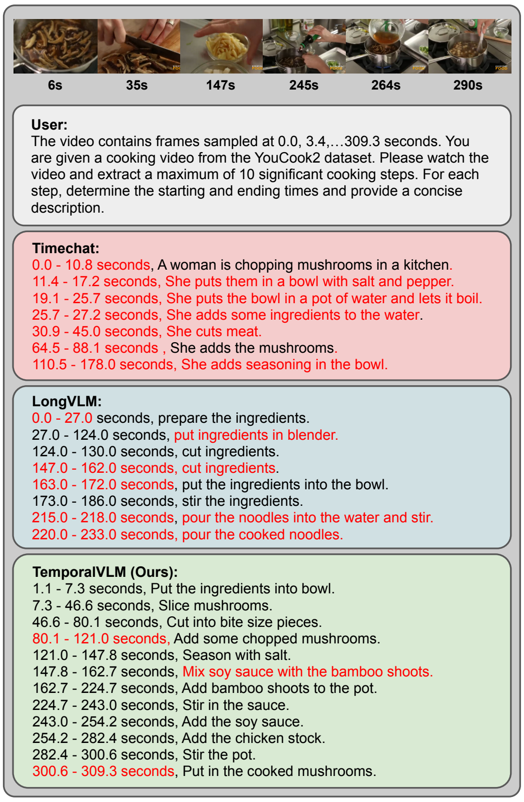


Figure 4: Dense video captioning in zero-shot setting on YouCook2. **Red** denotes inaccuracies.

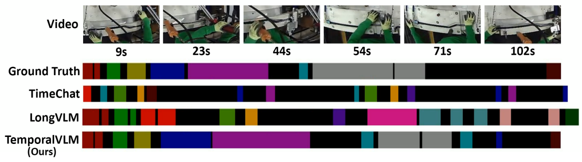


Figure 5: Temporal action segmentation in supervised setting on IndustryASM. **Black** denotes background.

6 Discussion

Novelty of TemporalVLM. Existing frameworks either lack explicit time-aware modeling (Video-ChatGPT (Maaz et al., 2023), LongVLM (Weng et al., 2024)) or do not adopt a coarse-to-fine design (Video-ChatGPT, TimeChat (Ren et al., 2024)), and typically aggregate global information via pooling (Video-ChatGPT, LongVLM) or query-based aggregation (TimeChat). In contrast, our time-aware coarse-to-fine encoder explicitly models temporal information, with (i) a time-aware clip encoder extracting fused time-aware local features and (ii) a BiLSTM aggregating these features to capture long-range temporal dependencies. This enables fine-grained video understanding and temporal reasoning within a unified framework, yielding consistent improvements across several benchmarks.

Generalizability of IndustryASM. While our focus is on industrial assembly, the underlying tasks, i.e., multi-step procedure understanding, temporal dependency reasoning, and long-range event modeling, are common across many long-video domains, including instructional, maintenance, and tutorial videos. Its complex temporal structures, long durations, and multi-step workflows make it a challenging benchmark. Consequently, despite domain specificity, IndustryASM captures the general characteristics of procedural long videos and provides a valuable benchmark for evaluating temporal reasoning models beyond industrial settings.

7 Conclusion

We propose TemporalVLM for temporal reasoning and fine-grained understanding in long videos. Our approach includes a time-aware clip encoder for extracting fused time-aware local features and a BiLSTM for global feature aggregation. Extracted features are time-sensitive and contain both local and global cues. Moreover, we present IndustryASM. Lastly, extensive experiments show our superior results over prior works. To our best knowledge, this is the first work to blend LSTMs into video LLMs. Our future work will explore advanced recurring models (Gu et al., 2021; Gu and Dao, 2024).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. 2022. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2917–2927.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and 1 others. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Soda: Story oriented dense video captioning evaluation framework. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 517–531. Springer.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275.

Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. 2021. Frameexit: Conditional early exiting for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15608–15618.

Shreyank N Gowda, Marcus Rohrbach, and Laura Sevilla-Lara. 2021. Smart frame selection for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1451–1459.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*.

Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Tengda Han, Weidi Xie, and Andrew Zisserman. 2022. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916.

Jing Hu, Yiming Yang, Huan Wang, and Eric P. Xing. 2023. Lora: Low-rank adaptation of large language models. In *Proceedings of the 2023 Conference on Neural Information Processing Systems (NeurIPS 2023)*.

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

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. 2024b. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pages 202–218. Springer.

Syed Waleed Hyder, Muhammad Usama, Anas Zafar, Muhammad Naufil, Fawad Javed Fateh, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. 2024. Action segmentation using 2d skeleton heatmaps and multi-modality fusion. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1048–1055. IEEE.

Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. 2021. Pytorch. *Programming with TensorFlow: solution for edge computing applications*, pages 87–104.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. 2024. 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.

Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. 2021. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16020–16030.

Deep Learning. 2016. Ian goodfellow. *Yoshua Bengio, and Aaron Courville*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858.

Hongyu Li, Jinyu Chen, Ziyu Wei, Shaofei Huang, Tianrui Hui, Jialin Gao, Xiaoming Wei, and Si Liu. 2025. Llava-st: A multimodal large language model for fine-grained spatial-temporal understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8603.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

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

Shijie Li, Yazan Abu Farha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. 2020. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):6647–6658.

Han Lin, Tushar Nagarajan, Nicolas Ballas, Mido Assran, Mojtaba Komeili, Mohit Bansal, and Koustuv Sinha. 2024. Vedit: Latent prediction architecture for procedural video representation learning. *arXiv preprint arXiv:2410.03478*.

Haogeng Liu, Qihang Fan, Tingkai Liu, Linjie Yang, Yunzhe Tao, Huaibo Huang, Ran He, and Hongxia Yang. 2023. Video-teller: Enhancing cross-modal generation with fusion and decoupling. *arXiv preprint arXiv:2310.04991*.

Ruyang Liu, Chen Li, Yixiao Ge, Thomas H Li, Ying Shan, and Ge Li. 2024. Bt-adapter: Video conversation is feasible without video instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13658–13667.

Zijia Lu and Ehsan Elhamifar. 2024. Fact: Frame-action cross-attention temporal modeling for efficient action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18175–18185.

Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. 2023a. Towards generalisable video moment retrieval: Visual-dynamic injection to image-text pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23045–23055.

Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. 2023b. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Syed Ahmed Mahmood, Ali Shah Ali, Umer Ahmed, Fawad Javed Fateh, M Zeeshan Zia, and Quoc-Huy Tran. 2026. Procedure learning via regularized gromov-wasserstein optimal transport. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6925–6935.

WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. 2023. 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.

OpenAI. 2024. Chatgpt: Generative pre-trained transformer.

Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. 2024. Momentor: Advancing video large language model with fine-grained temporal reasoning. *arXiv preprint arXiv:2402.11435*.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. 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.

Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. 2022. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106.

Fadime Sener, Dipika Singhania, and Angela Yao. 2020. Temporal aggregate representations for long-range video understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 154–171. Springer.

Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. 2021. An image is worth 16x16 words, what is a video worth? *arXiv preprint arXiv:2103.13915*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others. 2024. 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.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216.

Aakarsh Taori, Evan Chen, Sainbayar Sukhbaatar, and 1 others. 2023. Alpaca: A strong, replicable instruction-following model. *arXiv preprint arXiv:2303.11347*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. 2021. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6847–6857.

Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. 2022. Negative sample matters: A renaissance of metric learning for temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2613–2623.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2024. Longvlm: Efficient long video understanding via large language models. *arXiv preprint arXiv:2404.03384*.

Chao-Yuan Wu and Philipp Krahenbuhl. 2021. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. 2022. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597.

Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. 2025. Number it: Temporal grounding videos like flipping manga. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13754–13765.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10714–10726.

Linli Yao, Haoning Wu, Kun Ouyang, Yuanxing Zhang, Caiming Xiong, Bei Chen, Xu Sun, and Junnan Li. 2025. Generative frame sampler for long video understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17900–17917.

Fangqiu Yi, Hongyu Wen, and Tingting Jiang. 2021. Asformer: Transformer for action segmentation. *arXiv preprint*.

Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. 2022. 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.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

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

Yucheng Zhao, Chong Luo, Chuanxin Tang, Dongdong Chen, Noel Codella, and Zheng-Jun Zha. 2023. Streaming video model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14602–14612.

Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. 2023. Learning procedure-aware video representation from instructional videos and their narrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14825–14835.

Luowei Zhou, Chenliang Xu, and Jason Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Wanrong Zhu, Bo Pang, Ashish V Thapliyal, William Yang Wang, and Radu Soricut. 2022. End-to-end dense video captioning as sequence generation. *arXiv preprint arXiv:2204.08121*.

Supplementary Material

In this supplementary material, we first provide the additional details of our IndustryASM dataset (including dataset statistics, annotation process, maintenance and release plans) and our TemporalVLM implementation in Secs. A and B respectively. Next, we demonstrate the effects of number of training epochs in Sec. C. We then evaluate our model for general video understanding in Sec. D. Furthermore, Sec. E compares the sizes of our model against TimeChat (Ren et al., 2024) and LongVLM (Weng et al., 2024), while we report the run time comparisons and FLOP comparisons in Secs. F and G respectively to evaluate the efficiency of our model. Moreover, Sec. H assesses the impacts of number of clips, while we study the effects of employing a more advanced LLM in our model in Sec. I. We then present several qualitative results in Sec. J, including dense video captioning, video highlight detection, temporal video grounding, and more importantly, generalization results. Finally, we discuss the limitations and societal impacts of our work in Secs. K and L respectively.

A IndustryASM Details

A.1 Dataset Statistics

Our IndustryASM dataset comprises of 4851 videos in total and the average video duration is 105 seconds. Therefore, the total dataset duration is 142 hours. These videos are distributed among 47 industry assembly processes or datasets, ranging from automotive manufacturing, electronic device manufacturing, medical device manufacturing to heating, ventilation, and air conditioning (HVAC) manufacturing. Participants in our dataset are salaried factory workers based in the US. Informed consent is obtained from all participants in accordance with standard operating procedures prior to data collection. As an additional privacy safeguard, all faces and other personally identifiable information (PII) are blurred or otherwise anonymized in the recorded data. Each assembly process or dataset involves 12 steps or actions on average. We first categorize each step into one of the following classes:

- **Moving:** In this category, the worker(s) transfer parts or subassemblies within or between workstations, relocating items from one position to another within the workspace.

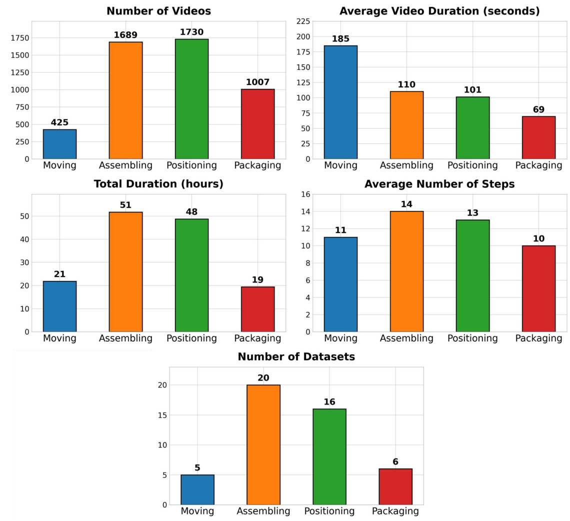


Figure 6: Dataset statistics for our IndustryASM dataset.

- **Assembling:** The worker(s) build or complete subassemblies by adding components or combining parts to form a complete unit.
- **Positioning:** This involves the worker(s) placing parts onto a subassembly or mounting a subassembly in a designated position within the workstation.
- **Packaging:** In this category, the worker(s) place subassemblies or components into boxes or other forms of packaging materials.

Next, we further classify each assembly process or dataset into one of the above categories which appears the most among all of its steps. For example, the Air_Cleaner dataset (P06) includes 10 steps (namely, 4 moving steps, 3 assembling steps, and 3 positioning steps), and hence it is classified as "Moving" which appears the most among its 10 steps. We provide the statistics for each category of the IndustryASM dataset in Fig. 6.

Finally, examples of these 4 categories are presented in Fig. 3 of the main paper. In addition, Fig. 7 presents more examples to better illustrate the diverse camera viewpoints, actors, backgrounds, and activities in our IndustryASM dataset that are common in manufacturing settings.

A.2 Annotation Process

Our IndustryASM videos are annotated with action names and action timestamps by industrial engineers. To ensure quality, two labelers are assigned for each video, i.e., one labeler provides the labels for the video, while another labeler checks the labels. If there are conflicts, both labelers discuss to

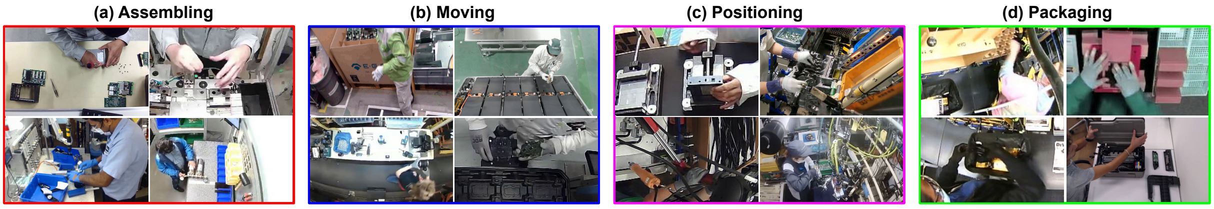
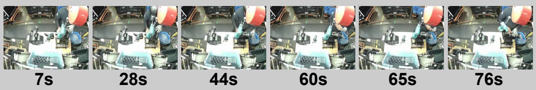


Figure 7: Example videos with different camera viewpoints, actors, backgrounds, and activities from our IndustryASM dataset.



User Instruction:
Locate and specify a series of precise actions or steps in the industrial assembly video, including their start and end timestamps. The output format of each predicted step should be like: start - end seconds, action. An specific example is: 14.4 - 15.9 seconds, place valve in fixture.

Target Response:
7.2 - 8.6 seconds, Place air cleaner into table jig.
8.6 - 10.4 seconds, Write the last 3 of sequence number on air cleaner.
25.8 - 34.7 seconds, Write the last 3 of sequence number on air cleaner.
37.2 - 40.5 seconds, Place air cleaner into table jig.
43.0 - 47.7 seconds, Align and insert the air duct locator notch and locator pin in the air cleaner until fully seated.
50.1 - 56.7 seconds, Secure the air duct to the air cleaner by tightening the band clamp.
59.4 - 61.0 seconds, Pick air flow meter and Verify that the part is aligned with arrow facing up.
61.0 - 63.2 seconds, Set air flow meter into air cleaner receiving slot.
62.6 - 68.6 seconds, Simultaneously pick up two screws.
70.0 - 74.6 seconds, Secure air flow meter to air cleaner with the two screws until tight.
76.9 - 77.4 seconds, Using a china marker mark the tops of each screw to verify they are tight.
78.7 - 82.0 seconds, Remove air cleaner from jig and place into pick rack

Figure 8: Examples of our instruction and target response for the temporal action segmentation task on an IndustryASM video.

fix them. In general, roughly 8% of the videos have conflicts and need fixing, resulting in an agreement rate of around 92% between labelers. We manually write the user instructions for the temporal action segmentation task and generate the ground truth responses by using the action names and action timestamps. Fig. 8 illustrates examples of our instruction and ground truth response for temporal action segmentation on an IndustryASM video.

A.3 Maintenance and Release Plans

We host our IndustryASM dataset on Google Drive. We will release version updates to incorporate any corrections and add additional data/annotations if and when available. These updates will follow a versioning scheme (e.g., v1.0) and there will be documentations on any changes in the release notes. Our dataset can be downloaded at https://retrocausal.ai/research/ and we plan to actively maintain it for at least 5 years following the release date.

B Implementation Details

We perform grid search to find the best hyperparameter settings for TemporalVLM on supervised dense video captioning on YouCook2 (Zhou et al., 2018) and use them for all the remaining experiments. We present our hyperparameter settings in Tab. 7. We sample $C = 6$ clips from the video, with each consisting of $N_f^c = 96$ frames. The frame encoder processes frames of sizes 224×224 across 14×14 patches. Instead of sending timestamps from across the entire video, we send clip-wise timestamps along with their corresponding frames to the frame encoder and clip encoder respectively. Inside the clip encoder, we divide the timestamp encoded input to the video Q-former into windows of size $q = 32$ frames and overlap $o = 16$ frames. We then fuse the concatenated window-wise outputs of the Q-former via multi-headed self-attention with 8 attention heads that have the same hidden size N_v as the Q-Former. The output of the clip encoder contains a fused time-aware representation of each clip.

We then concatenate the time-aware clip features according to their temporal order of appearance in the video. This sequence is of dimensions $(C \times N_f^c,\ N_V)$, with N_V denoting the dimension of the video tokens. This sequence is passed to the BiLSTM module which uses information from both past and future states to output a global representation of dimensions $(C \times N_f^c,\ 2 \times N_V)$. The BiLSTM module comprises of two hidden layers, with one hidden layer for the forward LSTM and another hidden layer for the backward LSTM. Projection layers are used to project the BiLSTM output into the LLM input space. The first layer projects the BiLSTM output to the dimensions

Hyperparameter	Value
Number of clips C	6
Number of frames sampled per clip N_f^c	96
Frame encoder patch size	14×14
Frame resolution	224×224
Frame sampling type	uniform
Number of epochs	7
Batch size	32
Learning rate	5e-5
Warmup learning rate	5e-6
Weight decay	0.01
Optimizer	Adam
AdamW β	(0.9,0.999)
Q-Former window size q	32
Overlap o	16
Number of visual tokens per window	32
Number of layers in clip encoder	2
Number of layers in image encoder	12
Number of attention heads in fusion module	8
Q-Former hidden size N_V	768
Fusion module hidden size D	768
BiLSTM input size	768
BiLSTM hidden size	768
LLM hidden size N_{LLM}	4096

Table 7: Hyperparameter settings.

of $(C \times N_f^c, N_{LLM})$ and then the second layer projects the output of the first layer to the sizes of (N_f^c, N_{LLM}) which are the input dimensions required by the LLM.

C Effects of Number of Training Epochs

Tab. 8 shows the performance of our model across different training epochs. Since our model uses a learnable fusion module in the clip encoder and a learnable BiLSTM module to aggregate clip-level features into a global representation, we observe the best results by training for 7 epochs. Tab. 8 reports our results with 3 epochs, which are worse than ours with 7 epochs but still outperform those of TimeChat (Ren et al., 2024) and LongVLM (Weng et al., 2024).

D General Video Understanding Results

Following LongVLM (Weng et al., 2024), we evaluate the performance of our TemporalVLM model on the general video understanding benchmark provided by Video-ChatGPT (Maaz et al., 2023) in Tab. 9. The evaluation metrics include Correctness Information (CI), Detail Orientation (DO), Contextual Understanding (CU), Temporal Understanding (TU), and Consistency (C). It is evident from Tab. 9 that our TemporalVLM model achieves the best performance across all metrics, outperforming all competing methods, including the original LongVLM model.

E Model Size Comparisons

Tab. 10 compares the sizes of our TemporalVLM model, TimeChat (Ren et al., 2024), and LongVLM (Weng et al., 2024) in terms of the number of learnable parameters (measured in millions). Our TemporalVLM model includes a learnable fusion module and a learnable BiLSTM module, leading to a 5% and 4% increase in the number of trainable parameters over TimeChat and LongVLM respectively. Nevertheless, our TemporalVLM model achieves the best performance across various temporal reasoning and fine-grained understanding tasks, despite using less training data than TimeChat. In particular, TimeChat is trained on the complete TimeIT (Ren et al., 2024) and Valley (Luo et al., 2023b) datasets, whereas our TemporalVLM model is trained on a subset[2] of the TimeIT and Valley datasets.

F Run Time Comparisons

We compare the inference times (measured in seconds) of different models on three tasks, i.e., Dense Video Captioning (DVC) on YouCook2 (Zhou et al., 2018), Video Highlight Detection (VHD) on QVHighlights (Lei et al., 2021), and Temporal Video Grounding (TVG) on Charades (Gao et al., 2017) in Tab. 11. From the results, TimeChat (Ren et al., 2024) is about 35% more efficient than LongVLM (Weng et al., 2024) and TemporalVLM, since both LongVLM and TemporalVLM perform clip sampling and clip encoding.

G FLOPS Comparisons

We further compute the number of floating point operations per second (FLOPS) during the forward pass of the model as a measure of runtime efficiency. As observed from Tab. 12, TemporalVLM conducts 4.46% more flops than LongVLM (Weng et al., 2024) and 16.83% more flops than TimeChat (Ren et al., 2024), since it relies on clip-level attention based fusion and BiLSTM aggregation.

H Effects of Number of Short-Term Clips

Tab. 13 presents the effects of number of short-term clips on our TemporalVLM model. Due to memory limitations, we have experimented with three values for the number of short-term clips,

[2]We could not download YT-Temporal (Zellers et al., 2022) due to its large size and the restricted number of downloads.

Model	Epochs	Dense Captioning			Highlight Detection		Temporal Grounding	
		YouCook2			QVHighlights		Charades-STA	
		SODA_c	**CIDEr**	**F1**	**mAP**	**HIT@1**	**R@1 (IoU=0.5)**	**R@1 (IoU=0.7)**
TimeChat (Ren et al., 2024)	3	3.1	10.3	19.5	21.7	37.9	46.7	23.7
LongVLM (Weng et al., 2024)	3	2.3	8.1	16.9	16.0	22.5	27.2	11.9
TemporalVLM	3	*3.2*	*12.9*	*19.7*	*23.9*	*42.3*	*50.0*	*25.9*
TemporalVLM	7	**3.4**	**13.2**	**20.0**	**25.1**	**43.0**	**54.4**	**29.0**

Table 8: Effects of number of training epochs. Best results are in **bold**, while second best ones are *underlined*.

Model	Data	CI	DO	CU	TU	C
Video-LLaMA (Zhang et al., 2023)	10M	1.96	2.18	2.16	1.82	1.79
Video-ChatGPT (Ren et al., 2024)	100K	2.50	2.57	2.69	2.16	2.20
Valley (Luo et al., 2023b)	234K	2.43	2.13	2.86	2.04	2.45
BT-Adapter (Liu et al., 2024)	10M	2.68	2.69	3.27	2.34	2.46
LongVLM (Weng et al., 2024)	100K	*2.76*	*2.86*	*3.34*	*2.39*	*3.11*
TemporalVLM	100K	**2.88**	**2.91**	**3.45**	**2.50**	**3.16**

Table 9: General video understanding results. Best results are in **bold**, while second best ones are *underlined*.

Model	Parameters
TimeChat (Ren et al., 2024)	241M
LongVLM (Weng et al., 2024)	244M
TemporalVLM	255M

Table 10: Model size comparisons.

Model	DVC	VHD	TVG
TimeChat (Ren et al., 2024)	10.1s	5.6s	3.7s
LongVLM (Weng et al., 2024)	13.2s	7.3s	4.9s
TemporalVLM	13.5s	7.6s	5.1s

Table 11: Run time comparisons.

Model	FLOPS/iteration
TimeChat (Ren et al., 2024)	1.01×10^{13}
LongVLM (Weng et al., 2024)	1.13×10^{13}
TemporalVLM	1.18×10^{13}

Table 12: FLOPS/iteration comparisons.

Clips	SODA_c	CIDEr	F1
2	2.6	9.2	18.5
4	*3.1*	*11.3*	*18.9*
6	**3.4**	**13.2**	**20.0**

Table 13: Effects of number of short-term clips. Best results are in **bold**, while second best ones are *underlined*.

namely 2, 4, and 6. It is evident from Tab. 13 that the performance is improved with increasing the number of short-term clips, since our model is able to access more data from the input video.

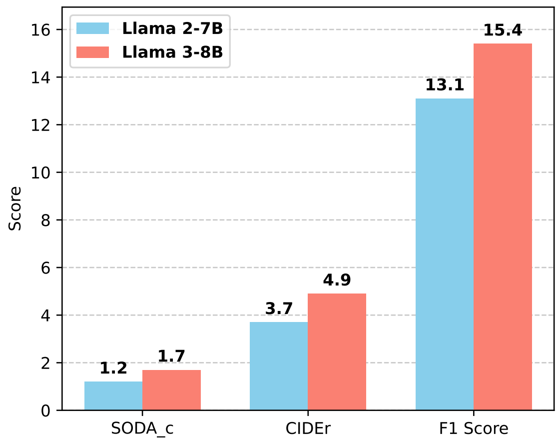


Figure 9: Results with a more recent LLM.

I Results with a More Recent LLM

The performance gain achieved by employing a more advanced LLM as the language decoder in our TemporalVLM model is illustrated in Fig. 9. As LLama 3-8B is pretrained on a significantly larger and diverse dataset, it outperforms Llama 2-7B by a significant margin. The results demonstrate that as LLMs improve, the performance of VLMs, including our TemporalVLM model, will also improve alongside it.

J Qualitative Results

In addition to Figs. 4 and 5 in the main paper, we provide additional qualitative results in this section. In particular, Figs. 10, 11, and 12 present qualitative results by TemporalVLM in the supervised setting for dense video captioning, video highlight detection, and temporal video grounding respectively. More importantly, qualitative results demonstrating the generalization abilities of TemporalVLM in the zero-shot setting are shown in Fig. 13. Overall, our TemporalVLM model demonstrates promising performance in a variety of temporal reasoning and fine-grained understanding tasks.

K Limitations

Despite promising performance in several temporal reasoning tasks, including dense video captioning, temporal video grounding, video highlight detection, and temporal action segmentation, our time-aware video LLM may struggle with complex temporal reasoning tasks and videos with extreme durations. In addition, our TemporalVLM model often has difficulty dealing with small objects, since it does not include an explicit object detector. Our future works will enhance TemporalVLM for tackling complex temporal reasoning tasks, videos with extreme durations, and dealing with small objects.

L Societal Impacts

Our time-sensitive video LLM could enable a variety of applications including optimizing and tracking industry assembly processes. More specifically, for assembly process optimization, industrial engineers could utilize TemporalVLM to automatically decompose a video recording of an assembly process into segments as well as generate a brief description for each segment. In addition, TemporalVLM could improve the performance of frontline workers by tracking their assembly process and notify them as soon as they miss a step. Nevertheless, we acknowledge that our time-aware video LLM could be misused for surveillance and monitoring of individuals, which emphasizes the importance of responsible AI principles to guide the use of this technology.

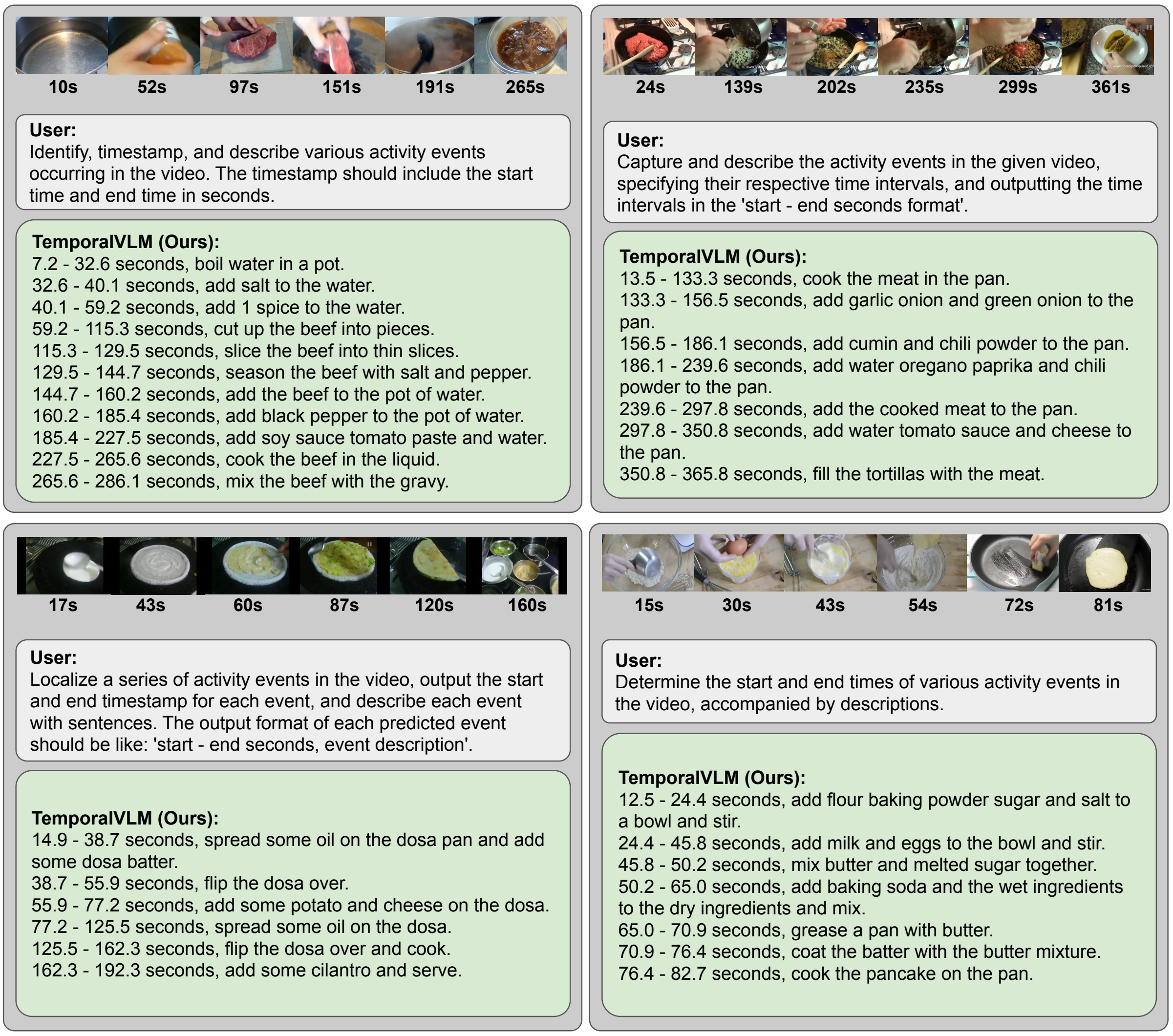


Figure 10: Qualitative examples highlighting the dense video captioning capabilities of TemporalVLM in the supervised setting. The model is asked to provide the timestamps of actions that occur in the videos along with brief descriptions of the actions.

Figure 11: Qualitative examples demonstrating the video highlight detection capabilities of TemporalVLM in the supervised setting. The model is given a video along with an action description. It is prompted to provide the frames that match the action description and their saliency scores.

Figure 12: Qualitative examples showing the temporal video grounding capabilities of TemporalVLM in the supervised setting. A video and a query is given to the model. It is prompted to provide the timestamps at which the query occurs. **Magenta** represents the ground truth and predicted timestamps for the first query, while **Orange** indicates those for the second query.

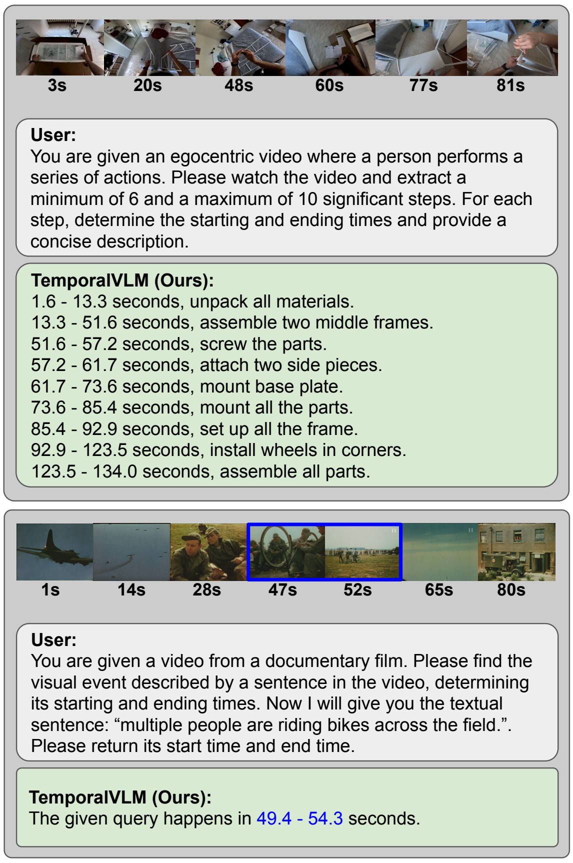


Figure 13: Examples of the generalization capabilities of TemporalVLM in the zero-shot setting. (Top) The model is prompted to provide the timestamps of actions and brief descriptions of actions for an egocentric video of furniture assembly. (Bottom) The model is provided with a documentary film. It is asked to predict the timestamps when the query happens. **Blue** denotes the ground truth and predicted timestamps.