ReVisionLLM: Recursive Vision-Language Model for Temporal Grounding in Hour-Long Videos

Supplementary Material

Our supplementary materials contain Section S1: Additional Implementation Details, Section S2: Calibration Confidence, Section S3: Generalization: Text-to-Video Retrieval, and Section S4: Aditional Qualitative Results.

S1. Additional Implementation Details

Multimodal Encoder. To handle multimodal inputs (refer to Fig. S1), we utilize the Frozen CLIP L/14 model [**?**], a 24-layer transformer pretrained on extensive image-text pairs using a contrastive learning objective [**?**]. The vision encoder processes inputs of size 224×224, producing both spatial tokens and a global CLS token. For computational efficiency, we use only the CLS token to represent each frame in long videos. Similarly, the CLIP text encoder, a 12-layer transformer, extracts feature representations for the queried event from the input text.

Hierarchical Adapter. As illustrated in Fig. S1, the Hierarchical Adapter processes the i^{th} video segment C^i to generate both sparse (S^i) and dense (D^i) features. For the MAD dataset, video features are divided into sliding windows of $L_w = 125$ seconds, while for the VidChapters-7M dataset, the window length is $L_w = 500$ seconds. From each segment, 250 frames are uniformly sampled and used as input to the hierarchical adapter.

Sparse features S^i are computed using a combination of cross-attention and self-attention mechanisms, each implemented with two layers ($N = 2$). This lightweight design ensures minimal computational overhead compared to the 24 transformer layers of the original CLIP Vision Encoder. Dense features D^i are derived by projecting the CLIP-encoded frame features (dimension 768) into the embedding space of the Large Language Model (dimension 4096) [**?**] using a linear transformation.

Large Language Model. We utilize a pre-trained Vicuna-7B [**?**] model to ground queried events using the adapted visual features. Built upon LLaMA [**?**], this model consists of 32 transformer layers and has been fine-tuned on 70K user-shared conversations from ShareGPT [**?**].

To enhance training efficiency, we adopt Low-Rank Adaptation (LoRA) [**?**], a method commonly used in recent works [**? ?**]. LoRA allows us to fine-tune the model without modifying its core weights by introducing lightweight, trainable modules. This significantly reduces computational overhead while retaining the model's flexibility. For our setup, we configure LoRA with a rank of $r = 64$ and a scaling factor of $\alpha = 128$.

Training on ReVisionLLM Model We begin by pretraining the Linear Projector (Fig. S1) using the LCS-558K dataset from LLaVA [**?**]. This step aligns the CLS token from the CLIP Vision Encoder with the LLM's embedding space. The projector is trained for 1 epoch with a batch size of 128 and a learning rate of 1×10^{-3}.

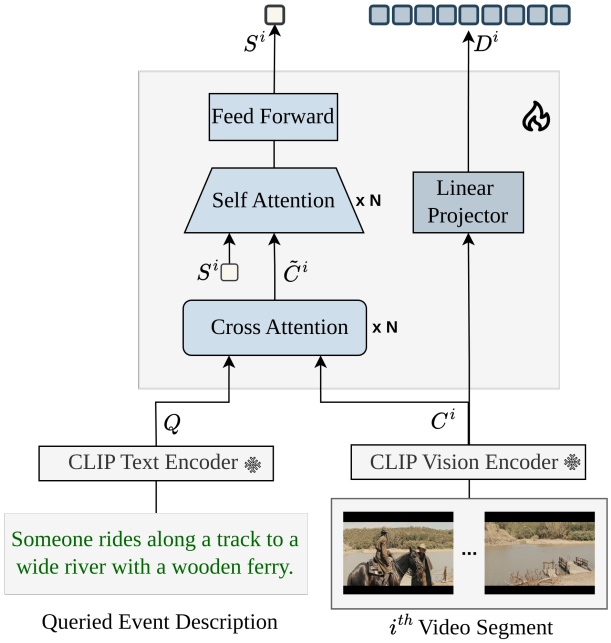

Figure S1. **Hierarchical Adapter** processes the features extracted by the multimodal encoder, using both the video segments and the textual description of the queried event as inputs. It generates two types of temporal features: sparse and dense. Sparse features are computed through a combination of cross-attention, self-attention, and a feed-forward network, while dense features are generated using a linear projection layer.

Following pretraining, we implement a two-stage training pipeline for ReVisionLLM, maintaining a consistent learning rate of 1×10^{-4}. We use the AdamW optimizer [**?**] with a warmup ratio 0.03 and a cosine scheduling strategy.

In the first stage, the Linear Projector is frozen, and the LLM is fine-tuned using LoRA on dense features, focusing on the lowest hierarchy level. This stage employs a batch size of 128, spanning 5 epochs for the MAD dataset and 1 epoch for the VidChapters-7M dataset. Sparse temporal features are introduced for upper hierarchies to reduce the LLM's visual input size. To enable sparse feature generation, we freeze the LoRA module and fine-tune the Cross-

Attention, Self-Attention, and Feed-Forward layers of the Hierarchical Adapter, using a batch size of 32 for 1 epoch.

The training in this stage incorporates contrastive video segments where the queried event is absent. These segments are randomly sampled from hour-long videos and do not overlap with the temporal boundaries of the ground truth event. By selecting contrastive segments from the same video, the model is trained to handle challenging inference scenarios, where it must distinguish the queried event from visually and contextually similar scenes within the video.

In the second stage, all components of the Hierarchical Adapter are frozen, and a new LoRA module is fine-tuned for long-video processing on the Stage 2 objective. This stage employs a batch size of 8 and runs for 2 epochs. Two separate LoRA modules are utilized: one optimized for short video training and another adapted for long video processing.

Training on ReVisionLLM-U Model The unified model variant differs from our default model only in its training methodology. Training a unified model with shared parameters across all hierarchical levels poses notable challenges. To address this, we adopt an enhanced two-stage strategy. The first stage, including pretraining, is similar to the procedure used in the ReVisionLLM framework. In the second stage, however, we introduce a dual-training approach, where the ReVisionLLM-U framework is simultaneously trained on both short video clips and hour-long videos. This approach reduces the risk of catastrophic forgetting, ensuring the retention of short-segment representations.

A key challenge arises from the significant differences between short-segment and long-video data. Short segments utilize dense temporal features, while long videos rely on sparse temporal representations, such as CLS features. Additionally, short-segment training involves only video features, whereas long-video descriptions require both video and text features as inputs. To reconcile these differences, we implement an alternating batching strategy. During training, batches of short segments and long videos are alternately sampled, enabling the model to learn effectively from both data types.

This alternating training strategy not only mitigates catastrophic forgetting but also facilitates the successful training of the ReVisionLLM-U framework, which maintains shared parameters across all hierarchical levels. For ReVisionLLM-U, we employ the same hyperparameters as those used in ReVisionLLM , including learning rate, batch size, training epochs, optimizer, and scheduler (as detailed in the previous section).

Inference for ReVisionLLM. During inference, video segments are created using a sliding window approach. For the MAD dataset, each segment spans 125 seconds with a stride of 25 seconds, while for the VidChapters-7M dataset, segments are 500 seconds long with a stride of 100 seconds. From each segment, 250 frames are uniformly sampled. Sparse temporal features are then extracted using the Hierarchical Adapter and provided as input to the LLM to identify relevant video segments.

In our implementation, we employ two hierarchies with long videos. However, it can be extended to more levels based on video length. At the top level, 100 video segments (approx. 150 minutes) are processed simultaneously, while the second level processes 33 segments (approx. 50 minutes) simultaneously. Both hierarchies identify regions of interest, refined at the lowest hierarchical level. In this final hierarchy, all 250 dense temporal features from the selected segments are processed to pinpoint the precise event boundaries.

Inference for ReVisionLLM-I. The inverse model variant differs from our default model only in the inference method. In this variant, the inference begins at the lowest hierarchical level. All video segments are processed together in a single input batch, with their dense temporal features fed into the LLM. The LLM predicts temporal boundaries for multiple segments, often resulting in a high number of false positives. To mitigate this, the false positives are recursively passed through the second and third hierarchical levels, where they are filtered out. These upper levels retain only the most confident predictions, reducing errors. Finally, the confidence scores from the higher hierarchies are used to adjust and normalize the scores of the initial predictions, improving the overall accuracy of the model. We will release the code and pre-trained models for further use.

S2. Calibration Confidence

Accurate calibration of our model's confidence is crucial for minimizing false positives and improving the overall effectiveness of our approach. To evaluate the impact of our training strategy on model calibration, we compare our method's performance to the baseline VTimeLLM [**?**]. A model is considered well-calibrated if its confidence scores match the actual proportion of correct predictions. Calibration is typically measured using the Expected Calibration Error (ECE) [**?**], which quantifies the difference between predicted probabilities and observed outcomes by dividing the predicted confidence into discrete bins. A lower ECE value indicates better calibration, with an ECE of 0 representing perfect calibration.

For a dataset of N video segments and B evenly spaced bins b_j, the ECE is computed as follows:

$$\widehat{\mathrm{ECE}} = \sum_{j=1}^{B} \frac{|b_j|}{N} \left|\mathrm{conf}(b_j) - \mathrm{acc}(b_j)\right|$$

where $\mathrm{conf}(b_j)$ is the average confidence of samples in bin b_j, $\mathrm{acc}(b_j)$ is the accuracy of predictions in bin b_j, and $|b_j|$ is the number of samples in bin b_j. In our experiments,

$\text{conf}(b_j)$ corresponds to the confidence score (R^i) defined in Section 3.4 of the main paper. A prediction is considered correct if the Intersection over Union (IoU) exceeds a threshold $\tau_{\text{iou}} \in \{0.1, 0.3, 0.5\}$.

By comparing ECE values across models, we can assess the effectiveness of our training strategy in improving calibration and generating more reliable confidence estimates.

Model	ECE @ IoU Thresholds (τ_{IoU})		
	$\tau = 0.1 \downarrow$	$\tau = 0.3 \downarrow$	$\tau = 0.5 \downarrow$
VTimeLLM*	0.6231	0.6233	0.6237
ReVisionLLM	**0.4614**	**0.4698**	**0.4791**

Table S1. **Expected Calibration Error (ECE)** comparison between *Baseline (VTimeLLM+CONE) and Our Model across IoU thresholds (τ_{IoU}). Our model demonstrates better calibration of confidence compared to the baseline across all IoU thresholds. Lower values indicate superior calibration performance.

Table S1 presents a comparison of Expected Calibration Error (ECE) values between the baseline VTimeLLM+CONE model and our ReVisionLLM across three Intersection over Union (IoU) thresholds: $\tau_{\text{IoU}=0.1}$, $\tau_{\text{IoU}=0.3}$, and $\tau_{\text{IoU}=0.5}$. Lower ECE values indicate better calibration performance. In this analysis, we set the number of bins, $B = 10$. Our model consistently outperforms the baseline across all thresholds, highlighting the effectiveness of calibrated fine-tuning in improving the reliability of confidence estimates. The increase in error with higher IoU thresholds (+0.01%) is negligible, further validating the robustness of our approach.

S3. Generalization: Text-to-Video Retrieval

Problem Statement. We show the generalizability of ReVisionLLM on the task of text-to-video retrieval, where the goal is to retrieve the most relevant videos from a given video set $\mathcal{V}$ for a provided query text t describing an event. This involves ranking the videos $v \in \mathcal{V}$ based on their similarity to the query. For this problem, the input consists of a video v and a text t. We represent a video $v \in \mathbb{R}^{T\times 3\times H\times W}$ as a sequence of T image frames, where $v = [v^1, v^2, \ldots, v^F]^T$, and each frame v^f has a spatial resolution of $H \times W$ with 3 color channels. Text S is represented the queried sentence with N_s words.

Task Adaptation for ReVisionLLM. In our original grounding task, we work with a single long video, which we divide into a set of shorter segments, denoted as C. In contrast, for the text-to-video retrieval task, we handle multiple videos, forming a set $\mathcal{V}$. To address this, we combine all the videos into one long sequence and predict the index of the relevant video. We uniformly sample 100 frames from each video and use our vision encoder to extract features, resulting in video features $\hat{\mathcal{V}} \in \mathbb{R}^{|\mathcal{V}|\times 100\times 768}$. These features serve as our video segments, so $C = \hat{\mathcal{V}}$. Additionally, we extract textual features from the query using the text encoder, resulting in $Q \in \mathbb{R}^{N_s\times 768}$, where N_s is the number of words in the query.

For the input prompt, we use: *"<video> Does the <event> happen in the video? Answer yes or no."* The model is trained to respond *"Yes."* for relevant videos and *"No."* for irrelevant ones. In this setup, we use the ReVisionLLM-I variant, where we first process each video at the lowest hierarchical level, then revise our predictions recursively at higher levels. At the upper hierarchies, the prompt becomes: *"<video> in which video can we see the <event> happening?"* The model responds with *"In video v."*, where v denotes the index of the relevant video.

Finally, we rank the predicted videos based on the calibrated confidence scores from our LLM, ensuring more accurate retrieval results.

Dataset Details. The MSR-VTT dataset contains 10,000 videos, each associated with around 20 human-annotated captions. Notably, the captions for a single video often describe distinct parts of the content, aligning with our goal of matching a specific textual query to the most relevant frames within a video. The videos in this dataset range in duration from 10 to 32 seconds. For training, we use *9k-Train* split, including approximately 9,000 videos as outlined in [**?**]. Unless specified otherwise, our experiments use the *9k-Train* split for training. To evaluate our models, we adopt the *1k-Test* set from [**?**], which comprises 1,000 carefully selected video caption pairs.

S4. Aditional Qualitative Results

In this section we provide additional qualitative results for MAD, VidChapters-7M and MSRVTT datasets.

MAD Dataset: In Figure S2, the qualitative results illustrate ReVisionLLM's ability to localize subtle and tiny moments within extremely long videos, often set in visually similar scenes. For Event 1, ReVisionLLM accurately identifies the brief instance where a woman walks off amidst a dimly lit street setting, despite the challenge of nearly identical surrounding frames. This demonstrates the model's precision in grounding temporal boundaries in extended sequences where minor actions must be differentiated. In Event 2, the model successfully localizes the moment of a hazy orange sunrise over the sprawling streets, slums, and skyscrapers of Mumbai. This event, embedded within a visually repetitive urban setting, showcases ReVisionLLM's capacity to detect subtle temporal shifts in lighting and atmosphere. These examples emphasize the model's ability to handle the intricacies of long-form videos, identifying precise moments even

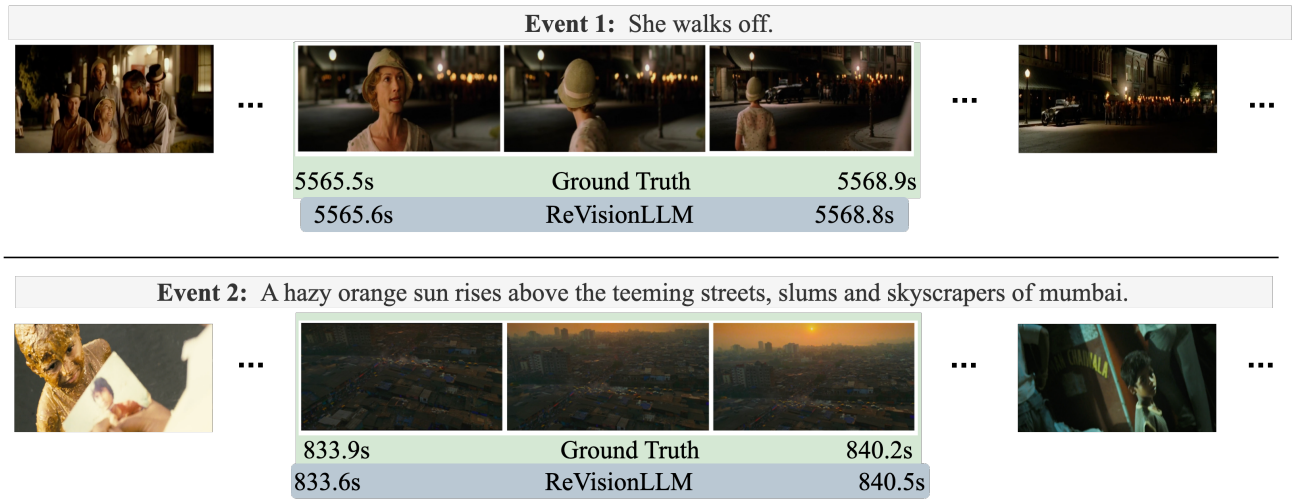

Figure S2. **Additional Qualitative Results** for Long video temporal grounding on the MAD dataset. ReVisionLLM effectively identifies moments within hour-long movies by leveraging a recursive processing approach that operates at both the short video segment level and hour-long videos. Our VLM baseline completely fails to locate the events in these scenarios.

Event 1: People are walking down a street holding signs.

Event 2: A man playing guitar and singing on road side.

Event 3: Dogs are walking across the road in a video.

Event 4: A car is racing on the road.

Figure S3. **Qualitative Results** for Text-to-video retrieval task on MSRVTT dataset. Here, we only show one representative video frame for four diverse queried events, which our model successfully retrieves.

when scenes exhibit minimal variation, thereby enabling enhanced retrieval and understanding of extended video content.

MSRVTT Dataset: In Figure S3, we show a representative video frame for each of the four diverse events that our model successfully retrieved. Events (1) and (2) are visible for short time intervals inside the video, and our model effectively captures these moments due to its ability to focus on fine-grained details. In contrast, events (3) and (4) involve objects with varying speeds and directions, interacting dynamically with their environment. For example, in (3), the dog crosses the road, while in (4), the car moves along the road. These examples demonstrate our model's ability to comprehend and capture both visual and action-related details, enabling it to retrieve the most relevant video from a large dataset.

VidChapers-7M Dataset: In Figure S4, the qualitative results from the VidChapters-7M dataset demonstrate ReVisionLLM's ability to enhance online video search and content retrieval across diverse platforms, including YouTube, educational portals, and news archives. In Event 1, ReVisionLLM accurately localizes a video latency test in a product review, capturing fine-grained temporal details essential for identifying technical demonstrations. Event 2 showcases the model's ability to navigate complex, sequential workflows, pinpointing design importing actions in Canva. Event 3 highlights ReVisionLLM's proficiency in localizing a news report on Trans Mountain pipeline construction, effectively distinguishing dynamic scenes involving machinery and landscapes—key for indexing news and documentaries. In Event 4, the model identifies a cameraman's emotional reaction during a podcast, illustrating its capability to understand contextual nuances and interactions. These results emphasize ReVisionLLM's effectiveness in improving content understanding and enabling precise event retrieval in long-form videos, with strong applicability to online video search engines and content recommendation systems.

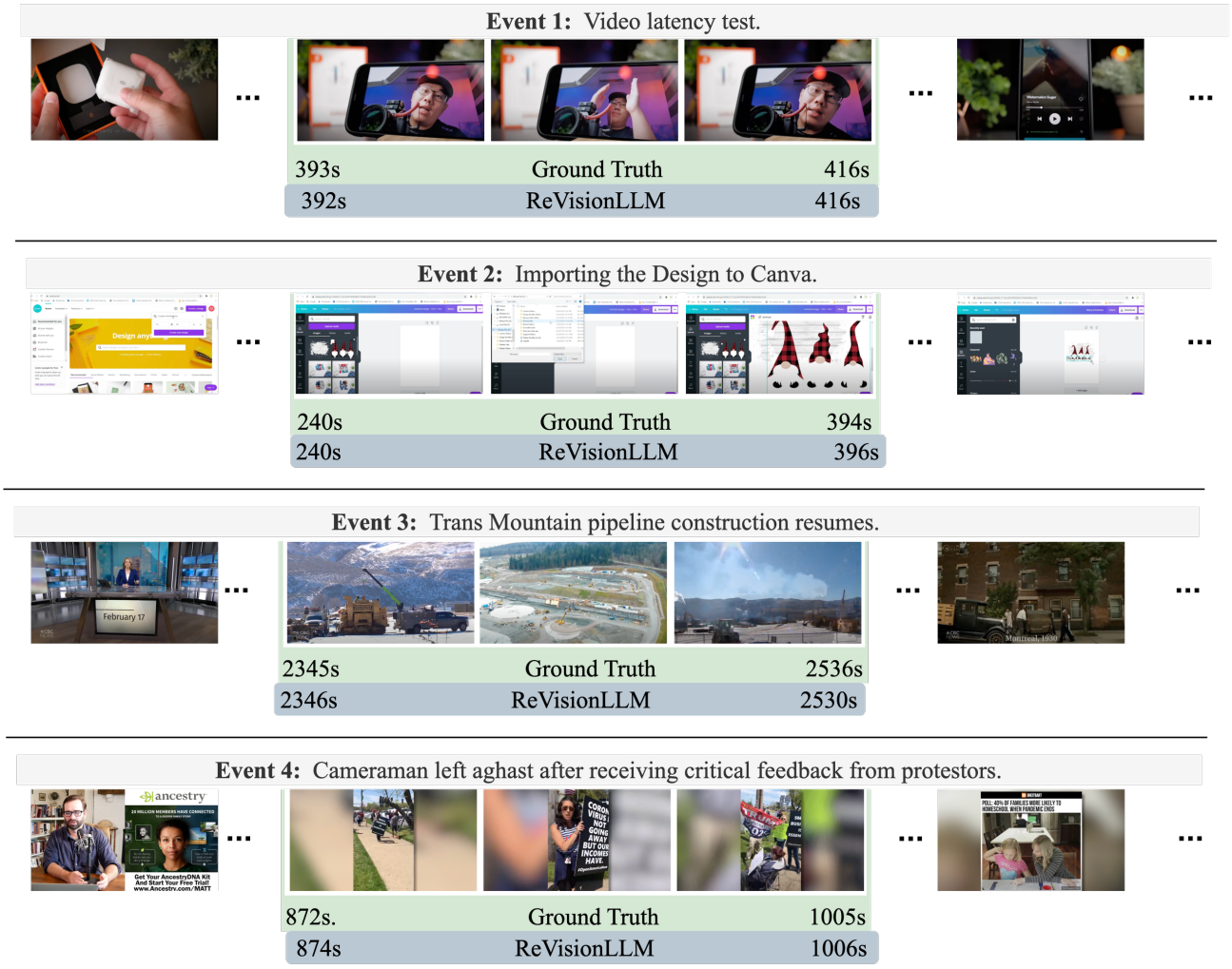

Figure S4. **Qualitative Results** of long video temporal grounding on the VidChapters-7M dataset. ReVisionLLM demonstrates its ability to accurately locate specific events within hour-long YouTube videos across diverse content types, including tutorials, product reviews, news, and podcasts. This precise localization of video chapters has the potential to streamline video search engines and enhance user experience across various online platforms.

Speed during Inference: On the MAD test set (average video length: 110 minutes), our most efficient variant processes queries in 2.6 seconds, achieving 11.9% R1@0.1. Our best-performing variant (17.3% R1@0.1) takes 5 seconds per query, comparable to the baseline speed.

Effciency during Training: ReVisionLLM converges in 1, and 5 epochs for VidChapters-7M and MAD, significantly faster than prior SOTA, RGNet[**?**] (35 epochs on MAD) and M-DETR[**?**] (50 epochs on VidChapters-7M).

Complicated Event Descriptions: ReVisionLLM is capable of locating complicated events with lengthy descriptions, For example, it successfully captures the event *"The jazz plays on as the movie cuts back and forth between the fast-moving someone, rushing down the SoHo streets, past a truck waiting for a light to change, past some pedestrians, some garbage cans, turning corners, and the strolling someone, walking up different streets, past different buildings."* from the MAD dataset, which the baseline method fails to locate.