



LiveStar: Live Streaming Assistant for Real-World Online Video Understanding

Zhenyu Yang^{1,2}, Kairui Zhang³, Yuhang Hu¹, Bing Wang¹, Shengsheng Qian^{1,2*},
Bin Wen⁴, Fan Yang⁴, Tingting Gao⁴, Weiming Dong^{1,2}, Changsheng Xu^{1,2,5}

¹Institute of Automation, Chinese Academy of Sciences, ²University of Chinese Academy of Sciences,

³ShanghaiTech University, ⁴Kuaishou Technology, ⁵Peng Cheng Laboratory

yangzhenyu2022@ia.ac.cn

Abstract

Despite significant progress in Video Large Language Models (Video-LLMs) for offline video understanding, existing online Video-LLMs typically struggle to simultaneously process continuous frame-by-frame inputs and determine optimal response timing, often compromising real-time responsiveness and narrative coherence. To address these limitations, we introduce LiveStar, a pioneering live streaming assistant that achieves always-on proactive responses through adaptive streaming decoding. Specifically, LiveStar incorporates: (1) a training strategy enabling incremental video-language alignment for variable-length video streams, preserving temporal consistency across dynamically evolving frame sequences; (2) a response-silence decoding framework that determines optimal proactive response timing via a single forward pass verification; (3) memory-aware acceleration via peak-end memory compression for online inference on 10+ minute videos, combined with streaming key-value cache to achieve 1.53× faster inference. We also construct an OmniStar dataset, a comprehensive dataset for training and benchmarking that encompasses 15 diverse real-world scenarios and 5 evaluation tasks for online video understanding. Extensive experiments across three benchmarks demonstrate LiveStar’s state-of-the-art performance, achieving an average 19.5% improvement in semantic correctness with 18.1% reduced timing difference compared to existing online Video-LLMs, while improving FPS by 12.0% across all five OmniStar tasks. Our model and dataset can be accessed at <https://github.com/yzy-bupt/LiveStar>.

1 Introduction

Recent advancements in Large Vision-Language Models (LVLMs) [1–5] have substantially propelled the development of Video Large Language Models (Video-LLMs) [6–10]. These cutting-edge models have been instrumental in pushing the boundaries of multimodal AI systems through innovations in spatial-temporal reasoning [11–13], memory-enhanced processing [14–16], and long-term video understanding [17–19], particularly enhancing capabilities in offline scenarios.

Building on these advances, researchers are increasingly exploring live streaming assistants for online video understanding [20–25], which remain always-on, reason through time, and provide real-time responses. However, existing online Video-LLMs [25–27] struggle to process frame-by-frame inputs and determine response timing, as shown in Fig 1(a). To effectively process continuous video streams and generate timely responses, such models must possess temporal decision-making capabilities—responding at contextually appropriate moments rather than producing repetitive outputs

*Corresponding author: shengsheng.qian@nlpr.ia.ac.cn

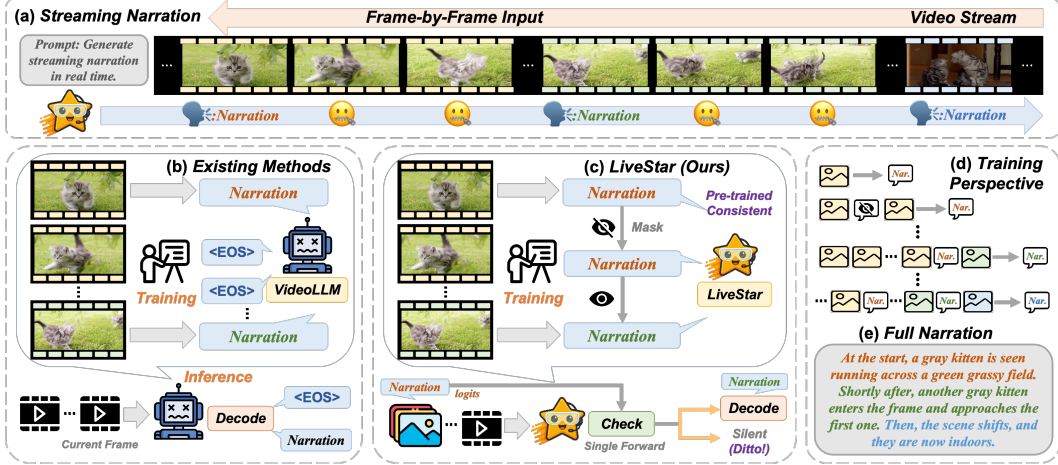


Figure 1: **Illustration of online video understanding.** (a) Taking the RNG task as an example, online video understanding requires Video-LLMs to handle continuous streams and output at appropriate times; (b) Existing methods overly rely on learning the EOS token, leading to poor inference performance; (c)-(e) LiveStar establishes an effective response-silence training and inference framework by SCAM and SVeD without compromising basic video understanding capabilities.

or default "I don't know" responses for each frame. To address this, VideoLLM-online [20] designs a streaming EOS (End-Of-Sequence) prediction mechanism (Fig. 1(b)). It ingests video streams, determines response timing based on user queries, and generates EOS tokens to mark silence intervals. Subsequent advancements like VideoLLM-MoD [21] and LION-FS [22] have further refined this EOS-based framework, achieving improved computational efficiency and response accuracy.

However, current online Video-LLMs exhibit excessive reliance on the EOS token, which leads to four critical issues: (1) **Response-Silence Imbalance**: Frames requiring EOS tokens as output vastly outnumber those generating normal responses [23]. For example, in a 1-minute video at 3 FPS with 5 response intervals, the response-to-silence ratio becomes 1:35. (2) **Consecutive Frame Inconsistency**: As shown in Fig. 1(b), adjacent visually similar frames may produce conflicting outputs—one generating a full narration while the other outputs only the EOS token. This inconsistency disrupts model convergence during fine-tuning. (3) **Pre-training Misalignment**: While pretraining aligns image-text pairs, silent states enforce a mapping from input frames to EOS tokens, contradicting the pretraining objective of meaningful visual-language correspondence. (4) **Vocabulary Confusion**: Since the EOS token is embedded in the vocabulary as a regular token, its frequent appearance in responses pollutes semantic coherence, introducing ambiguity and conflicting with normal outputs. These issues hinder model training effectiveness and degrade base video understanding capabilities, as the forced response-silence mechanism compromises the visual-language alignment. Motivated by this critical gap, we aim to address **Challenge 1**: How to establish an effective response-silence training and inference framework without compromising basic video understanding capabilities?

Moreover, online Video-LLMs face challenges due to limited diversity in training data and evaluation scope, failing to fully encompass the online applications encountered in real-world scenarios. For instance, prominent models [20–22] primarily focus on first-person video analysis through training on Ego4D [28]. Although StreamMind [23] extends its reach to sports scenarios with SoccerNet [29], its scope remains severely constrained. Recent efforts like SVBench [30], OVO-Bench [31], and StreamBench [32] have pioneered online evaluation by assessing models' ability to process videos synchronously during playback. However, these benchmarks focus exclusively on video question answering and lack coverage of diverse real-world tasks. In practice, beyond first-person devices, online scenarios span live streaming platforms, surveillance systems, and cinematic tools, requiring capabilities like real-time interactive editing [33, 34], streaming narration, and temporal grounding in continuous streams. Current evaluation benchmarks present three critical limitations: (1) **Narrow scenario coverage** limited to specific domains, (2) **Single-task evaluation** focused on video question answering, and (3) **Offline assessment** of temporal reasoning and response timing. These shortcomings motivate us to tackle **Challenge 2**: How to construct a comprehensive dataset and

benchmark encompassing diverse real-world scenarios and tasks for holistic evaluation of online video understanding?

To deal with the above challenges, we introduce **LiveStar**, an innovative live streaming assistant that enables online video understanding across diverse scenarios, delivering context-aware responses at semantically relevant moments through adaptive streaming decoding. For **Challenge 1**, we establish a streaming response-silence paradigm featuring two key innovations: (1) a streaming video-language alignment strategy using Streaming Causal Attention Masks (**SCAM**) during training, and (2) a Streaming Verification Decoding (**SVeD**) mechanism for inference. During training, we construct interleaved frame-caption sequences where revealed video frames progressively align with their corresponding captions through causal masked attention constraints (Fig. 1(d)), maintaining consistency with standard multimodal pre-training paradigms. For streaming inference, SVeD employs single-pass verification to determine optimal response timing: it suppresses redundant outputs by comparing current content with previous responses, maintaining narrative continuity through strategic silence. To handle extended video contexts, we implement a peak-end memory compression strategy that prioritizes salient keyframes and final captions, enabling efficient processing of long videos by focusing on semantically rich content while accelerating inference through streaming key-value cache. For **Challenge 2**, we introduce **OmniStar**, a dataset for training and benchmarking that encompasses diverse real-world scenarios and evaluation tasks in online video understanding. The dataset includes carefully annotated real-time narration, streaming QA, and streaming video grounding tasks on 20,137 videos, evaluated through an online setting that better simulates real-world conditions compared to traditional offline settings. Extensive experiments conducted on three benchmarks demonstrate that LiveStar achieves state-of-the-art results for online video understanding.

In summary, our contributions can be summarized as follows:

- We present **LiveStar**, an innovative live streaming assistant for online video understanding, capable of performing real-time comprehension on continuous video streams across diverse real-world scenarios and online tasks, while responding at contextually appropriate moments.
- We design a streaming video-language alignment training strategy using Streaming Causal Attention Masks (**SCAM**). By constructing interleaved frame-caption sequences, we train LiveStar to generate temporally consistent captions incrementally for variable-length streaming video inputs, maintaining narrative coherence across evolving frame prefixes.
- We propose Streaming Verification Decoding (**SVeD**), a response-silence decoding framework that determines optimal response timing via a single forward pass verification. Furthermore, SVeD mimics human-like memory consolidation through a peak-end rule-based strategy, distilling long video contexts by prioritizing salient frames to accelerate processing.
- We introduce **OmniStar**, a comprehensive dataset for training and benchmarking that encompasses 15 diverse real-world scenarios and 5 evaluation tasks in online video understanding. Extensive experiments across three benchmarks demonstrate LiveStar’s state-of-the-art performance, achieving an average 19.5% improvement in semantic correctness with 18.1% reduced timing difference compared to existing online Video-LLMs across all five OmniStar tasks.

2 Related Work

Video Large Language Models Recent advancements in Large Language Models (LLMs) [35–39] have paved the way for a significant research focus on Video-LLMs [6–10], particularly for enhancing video understanding tasks such as video captioning [40–42], video question answering [43–45] and temporal grounding [46–48] in offline settings. Currently, in addition to the popular closed-source LVLMs such as GPT-4V [49], GPT-4o [37], and Gemini 1.5 Pro [50], an increasing number of open-source Video-LLMs, including LLaVA-NeXT-Video [51], Video-LLaVA [52], and VILA [53], have also demonstrated impressive capabilities in these tasks. Despite these advances, achieving seamless human-computer interaction demands capabilities beyond static video analysis. Specifically, it must process real-time online videos while exhibiting temporal decision-making—responding at contextually appropriate moments and synthesizing histories to maintain coherence [20, 25]. Current Video-LLMs, however, remain limited in handling the dynamic complexities of continuous streaming video processing and often struggle to fully capture the nuances of real-world contexts.

Video-LLMs for Online Video Understanding Recently, there has been growing interest in online video understanding [25, 20], driven by the availability of diverse streaming video sources, including online video platforms [54], live streaming services [55], and wearable camera footage [28]. These resources enable Video-LLMs to learn from real-world contexts, enhancing their ability to interact with streaming content. For instance, VideoStreaming [25] is designed to understand video content in real-time through streaming encoding and memory selection. However, existing models [25–27] struggle to determine response timing and process frame-by-frame inputs. To address this, VideoLLM-online [20] introduces a streaming EOS (End-Of-Sequence) prediction mechanism, which continuously ingests video streams and generates EOS tokens to indicate periods requiring silence. Subsequent models [21, 22] refine this framework, improving computational efficiency and response accuracy. Despite this, excessive reliance on EOS tokens disrupts training and performance. This imbalanced dependency [23] causes most frames to be suppressed into silent outputs, while visually similar adjacent frames produce conflicting predictions (e.g., alternating between captions and silence). Furthermore, it misaligns with pretraining objectives that prioritize visual-language alignment [56, 57], as silent states incentivize trivial mappings rather than meaningful understanding. To bridge this gap, we introduce a live streaming assistant designed to perform real-time understanding of continuous video streams, delivering responses at contextually appropriate moments while maintaining the core capabilities of its pre-trained video-language alignment.

Online Video Understanding Benchmarks The exponential growth of video data has elevated video understanding to a critical field. To rigorously evaluate model capabilities, researchers have established a range of standardized benchmarks [44, 58–68] for offline Video-LLMs. While these benchmarks enable comparative evaluations and drive progress in Video-LLMs, they primarily rely on complete videos or individual clips for offline analysis. Recent efforts like SVBench [30], OVO-Bench [31], and StreamBench [32] have pioneered online evaluation by assessing models’ ability to process videos synchronously during playback. However, these benchmarks focus exclusively on video question answering and lack coverage of diverse real-world scenarios. Current Video-LLM evaluations remain constrained, with most models primarily evaluated on the first-person Ego4D dataset [28]. Therefore, we introduce a comprehensive dataset for training and benchmarking that encompasses diverse real-world scenarios and evaluation tasks in online video understanding.

3 Method: LiveStar

3.1 Training Strategy

We design a streaming training strategy for aligning streaming video and language using Streaming Causal Attention Masks (**SCAM**). By constructing interleaved frame-caption sequences, we train LiveStar to incrementally generate temporally consistent captions for streaming video inputs of varying lengths, maintaining narrative coherence across evolving prefixes as the frames progress.

Streaming Video-Language Alignment Current Video-LLMs primarily rely on static vision-language foundation models [1, 2, 69] pre-trained on image-text pairs, exhibiting limited adaptation to streaming video-text relationships [10]. These models typically optimize the objective by image/video-text pair alignment:

$$\max P([T x t_i] \mid [I m g_i]/[V i d_i]), \quad (1)$$

which fails to address two critical challenges in online video understanding: (1) incremental processing of streaming frames, and (2) dynamic alignment between evolving visual contexts and linguistic outputs. To address them, we propose multi-turn, frame-by-frame instruction tuning that bridges streaming video inputs with language generation through the reformulated objective:

$$\max P([T x t^k] \mid [C t x^{< t_i}], [F r m^{t_i}]), \forall t_i \in C_k, \quad (2)$$

where $[F r m^{t_i}]$ denotes the video frame at timestamp t_i , $[C t x^{< t_i}]$ represents the accumulated multi-modal context, and $C_k = \{t_i\}_{i=m}^n$ indicates a **semantic clip** (frames from t_m to t_n within the same scene/event/action) sharing semantic text $[T x t^k]$. This addresses the pretraining misalignment in existing EOS-based models [20–23] that enforce end-of-sequence token prediction (i.e., $\max P(\text{EOS} \mid [C t x^{< t_i}], [F r m^{t_i}])$) across non-response frames. Unlike these models, LiveStar achieves streaming consistency through incremental context integration over variable-length video streams. Our method maintains narrative coherence by dynamically updating frame prefixes while preserving evolving visual contexts, establishing *a new paradigm* for streaming video understanding.

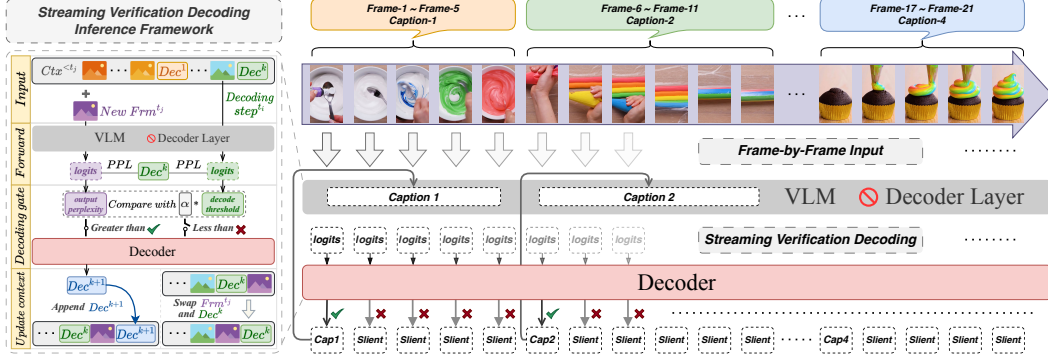


Figure 2: **Overview of the streaming verification decoding (SVeD) inference framework:** A dynamic response-silence decoding framework designed to determine optimal response timing for online video understanding.

Interleaved Frame-Caption Sequences To enable multi-turn, frame-by-frame instruction tuning, we construct interleaved frame-caption sequences using a chat-inspired format, which allows LiveStar to incrementally ingest visual inputs while preserving temporal awareness for accurate scene transition detection. Each conversational turn includes: (1) a frame $[Frm^k]$ (at timestep t_i) and (2) a corresponding caption $[Cap^k]$ (for the k -th semantic clip). Consecutive frames from the same semantic clip share captions with identical semantics. To mitigate overfitting from repeated caption exposure, we randomly sample a caption $[Cap_j^k]$ from a pool of M paraphrased captions.

Streaming Causal Attention Masks To train LiveStar autoregressively on interleaved frame-caption sequences, we address three challenges: (1) Preventing leakage: Already-generated captions within the current semantic clip must be masked when generating the current caption to prevent trivial copying, as they are identical in content. (2) Token-specific context: During autoregression, the model must retain visibility into previously predicted tokens for the current caption to maintain coherence. (3) Scene transition signaling: The last caption of each semantic clip must persist across subsequent frames to explicitly demarcate semantic boundaries. To address these, we propose Streaming Causal Attention Masks (SCAM) (see Fig. 3 for the mask matrix), replacing standard causal attention masks. Formally, when generating the caption for frame Frm^{t_i} in the clip C_k , the model is constrained to attend only to all video frames from the preceding clips $\{C_1, C_2, \dots, C_{k-1}\}$ and the last captions of those clips. The final optimization objective is as follows:

$$\max P([Cap_j^k] \mid [Ctx^{<t_i} \{Mask^{\leq t_i}\}], [Frm^{t_i}]), \forall t_i \in C_k. \quad (3)$$

Here, the mask matrix $Mask^{\leq t_i}$ is designed to block attention to all non-terminal caption tokens from semantic clips $\{C_1, C_2, \dots, C_k\}$ prior to timestep t_i , as illustrated in Fig. 1(d).

3.2 Inference Framework

While LiveStar can generate real-time captions at each timestep, determining when to update the response while maintaining content coherence and avoiding redundant outputs remains a critical challenge. To address this, we propose a dynamic response-silence decoding (SVeD) framework (see Fig. 2) and employ memory-aware techniques for acceleration.

Streaming Verification Decoding We present SVeD, a dynamic response-silence decoding framework designed to determine optimal response timing for online video understanding. It intro-

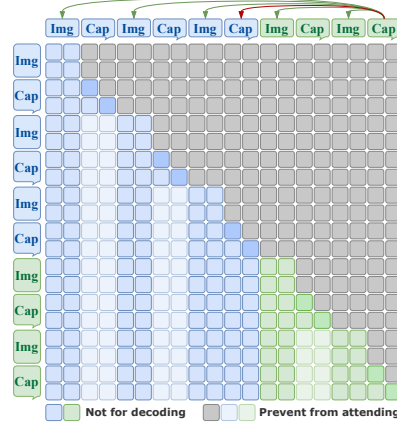


Figure 3: Mask matrix of SCAM.

duces a *decoding gate* that selectively triggers caption generation based on a streaming verification mechanism. At each triggered decoding step t_i , we compute the perplexity of the generated caption $[Dec]$ as: $PPL^{t_i}([Dec]) = \sqrt[N]{1/P([Dec] | [Ctx^{<t_i}], [Frm^{t_i}])}$, where N represents the token count of $[Dec]$, and $P(\cdot)$ is the autoregressive probability derived from token logits. For each incoming frame $[Frm^{t_j}]$, SVeD performs a single forward pass to verify the latest caption’s validity by recomputing $PPL^{t_j}([Dec])$. If $PPL^{t_j}([Dec]) > \alpha \cdot PPL^{t_i}([Dec])$ (where α is a tunable scaling factor), the gate activates decoding at t_j , generating an updated caption. Otherwise, move $[Dec]$ to the end of $[Ctx]$ without decoding, preserving temporal coherence while minimizing latency. Under the same model architecture, SVeD achieves faster inference than decoding an EOS token to indicate a silent response. This lightweight verification step ensures adaptive response timing, balancing accuracy and efficiency. For details, refer to Alg. 1.

Peak-End Memory Compression Modern streaming videos often span hours with high frame rates, posing computational challenges for long-term understanding. Inspired by the Peak-End cognitive rule [70–72]—where human memory retention prioritizes salient moments (keyframes) and recent experiences (summaries)—we propose a memory compression framework tailored for 10+ minute video analysis at 3 fps. Our method leverages two critical signals: (1) *Keyframe detection*: We have computed the perplexity $PPL^{t_i}([Dec])$ for each frame, where lower values indicate higher semantic importance. (2) *Temporal summarization*: The final frame’s caption of each semantic clip encapsulates event semantics. To optimize memory, we probabilistically prune frames older than a window W , with deletion likelihood proportional to both the relative PPL within its semantic clip and elapsed time. Experiments demonstrate that it achieves optimal semantic accuracy (SemCor) and minimal timing difference (TimDiff) compared to Uniform Dropout and FIFO Forgetting, as shown in Tab. 4.

Algorithm 1: Streaming Verification Decoding

Input: Video frame stream $\{[Frm^t]\}_{t=1}^T$
Output: Dynamically generated caption $[Dec]$
Initialize $[Dec], [Ctx] \leftarrow \emptyset$
Initialize reference timestamp $t_i \leftarrow 0$
for each incoming frame $[Frm^{t_j}]$ **do**
 Append $[Frm^{t_j}]$ to $[Ctx]$
 if $[Dec] \neq \emptyset$ **then**
 Compute verification perplexity:
 $PPL^{t_j}([Dec]) = \sqrt[N]{\frac{1}{P([Dec] | [Ctx^{\leq t_j}])}}$
 if $PPL^{t_j}([Dec]) > \alpha PPL^{t_i}([Dec])$ **then**
 Activate decoding: Generate new
 tokens $[Dec]_{new}$ using $[Ctx^{\leq t_j}]$
 Update $[Dec] \leftarrow [Dec]_{new}$
 Append $[Dec]$ to $[Ctx]$
 $t_i \leftarrow t_j$
 else
 Swap the last two elements in $[Ctx]$
 // Move $[Dec]$ to the end
 else
 Perform initial decoding to generate $[Dec]$
 Append $[Dec]$ to $[Ctx]$
 $t_i \leftarrow t_j$

Streaming Key-Value Cache. We implement an effective streaming key-value (KV) cache mechanism for multi-turn conversational interactions. We introduce a dual-level caching architecture that maintains both intra-dialogue KV caches for frame-level processing and inter-dialogue streaming caches for long-context preservation across conversations. This eliminates redundant recomputation of historical context representations (keys and values) during new video frame processing while ensuring low-latency inference. It specifically addresses two critical challenges: (1) Cache sequence integrity under swap operations in our SVeD module, and (2) Dynamic length adaptation through Peak-End Memory Compression that strategically prunes redundant tokens while preserving temporal coherence. Experiments (see Tab. 4) demonstrate that it achieves 1.53× faster inference than methods without KV caching while incurring negligible performance loss on video streams.

4 Dataset: OmniStar

Datasets Statistics. We introduce OmniStar, a comprehensive online video understanding dataset featuring 15 diverse real-world scenarios (46 specific categories) and 5 online evaluation tasks. The dataset comprises 20,137 expert-annotated video streams with temporally dense annotations, rigorously split into 19,137 training and 1,000 evaluation instances (200 per task) without overlap. It establishes five synergistic evaluation tasks through a unified streaming protocol: (1) Real-time narration generation [RNG], (2) Online temporal grounding [OTG], (3) Frame-level dense QA [FDQ], (4) Contextual online QA [COQ], and (5) Multi-turn interactive QA [MIQ]. More detailed online tasks are described in detail in App. D. The 15 scenarios include: Travel & Events, Sports, Pets & Animals, Music, Autos & Vehicles, Film & Animation, Nonprofits & Activism, Science &

Technology, Education, Howto & Style, News & Politics, Entertainment, Comedy, People & Blogs, and Gaming, each subdivided into 2-4 fine-grained categories using YouTube’s native annotation system.

Temporally Dense Annotation. In contrast to prior online video understanding benchmarks, which are typically limited to single question-answering task and lack coverage of real-world scenarios [30–32], OmniStar adopts a semi-automated, temporally dense annotation pipeline to obtain denser labeled data. Every video is annotated with multiple caption segments, where each segment is precisely aligned with its corresponding start and end timestamps. Instead of providing disjointed descriptions, OmniStar’s annotations are constructed as narrative-consistent caption sequences that collectively form coherent storylines (e.g., *"First, the woman gathers the ingredients..."* → *"Then, she mixes them in a large bowl..."* → *"Finally, she places the dish in the oven..."*), thereby facilitating the RNG task by effectively modeling long-range narrative flow.

Building on this foundation, OmniStar introduces two forms of streaming QA supervision: (1) temporally distributed multi-turn QA chains (Task: MIQ, COQ), where earlier and later questions and answers are semantically linked, and (2) frame-level streaming QA (Task: FDQ), where a standing query corresponding to **time-varying answers** (e.g., *Q: "What is the person showing?"* → *"A pen."* → *"A book."* → *"A smartphone."*). Both supervision paradigms are designed to move beyond static, isolated QA settings by encouraging models to maintain temporal awareness, integrate past context, and update answers according to the video streams. Case examples are listed in Tabs. 7, 8, and 9.

5 Experiments

5.1 Experimental Setup

Datasets. We validate LiveStar’s effectiveness across offline and online settings, including Ego4D [28] (egocentric narration stream), SVBench [30] (streaming QA) and OmniStar (5 real-world online tasks). We build the Ego4D Narration Stream benchmark using dense timestamped narrations, following VideoLLM-online [20]. We leverage SVBench’s evaluation set for dialogue and streaming QA. For real-world online evaluation, we benchmark Video-LLMs on OmniStar through five online tasks, with real-time narration generation assessed in both offline and online settings. We design a two-phase progressive training paradigm for LiveStar, detailed in App. A.2.

Implementation Details. We conducted full fine-tuning of LiveStar on 8× NVIDIA A800 GPUs. Building upon InternVideo2.5 [10], the model consists of a vision encoder (InternViT [1]), an MLP projector, and a large language model (InternLM2.5-7B [73]). For the vision encoder, we employ InternViT to extract video frame embeddings at a rate of 1-4 FPS, with each frame represented by 16 tokens. To optimize efficiency in online video understanding, we fine-tuned the model using a static resolution strategy, enabling it to process multi-minute video content within an 8K-token context window. The extracted frame embeddings are then fed into an MLP projector to generate frame tokens, following the approach used in LLaVA-1.5 [74]. These frame tokens are interleaved with language tokens and input into the large language model, InternLM2.5. Additional details about model architecture and model training can be found in App. A.1.

Parameter Settings. During training, we conducted full fine-tuning of LiveStar using a total of 83K data, which includes the OmniStar training set. We trained the models for 1 epoch with a learning rate of 4×10^{-5} using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.05), a per-device batch size of 1, and gradient accumulation over 4 steps to achieve an effective batch size of 32. We adopted cosine learning rate scheduling with a warmup ratio of 0.03. Input frames were uniformly resized to 448×448 , with a patch downsampling ratio of 0.5. The vision encoder was frozen during training, while the MLP projector and language model components were fully updated. Each training sequence contains up to 8192 tokens, consisting of interleaved frame and language tokens following the InternVL2.5 conversational template. We optimized the model using the standard autoregressive cross-entropy loss computed over the language tokens, where loss was computed only on assistant response tokens, and inter-frame language segments were excluded via our SCAM strategy. For inference, the tunable scaling factor in SVeD was set to 1.03 by default, the prune window W of streaming video-language alignment was set to $M=1$ by default for better temporal alignment.

Table 1: Evaluation results of various models on OmniStar-RNG in offline and online evaluation. PPL is a commonly used metric for narration, but due to the different vocabulary sizes of LLMs, it is only for reference. "-" indicates the inability to perform the required test.

Method	Offline Evaluation (Fixed decoding)				Online Evaluation (Response-silence decoding)				
	PPL↓	TokAcc↑	SemCor↑	SumFluen↑	TimDiff↓	TimRedun↓	TimCover↑	SemCor↑	SumFluen↑
Human	-	-	6.73	7.17	1.08	1.24	0.84	6.09	6.81
Offline Video-LLMs/LVLMs									
GPT-4V	-	-	4.97	5.37	-	-	-	-	-
GPT-4o	-	-	5.03	5.45	-	-	-	-	-
LLaVA-Video	12.42	0.53	3.40	2.88	-	-	-	-	-
InternVideo2.5	6.91	0.56	4.32	3.61	-	-	-	-	-
InternVL2.5	9.81	0.51	3.40	2.94	-	-	-	-	-
MiniCPM-V 2.6	9.46	0.57	4.34	4.13	-	-	-	-	-
Qwen2.5-VL	13.80	0.59	4.42	4.24	-	-	-	-	-
Online Assistants									
VideoLLM-online	9.73	0.49	3.01	0.69	2.67	2.15	0.80	1.68	0.59
VideoLLM-MoD	9.93	0.48	2.89	0.65	2.54	2.49	0.90	1.66	0.55
MMDuet	5.69	0.60	4.29	3.40	2.32	0.62	0.51	1.93	2.69
LiveStar (Ours)	5.14	0.62	4.62	4.55	1.91	0.95	0.71	3.19	4.25

5.2 Online Experiments

To simulate real-world scenarios of online video understanding, we evaluated Video-LLMs during the inference phase using the OmniStar test set. Unlike the Ego4D Narration Stream benchmark, which is offline due to the absence of scoring for inference results and fixed decoding points (as is the case with SVBench), our OmniStar allows Video-LLMs to determine the response-silence timing across five tasks in an online setting. This evaluation not only assesses the accuracy of the output timing but also examines whether the output quality aligns with the ground truth.

Evaluation Metrics. We use the following metrics to evaluate the model as an online video assistant:

- **TimDiff (Timing Difference)** quantifies the absolute time difference between model responses and ground-truth annotations (missing outputs are penalized with full scene duration; multiple outputs are penalized by cumulative time), evaluating real-time response precision. Lower values indicate better synchronization with visual events.
- **TimRedun (Timing Redundancy)** measures redundancy through the average number of unnecessary responses per scene, where a perfect single response could fully capture the scene content.
- **TimCover (Timing Coverage)** computes the average coverage of responses across all scenes, where a scene is scored 1 if it contains at least one valid response and 0 otherwise.
- **SemCor (Semantic Correctness)** scores semantic alignment between responses and ground truth using GPT-4o across three axes: (1) Semantic Accuracy, (2) Language Quality, and (3) Information Completeness. The final SemCor averages all dimensions (0–10 each). See App. E for details.
- **SumFluen (Summarize Fluency)** evaluates holistic fluency of concatenated responses against aggregated ground truth using GPT-4o, with scores for (1) Writing Logicality, (2) Language Fluency, (3) Writing Conciseness, (4) Semantic Consistency, and (5) Narrative Completeness.

Results. The right side of Tab. 1 presents the results of various models on OmniStar-RNG in online evaluation (under each model’s own response-silence strategy), indicating that LiveStar achieves lower response latency and higher semantic precision. Notably, VideoLLM-online and VideoLLM-MoD output on nearly every frame, leading to the highest TimCover, but this aggressive strategy also results in poorer performance on other metrics. In contrast, MMDuet produces much sparser outputs, resulting in the lowest TimRedun, but at the cost of lower scores on other metrics. These deficiencies are clearly

Table 2: Evaluation results of online Video-LLMs on OmniStar. "-" in OTG means no generation needed for scoring; "-" in COQ and MIQ indicates real-time QA. Note that VideoLLM-online did not achieve the reported 10 FPS.

Method	Online Evaluation (SemCor↑/TimDiff↓)					FPS↑ (5min)
	RNG	OTG	FDQ	COQ	MIQ	
Human	6.09/1.08	-/1.81	9.12/1.01	7.96/-	7.83/-	-
VideoLLM-online	1.68/2.67	-/9.69	2.35/2.15	4.01/-	3.83/-	3.37
VideoLLM-MoD	1.66/2.54	-/9.83	2.11/2.23	3.99/-	3.75/-	3.41
MMDuet	1.63/2.32	-/4.42	4.78/2.65	5.71/-	5.62/-	0.91
LiveStar (Ours)	3.19/1.91	-/3.57	6.44/1.80	5.85/-	5.78/-	3.82

the lowest TimRedun, but at the cost of lower scores on other metrics. These deficiencies are clearly

demonstrated in Tab. 2, where LiveStar consistently outperforms all other online Video-LLMs across 5 tasks of the OmniStar benchmark while maintaining the fastest inference speed-with a 19.5% increase SemCor score, an 18.1% reduction TimDiff value, and a 12.0% increase in FPS compared to the second-best model.

5.3 Offline Experiments

Unlike online evaluation where models *autonomously determine response timing*, our offline evaluation on Ego4D, OmniStar-RNG, and SVBench pre-specifies decoding timing. For fairness, we strictly adopt the evaluation protocol and metrics from previous works [20–22] despite their non-generative constraints (i.e., ground-truth verification only). On OmniStar, we extend beyond standard perplexity and token accuracy verification for ground truth to implement full model generation with online scoring – a critical advancement enabling comprehensive online capability assessment.

Evaluation Metrics. For rigorous benchmarking, we adopt established offline metrics: PPL, TokAcc (Token Accuracy, previously referred to as LM-Correctness), TimeDiff, and Fluency from prior works [20–22]. Additionally, we implement online generation evaluation with offline prescribed timing, measured through SemCor and SumFluen metrics, both scored based on GPT-4o.

Table 3: Evaluation results of online Video-LLMs on Ego4D Narration Stream benchmark in offline settings. "-" indicates the model lacks EOS tokens for Fluency.

Method	PPL↓	TimeDiff↓	Fluency↑	TokAcc↑
VideoLLM-online	2.43	2.04	45.1%	48.1%
VideoLLM-MoD	2.41	2.04	45.2%	48.9%
LION-FS	2.09	2.15	46.1%	52.4%
MMDuet	4.51	1.97	-	39.3%
LiveStar (Ours)	1.97	1.76	-	61.1%

Results. The left side of Tab. 1 presents the results of various models on OmniStar-RNG in offline evaluation (given decoding timing conditions), showing that LiveStar surpasses all other online assistants and open-source offline Video-LLMs/LVLMs across all metrics, though there remains a performance gap compared to human performance and GPT-4V/4o. Tab. 3 demonstrates that LiveStar comprehensively outperforms all other online assistants on the Ego4D Narration benchmark, particularly achieving an 8.7% higher TokAcc than the second-best LION-FS. Experimental results and analyses for dialogue and streaming evaluations on SVBench can be found in App. C.1.

5.4 Ablation Study

Impact of Response-Silence Threshold. LiveStar employs a dynamic response-silence decoding framework governed by an adaptive threshold mechanism. The decoding threshold is defined as $\alpha \cdot \text{PPL}^{t_i}([Dec])$, where $\alpha \geq 1$ is a tunable scaling factor applied to the timestep-specific perplexity. To rigorously evaluate the parameter sensitivity of α , we conducted an empirical analysis over the interval $[1.0, 1.1]$ using the OmniStar-RNG benchmark. Our online evaluation (see Fig. 4) demonstrates α 's critical role in balancing timing difference, timing redundancy, and timing coverage. Controlled tests revealed optimal performance within the narrow range $\alpha = 1.02$ – 1.04 , with $\alpha = 1.03$ selected as the final configuration.

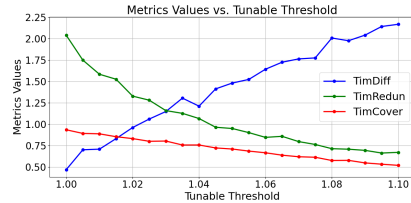


Figure 4: Ablation study on the impact of response-silence threshold.

Strategy for Inference Acceleration. Tab. 4 presents ablation studies on memory dropout and KV cache strategies for inference acceleration. For memory dropout mechanisms: (1) Uniform dropout causes a 4.70% SemCor degradation due to critical recent frames being discarded; (2) FIFO forgetting impairs temporal reasoning by discarding historical event captions, increasing TimDiff by 9.42% with 3.76% SemCor reduction; (3) Our Peak-End Memory Compression preserves semantic clip summaries and keyframes through precomputed PPL dropout of non-critical frames, achieving optimal SemCor and minimal TimDiff. Regarding KV cache: While removing both inter/intra-dialog caching or intra-dialog caching shows negligible performance loss, adopting both strategies boosts

Table 4: Ablation study on strategies for inference acceleration on OmniStar-RNG.

Dropout	KV Cache	SemCor↑	TimDiff↓	FPS↑
Uniform	Both	3.04	2.01	3.77
FIFO	Both	3.07	2.09	3.91
Peak-End	Neither	3.19	1.95	2.50
	w/o Inter-Dialog	3.17	1.87	2.92
	Both	3.19	1.91	3.82

FPS by 1.53×/1.31× respectively under 5-minute-videos inference by eliminating redundant recomputation of historical key-value representations during new frame processing, enabling low-latency inference without quality degradation.

Size of Paraphrased Caption Pool. To mitigate overfitting induced by repeated caption exposure during training, we employ a stochastic sampling approach that selects captions $[Cap_j^k]$ from a paraphrased caption pool of size M . Experiments reveal a critical trade-off between semantic richness and temporal consistency: Increasing M enhances semantic correctness (SemCor), with $M = 3$ achieving a +1.57% improvement over $M = 1$, but simultaneously degrades temporal alignment (TimDiff) by +3.14%. While larger pools better capture semantic diversity, the incurred temporal inconsistency becomes prohibitive for real-time applications. Given the heightened sensitivity of temporal alignment in online tasks, we adopt $M = 1$ as the default configuration. This framework allows task-aware adaptation, enabling practitioners to optimize the pool size according to their specific robustness-latency requirements.

Table 5: Ablation study on the size of the caption pool.

Pool Size	SemCor↑	TimDiff↓
M=1	3.19	1.91
M=2	3.23	1.98
M=3	3.24	1.97

6 Case Study

We conduct qualitative comparisons across multiple tasks, including RNG, MIQ, COQ, and FDQ, between our LiveStar model and existing representative online video understanding models, VideoLLM-online and MMDuet [24], with details shown in Figs. 5, 6 and 7. These cases demonstrate that VideoLLM-online and MMDuet often suffer from limited contextual understanding, hallucinations, and insufficient fine-grained recognition, while LiveStar consistently provides more accurate, grounded, and timely responses. These comparisons collectively highlight the advantages of LiveStar and reveal the limitations of current LVLMs in handling fine-grained and context-sensitive video question answering.

7 Conclusion

This paper introduces LiveStar, an innovative live streaming assistant that pioneers always-on proactive responsiveness through adaptive streaming decoding. We establish a streaming response-silence paradigm featuring two technical contributions: (1) a streaming video-language alignment framework utilizing Streaming Causal Attention Masks (SCAM) during model training, and (2) a Streaming Verification Decoding (SVeD) mechanism for real-time inference optimization. Through extensive experiments across three benchmark datasets, LiveStar demonstrates state-of-the-art performance in online video understanding while achieving practical deployment capabilities of processing 10+ minute video streams at 3 FPS. This paper significantly advances the field by introducing a novel response-silence paradigm and establishing a comprehensive benchmark, thereby stimulating future research in developing advanced models for complex online video understanding tasks.

Limitations. Despite the superior performance of our proposed LiveStar on several benchmarks for online video understanding, there remain some limitations: First, to improve inference efficiency, we compress each video frame into 16 visual tokens. While this compact representation significantly reduces computational cost, it inevitably compromises the model’s ability to capture fine-grained visual details. As a result, the model may miss critical cues in scenarios involving subtle motion changes or complex scene dynamics. Second, the current version of LiveStar only supports vision-text modalities and does not incorporate audio information. This limitation constrains the model’s capacity to fully leverage multimodal cues in video understanding tasks. In future work, we aim to extend the model to include audio modalities for more comprehensive multimodal reasoning.

Acknowledgment

We would like to present our appreciation to the anonymous reviewers and ACs for their constructive suggestions. This work is supported by the National Key Research and Development Program of China (No.2023YFC3310700), the Beijing Natural Science Foundation (JQ23018, L252032), and the National Natural Science Foundation of China (No.62036012, 62276257).

References

- [1] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu *et al.*, “Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling,” *arXiv preprint arXiv:2412.05271*, 2024.
- [2] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” *arXiv preprint arXiv:2409.12191*, 2024.
- [3] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, H. Li, W. Zhao, Z. He *et al.*, “Minicpm-v: A gpt-4v level mllm on your phone,” *arXiv preprint arXiv:2408.01800*, 2024.
- [4] P. Zhang, X. Dong, Y. Zang, Y. Cao, R. Qian, L. Chen, Q. Guo, H. Duan, B. Wang, L. Ouyang *et al.*, “Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output,” *arXiv preprint arXiv:2407.03320*, 2024.
- [5] T. GLM, A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Zhang, D. Rojas, G. Feng, H. Zhao *et al.*, “Chatglm: A family of large language models from glm-130b to glm-4 all tools,” *arXiv preprint arXiv:2406.12793*, 2024.
- [6] K. Ataallah, X. Shen, E. Abdelrahman, E. Sleiman, D. Zhu, J. Ding, and M. Elhoseiny, “Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens,” *arXiv preprint arXiv:2404.03413*, 2024.
- [7] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, “Video-chatgpt: Towards detailed video understanding via large vision and language models,” *arXiv preprint arXiv:2306.05424*, 2023.
- [8] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao, “Videochat: Chat-centric video understanding,” *arXiv preprint arXiv:2305.06355*, 2023.
- [9] A. Yang, A. Nagrani, P. H. Seo, A. Miech, J. Pont-Tuset, I. Laptev, J. Sivic, and C. Schmid, “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*, 2023, pp. 10 714–10 726.
- [10] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang *et al.*, “Internvideo: General video foundation models via generative and discriminative learning,” *arXiv preprint arXiv:2212.03191*, 2022.
- [11] Z. Cheng, S. Leng, H. Zhang, Y. Xin, X. Li, G. Chen, Y. Zhu, W. Zhang, Z. Luo, D. Zhao *et al.*, “Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms,” *arXiv preprint arXiv:2406.07476*, 2024.
- [12] Z. Liu, Y. Dong, Z. Liu, W. Hu, J. Lu, and Y. Rao, “Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution,” *arXiv preprint arXiv:2409.12961*, 2024.
- [13] S. Ren, L. Yao, S. Li, X. Sun, and L. Hou, “Timechat: A time-sensitive multimodal large language model for long video understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 313–14 323.
- [14] H. Zhang, Y. Wang, Y. Tang, Y. Liu, J. Feng, J. Dai, and X. Jin, “Flash-vstream: Memory-based real-time understanding for long video streams,” *arXiv preprint arXiv:2406.08085*, 2024.
- [15] E. Song, W. Chai, T. Ye, J.-N. Hwang, X. Li, and G. Wang, “Moviechat+: Question-aware sparse memory for long video question answering,” *arXiv preprint arXiv:2404.17176*, 2024.
- [16] B. He, H. Li, Y. K. Jang, M. Jia, X. Cao, A. Shah, A. Shrivastava, and S.-N. Lim, “Ma-lmm: Memory-augmented large multimodal model for long-term video understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 504–13 514.
- [17] X. Wang, D. Song, S. Chen, C. Zhang, and B. Wang, “Longllava: Scaling multi-modal llms to 1000 images efficiently via hybrid architecture,” 2024. [Online]. Available: <https://arxiv.org/abs/2409.02889>
- [18] F. Xue, Y. Chen, D. Li, Q. Hu, L. Zhu, X. Li, Y. Fang, H. Tang, S. Yang, Z. Liu, Y. He, H. Yin, P. Molchanov, J. Kautz, L. Fan, Y. Zhu, Y. Lu, and S. Han, “Longvila: Scaling long-context visual language models for long videos,” *null*, 2024.
- [19] P. Zhang, K. Zhang, B. Li, G. Zeng, J. Yang, Y. Zhang, Z. Wang, H. Tan, C. Li, and Z. Liu, “Long context transfer from language to vision,” *arXiv preprint arXiv:2406.16852*, 2024.

- [20] J. Chen, Z. Lv, S. Wu, K. Q. Lin, C. Song, D. Gao, J.-W. Liu, Z. Gao, D. Mao, and M. Z. Shou, “Videollm-online: Online video large language model for streaming video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 407–18 418.
- [21] S. Wu, J. Chen, K. Q. Lin, Q. Wang, Y. Gao, Q. Xu, T. Xu, Y. Hu, E. Chen, and M. Z. Shou, “Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 109 922–109 947, 2024.
- [22] W. Li, B. Hu, R. Shao, L. Shen, and L. Nie, “Lion-fs: Fast & slow video-language thinker as online video assistant,” *arXiv preprint arXiv:2503.03663*, 2025.
- [23] X. Ding, H. Wu, Y. Yang, S. Jiang, D. Bai, Z. Chen, and T. Cao, “Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition,” *arXiv preprint arXiv:2503.06220*, 2025.
- [24] Y. Wang, X. Meng, Y. Wang, J. Liang, J. Wei, H. Zhang, and D. Zhao, “Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format,” *arXiv preprint arXiv:2411.17991*, 2024.
- [25] R. Qian, X. Dong, P. Zhang, Y. Zang, S. Ding, D. Lin, and J. Wang, “Streaming long video understanding with large language models,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 119 336–119 360, 2024.
- [26] X. Zhou, A. Arnab, S. Buch, S. Yan, A. Myers, X. Xiong, A. Nagrani, and C. Schmid, “Streaming dense video captioning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 243–18 252.
- [27] H. Xiong, Z. Yang, J. Yu, Y. Zhuge, L. Zhang, J. Zhu, and H. Lu, “Streaming video understanding and multi-round interaction with memory-enhanced knowledge,” *arXiv preprint arXiv:2501.13468*, 2025.
- [28] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, “Ego4d: Around the world in 3,000 hours of egocentric video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 995–19 012.
- [29] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, “Soccernet: A scalable dataset for action spotting in soccer videos,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1711–1721.
- [30] Z. Yang, Y. Hu, Z. Du, D. Xue, S. Qian, J. Wu, F. Yang, W. Dong, and C. Xu, “Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding,” *arXiv preprint arXiv:2502.10810*, 2025.
- [31] Y. Li, J. Niu, Z. Miao, C. Ge, Y. Zhou, Q. He, X. Dong, H. Duan, S. Ding, R. Qian *et al.*, “Ovo-bench: How far is your video-llms from real-world online video understanding?” *arXiv preprint arXiv:2501.05510*, 2025.
- [32] H. Xiong, Z. Yang, J. Yu, Y. Zhuge, L. Zhang, J. Zhu, and H. Lu, “Streaming video understanding and multi-round interaction with memory-enhanced knowledge,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.13468>
- [33] Z. Yang, D. Xue, S. Qian, W. Dong, and C. Xu, “Ldre: Llm-based divergent reasoning and ensemble for zero-shot composed image retrieval,” in *Proceedings of the 47th International ACM SIGIR conference on research and development in information retrieval*, 2024, pp. 80–90.
- [34] Z. Yang, S. Qian, D. Xue, J. Wu, F. Yang, W. Dong, and C. Xu, “Semantic editing increment benefits zero-shot composed image retrieval,” in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 1245–1254.
- [35] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [36] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, “Gemini: a family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [37] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [38] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

- [39] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, “Improving language understanding by generative pre-training,” *Unknown*, 2018.
- [40] L. Chen, X. Wei, J. Li, X. Dong, P. Zhang, Y. Zang, Z. Chen, H. Duan, Z. Tang, L. Yuan *et al.*, “Sharegpt4video: Improving video understanding and generation with better captions,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 19 472–19 495, 2024.
- [41] L. Xu, Y. Zhao, D. Zhou, Z. Lin, S. K. Ng, and J. Feng, “Pllava: Parameter-free llava extension from images to videos for video dense captioning,” *arXiv preprint arXiv:2404.16994*, 2024.
- [42] M. M. Islam, N. Ho, X. Yang, T. Nagarajan, L. Torresani, and G. Bertasius, “Video recap: Recursive captioning of hour-long videos,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 198–18 208.
- [43] D. Ko, J. S. Lee, W. Kang, B. Roh, and H. J. Kim, “Large language models are temporal and causal reasoners for video question answering,” *arXiv preprint arXiv:2310.15747*, 2023.
- [44] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, Y. Liu, Z. Wang, J. Xu, G. Chen, P. Luo *et al.*, “Mvbench: A comprehensive multi-modal video understanding benchmark,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 195–22 206.
- [45] M. Maaz, H. Rasheed, S. Khan, and F. Khan, “Videogpt+: Integrating image and video encoders for enhanced video understanding,” *arXiv preprint arXiv:2406.09418*, 2024.
- [46] Y. Guo, J. Liu, M. Li, D. Cheng, X. Tang, D. Sui, Q. Liu, X. Chen, and K. Zhao, “Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 3, 2025, pp. 3302–3310.
- [47] Y. Xu, Y. Sun, Z. Xie, B. Zhai, and S. Du, “Vtg-gpt: Tuning-free zero-shot video temporal grounding with gpt,” *Applied Sciences*, vol. 14, no. 5, p. 1894, 2024.
- [48] Y. Wang, X. Meng, J. Liang, Y. Wang, Q. Liu, and D. Zhao, “Hawkeye: Training video-text llms for grounding text in videos,” *arXiv preprint arXiv:2403.10228*, 2024.
- [49] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, “The dawn of lmms: Preliminary explorations with gpt-4v (ision),” *arXiv preprint arXiv:2309.17421*, vol. 9, no. 1, p. 1, 2023.
- [50] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser *et al.*, “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context,” *arXiv preprint arXiv:2403.05530*, 2024.
- [51] Y. Zhang, B. Li, h. Liu, Y. j. Lee, L. Gui, D. Fu, J. Feng, Z. Liu, and C. Li, “Llava-next: A strong zero-shot video understanding model,” April 2024. [Online]. Available: <https://llava-vl.github.io/blog/2024-04-30-llava-next-video/>
- [52] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, and L. Yuan, “Video-llava: Learning united visual representation by alignment before projection,” *arXiv preprint arXiv:2311.10122*, 2023.
- [53] J. Lin, H. Yin, W. Ping, P. Molchanov, M. Shoenybi, and S. Han, “Vila: On pre-training for visual language models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 689–26 699.
- [54] R. Zellers, J. Lu, X. Lu, Y. Yu, Y. Zhao, M. Salehi, A. Kusupati, J. Hessel, A. Farhadi, and Y. Choi, “Merlot reserve: Neural script knowledge through vision and language and sound,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 375–16 387.
- [55] J. Gao, Y. Lian, Z. Zhou, Y. Fu, and B. Wang, “Livechat: A large-scale personalized dialogue dataset automatically constructed from live streaming,” *arXiv preprint arXiv:2306.08401*, 2023.
- [56] B. Zhu, B. Lin, M. Ning, Y. Yan, J. Cui, H. Wang, Y. Pang, W. Jiang, J. Zhang, Z. Li *et al.*, “Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment,” *arXiv preprint arXiv:2310.01852*, 2023.
- [57] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, “Flava: A foundational language and vision alignment model,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 15 638–15 650.
- [58] K. Mangalam, R. Akshulakov, J. Malik *et al.*, “Egoschema: A diagnostic benchmark for very long-form video language understanding,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

- [59] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao, "Activitynet-qa: A dataset for understanding complex web videos via question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9127–9134.
- [60] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "Hero: Hierarchical encoder for video+ language omni-representation pre-training," *arXiv preprint arXiv:2005.00200*, 2020.
- [61] V. Patraucean, L. Smaira, A. Gupta, A. Recasens, L. Markeeva, D. Banarse, S. Koppula, M. Malinowski, Y. Yang, C. Doersch *et al.*, "Perception test: A diagnostic benchmark for multimodal video models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [62] A. Zadeh, M. Chan, P. P. Liang, E. Tong, and L.-P. Morency, "Social-iq: A question answering benchmark for artificial social intelligence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8807–8817.
- [63] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang, "Video question answering via gradually refined attention over appearance and motion," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1645–1653.
- [64] J. Lei, L. Yu, M. Bansal, and T. L. Berg, "Tvqa: Localized, compositional video question answering," *arXiv preprint arXiv:1809.01696*, 2018.
- [65] J. Xiao, X. Shang, A. Yao, and T.-S. Chua, "Next-qa: Next phase of question-answering to explaining temporal actions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9777–9786.
- [66] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang *et al.*, "Moviechat: From dense token to sparse memory for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 221–18 232.
- [67] W. Wang, Z. He, W. Hong, Y. Cheng, X. Zhang, J. Qi, S. Huang, B. Xu, Y. Dong, M. Ding *et al.*, "Lvbench: An extreme long video understanding benchmark," *arXiv preprint arXiv:2406.08035*, 2024.
- [68] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim, "Tgif-qa: Toward spatio-temporal reasoning in visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2758–2766.
- [69] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.
- [70] D. Kahneman, "Evaluation by moments: Past and future," *Choices, values, and frames*, pp. 693–708, 2000.
- [71] D. Kahneman, B. L. Fredrickson, C. A. Schreiber, and D. A. Redelmeier, "When more pain is preferred to less: Adding a better end," *Psychological science*, vol. 4, no. 6, pp. 401–405, 1993.
- [72] B. L. Fredrickson and D. Kahneman, "Duration neglect in retrospective evaluations of affective episodes," *Journal of personality and social psychology*, vol. 65, no. 1, p. 45, 1993.
- [73] Z. Cai, M. Cao, H. Chen, K. Chen, K. Chen, X. Chen, X. Chen, Z. Chen, Z. Chen, P. Chu *et al.*, "Internlm2 technical report," *arXiv preprint arXiv:2403.17297*, 2024.
- [74] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 296–26 306.
- [75] M. Han, L. Yang, X. Chang, and H. Wang, "Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos," *arXiv preprint arXiv:2312.10300*, 2023.
- [76] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.
- [77] Y. Zhang, B. Li, h. Liu, Y. j. Lee, L. Gui, D. Fu, J. Feng, Z. Liu, and C. Li, "Llava-next: A strong zero-shot video understanding model," April 2024. [Online]. Available: <https://llava-vl.github.io/blog/2024-04-30-llava-next-video/>
- [78] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly state our contributions and summarize them at the end of Sec. 1, and the claims match the experimental results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations are discussed in Sec. 7.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Detailed instructions for replicating the results are provided in Sec. 5.1 and App. A. Additionally, we provide detailed code, model weights, datasets, and the complete training/inference process in the anonymous GitHub repository <https://anonymous.4open.science/r/LiveStar-5272>.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Detailed instructions for replicating the results are provided in Sec. 5.1 and App. A. Additionally, we provide detailed code, model weights, datasets, and the complete training/inference process in the anonymous GitHub repository <https://anonymous.4open.science/r/LiveStar-5272>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please see Sec. 5.1 and App. A for experimental settings/details, and Sec. 4 for data splits.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to resource constraints, we did not include error bars in our analysis, following the common practice in prior studies within this field to maintain consistency.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report details on computational requirements to run the experiments in App. A.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research adheres to the NeurIPS Code of Ethics. Anonymity is preserved during the current stage.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The potential impacts are discussed in App. G.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: No such models or datasets are involved, so there are no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, we have properly credited the creators or original owners of the assets used in the paper, and we have done so in appropriate ways under the CC-BY 4.0 license.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: All new assets introduced in this paper will be well documented upon their release.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: We hired 20 professional annotators from the outsourcing department of our company. All participants received appropriate and fair compensation according to the standards of their respective regions. The annotation guidelines can be found in App.D.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[No\]](#)

Justification: There were no potential risks incurred by the study participants. The hiring and annotation work were approved by our company's legal department.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: In addition to correcting grammatical errors and editing the format, this paper also used LLMs to score the model output as part of the evaluation. Detailed evaluation criteria can be found in App. E, and the prompt is provided in App. H.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A Implementation Details	22
A.1 Model Architecture	22
A.2 Training Datasets.....	22
B Baselines	23
C Additional Experiments	24
C.1 Dialogue and Streaming Evaluations on SVBench.....	24
D Online Tasks	24
D.1 Real-time Narration Generation	25
D.2 Online Temporal Grounding	25
D.3 Frame-level Dense Question-Answering	25
D.4 Contextual Online Question-Answering	25
D.5 Multi-turn Interactive Question-Answering	26
E Online Evaluation Metrics	29
E.1 Semantic Correctness	29
E.2 Summarize Fluency	30
F Visualization of LiveStar	34
G Impact Statements	34
H Prompt	35

A Implementation Details

A.1 Model Architecture

Inspired by InternVideo2.5-Chat-8B [10], the model consists of a vision encoder (InternViT [1]), an MLP projector, and a large language model (InternLM2.5-7B [73]). For the vision encoder, we employ InternViT, a model pre-trained on a hybrid dataset combining image captioning and OCR-specific data, to extract video frame embeddings at a rate of 1-4 FPS, with each frame represented by 16 tokens. To optimize efficiency in online video understanding, we fine-tuned the model using a static resolution strategy, enabling it to process multi-minute video content within an 8K-token context window. The extracted frame embeddings are then fed into an MLP projector to generate frame tokens, following the approach used in LLaVA-1.5 [74]. These frame tokens are interleaved with language tokens and input into the large language model, InternLM2.5.

A.2 Training Datasets

Our proposed LiveStar builds upon the pre-trained InternVideo2.5 foundation model [10], which demonstrates robust capabilities in offline video comprehension through spatial-temporal reasoning and long-term temporal modeling, achieved through large-scale pretraining on 3.7M video-text pairs. To bridge the gap between static video understanding and dynamic streaming perception, we design a two-phase progressive training paradigm, implementing streaming video-language alignment through Streaming Causal Attention Masks (SCAM) as described in Sec. 3.1, to enable effective adaptation to online video understanding scenarios.

Phase I: Temporal Alignment Pretraining. We conduct dense video-language alignment using 63K carefully curated video segments from carefully selected high-quality *subsets* of three complementary datasets: ActivityNet Captions [59] (9K selected from 20K raw samples), Shot2Story [75] (33K selected from 43K raw samples), Ego4D Narration Stream [28] (20K selected from 113K raw samples), and MVBench [44] (1K selected from 4K raw samples). This stage focuses on establishing frame-level semantic correspondences and enhancing the model’s capability to process streaming visual input through SCAM.

Phase II: Multi-Task Online Adaptation. We employ the OmniStar benchmark’s 20K training samples across five distinct online video understanding tasks: (1) Real-time narration generation, (2) Online temporal grounding, (3) Frame-level dense QA, (4) Contextual online QA, and (5) Multi-turn interactive QA. Through task-specific adapters, we achieve simultaneous multi-objective alignment that preserves general video understanding capabilities while specializing for streaming scenarios.

This hierarchical training strategy demonstrates two key advantages: 1) Effective knowledge transfer from static to streaming domains through progressive alignment, and 2) Efficient utilization of limited annotated data (83K total samples) via our SCAM-enhanced streaming video-language alignment framework.

B Baselines

In this paper, we present a comparative analysis of online and offline Video-LLMs/LVLMs, evaluating their capabilities in two critical aspects of video understanding: (1) determining optimal response timing and (2) processing continuous video streams. Our study focuses on benchmarking their performance in dynamic, real-world scenarios. We categorize baseline models into two groups based on their temporal processing paradigm. **Online models** are evaluated for their ability to incrementally analyze streaming video frames with minimal timing difference, with the key feature being their ability to autonomously decide when to output. In contrast, **offline models** process complete video sequences as static inputs or can handle streaming video but cannot autonomously decide when to output. This dichotomy allows us to systematically compare temporal reasoning strategies and online video understanding capabilities across different architectures.

Online Video-LLMs:

- VideoLLM-online [20]: a streaming-oriented architecture employs continuous key-value cache to enable low-latency incremental processing of video streams. It designs a streaming EOS (End-Of-Sequence) prediction mechanism to determine response timing based on user queries and generates EOS tokens to mark silence intervals.
- VideoLLM-MoD (reproduced) [21]: a modified variant of VideoLLM-online utilizing mixture-of-depths token selection to balance computational efficiency and temporal reasoning accuracy in streaming video analysis. Since the code is not open-sourced, we reproduced it based on VideoLLM-online.
- MMDuet [24]: a multimodal dialogue-oriented model featuring dual-stream encoders for synchronized audio-visual processing is specifically designed for interactive video applications. Unlike VideoLLM-online and VideoLLM-MoD, this model uses a classification head to decide whether to proceed with decoding and output.
- LION-FS (paper-reported results) [22]: a dual-path video-language framework that resolves the efficacy-efficiency trade-off in online video assistance through dynamic routing-based real-time response determination and multi-granular keyframe augmentation with spatial-interaction modeling. Since it is not open-sourced, we used the reported results from the original paper for comparison on the Ego4D Narration Stream benchmark.

These models use their respective response-silence decoding strategies to effectively process continuous video streams and generate timely responses. They possess temporal decision-making capabilities, responding at contextually appropriate moments, processing frame-by-frame inputs, and determining response timing.

Offline Video-LLMs/LVLMs:

- GPT-4V & GPT-4o [76]: both models process real-time video streams alongside synchronized audio, text (e.g., subtitles), and visual frames, enabling holistic comprehension of dynamic scenes. Their autoregressive architecture supports sequential processing of video frames, capturing temporal dependencies critical for tasks like action recognition, event prediction, or summarizing long-form content.
- LLaVA-NeXT-Video [77]: an open-source chatbot advances video understanding through zero-shot AnyRes representation and linear scaling for long-context processing.

- InternVideo2.5 [10] & InternVL2.5 [1]: the first open-source MLLMs to achieve over 70% on the MMMU benchmark, matching the performance of leading closed-source commercial models like GPT-4o.
- MiniCPM-V 2.6 [3]: the latest and most capable model in the MiniCPM-V series.
- Qwen2.5-VL [78]: the flagship multimodal model in the Qwen series, advances visual-language intelligence through a native dynamic-resolution ViT with Window Attention, enabling efficient high-fidelity processing of variable-sized images and hour-long videos with second-level temporal localization.

Implementation Notes: Due to the lack of open-source code for VideoLLM-MoD, we reproduced it based on the VideoLLM-online framework. For LION-FS, which is also not open-sourced, we utilized the results reported in the original paper for comparison on the Ego4D Narration Stream benchmark. StreamMind was excluded from our comparisons due to the absence of open-source datasets and model weights.

C Additional Experiments

C.1 Dialogue and Streaming Evaluations on SVBench

We benchmarked open-source Video-LLMs/LVLMs and closed-source LVLMs aligned with OmniStar on SVBench. The dialogue and streaming evaluation results on SVBench, outlined in Table 6, provide a comprehensive comparison among various Video-LLMs. We have published the evaluation results on the official leaderboard website². Notably, closed-source models such as GPT-4o and GPT-4V attain significantly higher scores across all metrics, with GPT-4o achieving an Overall Score (OS) of 62.57 in dialogue evaluation and 59.97 in streaming evaluation. LiveStar’s zero-shot performance on SVBench in dialogue and streaming evaluations outperforms LLaVA-NeXT-Video, InternVL2.5, and InternVideo2.5, but underperforms MiniCPM-V 2.6 and Qwen2.5-VL. However, after fine-tuning on the SVBench training set, LiveStar achieves an average improvement of 15.37% across metrics, surpassing all open-source Video-LLMs/LVLMs and closing the gap with GPT-4V. This demonstrates LiveStar’s robust online video understanding capabilities.

Table 6: Evaluation results of various models on SVBench in dialogue and streaming evaluation. † denotes fine-tuning on the SVBench training set.

Model	Dialogue Evaluation						Streaming Evaluation						AVG
	SA	CC	LC	TU	IC	OS	SA	CC	LC	TU	IC	OS	
Closed-source LVLMs													
GPT-4V	56.03	62.61	69.09	65.36	53.73	60.30	56.37	61.41	65.80	59.18	57.16	57.93	59.12
GPT-4o	58.26	64.76	70.75	67.68	55.82	62.57	57.99	63.52	67.72	60.18	59.25	59.97	61.27
Open-source Video-LLMs/LVLMs													
LLaVA-NeXT-Video	37.71	44.59	52.05	41.80	36.58	41.40	34.29	39.68	47.65	35.33	36.68	36.12	38.76
InternVL2.5	43.73	50.70	56.61	55.03	43.46	48.73	40.44	48.34	52.84	46.93	45.10	45.04	46.89
InternVideo2.5	46.83	53.48	58.22	58.91	47.02	51.73	41.76	49.72	53.25	48.44	47.10	46.58	49.16
MiniCPM-V 2.6	51.70	59.50	65.33	61.72	50.09	56.63	46.44	52.73	58.35	53.48	48.32	49.67	53.15
Qwen2.5-VL	52.54	59.85	65.52	64.64	51.23	57.57	48.21	56.12	60.31	56.33	52.46	52.84	55.21
Online Assistants													
LiveStar	46.43	53.75	59.36	57.29	45.64	51.37	43.56	51.52	55.71	50.79	47.77	48.15	49.76
LiveStar [†]	54.06	61.08	66.43	66.06	52.67	58.95	52.19	59.00	62.85	58.35	54.95	55.87	57.41

D Online Tasks

Our dataset is organized into 5 distinct online evaluation tasks designed to comprehensively assess Video-LLMs’ capabilities in streaming video understanding. Each task targets a specific aspect of temporal reasoning and contextual awareness under causal constraints. Detailed case examples for each task are provided in Tabs. 7, 8, and 9.

²<https://huggingface.co/spaces/yzy666/SVBench>

D.1 Real-time Narration Generation

This task requires Video-LLMs to generate accurate and temporally coherent descriptions for streaming video in real time, where descriptive segments are selectively triggered at transitions between semantic clips to avoid redundancy. Rather than producing isolated captions, the output is expected to form a coherent and continuous narrative that evolves with the visual content. Each segment should be logically connected to the previous ones through transitional phrases or discourse markers (e.g., *first, then, meanwhile, as a result, finally*) to ensure a smooth and natural flow. Coreference resolution is also required: entities introduced earlier (e.g., *"a man"*) may be subsequently referred to using appropriate pronouns (e.g., *"he"*), demonstrating contextual awareness. Although generated in multiple parts, the narration should ultimately come together as a single, fluid, and unified story—one that maintains semantic clarity, narrative cohesion, and temporal alignment with the unfolding video.

D.2 Online Temporal Grounding

This task requires Video-LLMs to detect and localize temporal segments in a streaming video that semantically correspond to a given textual query, enabling real-time moment retrieval without access to future frames. For each query, the task involves identifying the start point where the visual content begins to align with the described event and the end point where the scene transitions to a different semantic clip. Importantly, multiple distinct segments in a video may independently satisfy the same query—for example, the phrase *"Video of a car in travel."* might correspond to several non-contiguous intervals throughout a long video, each relevant interval should be detected independently and promptly as it unfolds. It assesses a model’s capacity for fine-grained, online temporal localization and its ability to detect multiple instances of the same semantic clip over time.

Since current Video-LLMs lack native online temporal grounding capability, we propose a surrogate evaluation strategy. We first divide the dataset based on the length of the relevant window (e.g., 10s, 20s, 30s), forming candidate temporal intervals of varying durations. For each video and query pair, we compute the model’s per-frame perplexity (*PPL*) with respect to the query, treating lower perplexity as an indicator of stronger semantic alignment. Among the candidate intervals of the specified length, we select the one with the lowest average perplexity as the predicted segment. Finally, we compare the predicted segment with the ground-truth relevant window to compute a temporal deviation metric (*TimDiff*) that quantifies alignment accuracy.

D.3 Frame-level Dense Question-Answering

This task requires Video-LLMs to generate multiple temporally grounded answers to a single persistent question throughout a video, producing dense yet non-redundant responses that are triggered only by meaningful semantic changes in the visual content. For instance, given the question *"How many people are there in the video now?"*, the model should initially respond with *"One person"* and remain silent until a second individual appears, at which point the response should be updated to *"Two people"*. The task emphasizes the production of sparse but informative answer sequences that reflect event-driven and temporally consistent reasoning. It ensures that an online video model’s outputs align closely with significant shifts in the video’s semantic clips, prioritizing relevance and conciseness over frequency.

D.4 Contextual Online Question-Answering

This task requires Video-LLMs to perform contextual question answering on a continuously streaming video by addressing queries issued at arbitrary time points, based on the visual content observed up to that moment. These questions are organized into QA chains—temporally distributed, multi-turn interactions grounded in the same video segment, which may exhibit implicit dependencies between successive questions and answers. Each question must be answered using only the video frames that have been observed at the time of the query, without access to future information. As the video progresses, new questions may arise, which may exhibit contextual dependencies on both historical video segments and dialogues. For example, given the QA chain *Q1: "Who is the competitor in this event?"*, a follow-up question *Q2: "What did he continue to do?"* relies on coreference resolution, where *"he"* refers to the previously identified *"competitor"*. The task emphasizes incremental understanding, memory retention, and context-aware reasoning under causal constraints. It evaluates

the ability to utilize historical information and engage in coherent multi-turn dialogue within a real-time video stream.

D.5 Multi-turn Interactive Question-Answering

This task requires Video-LLMs to perform multi-turn question answering on a continuously streaming video by responding to multiple queries—potentially issued simultaneously at the same time point—based on the visual content observed up to that moment. Similar to the **Contextual Online Question-Answering Task D.4**, all queries are organized into QA chains, with each chain representing a temporally distributed sequence of questions and answers pertaining to the same semantic segment of the video. In this interactive setting, multiple QA chains may evolve in parallel, and the questions within or across chains may exhibit implicit contextual dependencies, requiring the model to reason jointly over the video history and multiple dialogue threads. The task emphasizes the ability to handle concurrent multi-turn interactions in a streaming context, assessing whether the model can track and resolve complex inter-question relationships while maintaining temporal grounding and causal consistency.

Table 7: Case examples of OmniStar (RNG, OTG, COQ, MIQ).

Online Task	Case Examples
Real-time Narration Generation	<i><RNG task prompt> Please describe the content of the video in real time and use cohesive words to keep the narrative coherent.</i> 00:00: In a room, there is a woman with short purple hair wearing a black top, and there are cartoon characters above her head. 00:01 ~ 00:03: Silent. 00:04: Then the cartoon characters above her head keep changing, and finally stay on a cartoon character of a little red man. 00:05 ~ 00:10: Silent. 00:11: Afterwards, the woman plays the cartoon character. She wears a black top hat decorated with a row of white teeth and a red suit against a red background. 00:12 ~ 00:20: Silent. 00:21: Finally, the woman holds a white balloon with a black mouth and eyes drawn on it. She keeps staring at the balloon, and the picture zooms in on her eyes.
Online Temporal Grounding	<i><OTG task prompt> A man reporting from outside of a destroyed house.</i> 00:00 ~ 01:13: Silent. (Not related) 01:14: Start. 01:15 ~ 01:23: Silent. (Related) 01:24: End. 01:25 ~ 01:29: Silent. (Not related) 01:30: Start. 01:31 ~ 01:39: Silent. (Related) 01:40: End. 01:41 ~ 02:00: Silent. (Not related)
Contextual Online QA	<i><COQ task prompt></i> 00:00 ~ 00:16: Silent. 00:17: <i>Q: What were the two ladies talking about at the beginning of the video clip?</i> A: They were discussing entering the house. 00:18 ~ 00:22: Silent. 00:23: <i>Q: What are they discussing?</i> A: They are discussing how to express gratitude for each other’s help. 00:24 ~ 00:33: Silent. 00:34: <i>Q: Why are they talking about the topic of gratitude?</i> A: They are talking about the topic of gratitude because one of them helped the other, prompting this discussion.
Multi-turn Interactive QA	<i><MIQ task prompt></i> 00:00 ~ 00:06: Silent. 00:07: <i>Q1: What appears at the beginning of the video?</i> A1: A row of bright windows appears. <i>Q2: What is above the window?</i> A2: It’s a white roof. <i>Q3: What is in front of it?</i> A3: There are several white tables. 00:08 ~ 00:14: Silent. 00:15: <i>Q1: Where is this row of windows located?</i> A1: Located within an indoor space. <i>Q2: What is inside this indoor space?</i> A2: There is a lady and some equipment. <i>Q3: What does the equipment include?</i> A3: Includes yoga mats and fitness balls. 00:16 ~ 00:27: Silent. 00:28: <i>Q1: What is there at the beginning of the video?</i> A1: There is a room for fitness and exercise. <i>Q2: What are the equipments in the indoor space?</i> A2: There are yoga mats, fitness balls, dumbbells, and other exercise equipment. <i>Q3: What other furniture and decorations are in the room?</i> A3: In the room, there are also mirrors, potted plants, and a basket among other decorations.

Table 8: Case examples of OmniStar (FDQ-1).

Online Task	Case Examples
Frame-level Dense QA	<i><FDQ task prompt> What is the person showing to the camera?</i> 00:00: The person shows nothing to the camera. 00:01 ~ 00:05: Silent. 00:04: The person is showing a white cup to the camera. 00:07 ~ 00:10: Silent. 00:11: The person is showing a grey cup to the camera. 00:12 ~ 00:16: Silent. 00:17: The person is showing a red book to the camera. 00:18 ~ 00:23: Silent.
	<i><FDQ task prompt> How many times did the person launch the toy car on the slanted plane?</i> 00:00: The person did not launch the toy car on the slanted plane. 00:01 ~ 00:04: Silent. 00:05: The person launched the toy car once on the slanted plane. 00:06 ~ 00:12: Silent. 00:13: The person launched the toy car twice on the slanted plane. 00:14 ~ 00:19: Silent. 00:20: The person launched the toy car three times on the slanted plane. 00:21 ~ 00:30: Silent.
	<i><FDQ task prompt> What repeated actions did the person do?</i> 00:00: The person did not do repeated actions. 00:01 ~ 00:16: Silent. 00:17: The person clapped hands. 00:18 ~ 00:21: Silent. 00:22: The person tapped the table with a ruler. 00:23 ~ 00:27: Silent. 00:28: The person folded paper. 00:29 ~ 00:34: Silent. 00:35: The person did not do repeated actions. 00:36 ~ 00:45: Silent.
	<i><FDQ task prompt> What letters is the person writing on the paper?</i> 00:00: Not writing. 00:01 ~ 00:07: Silent. 00:08: The person is writing the letter 'B'. 00:09 ~ 00:14: Silent. 00:15: The person is writing the letter 'E'. 00:16 ~ 00:18: Silent. 00:19: The person is writing the letter 'D'. 00:20 ~ 00:23: Silent. 00:24: Not writing. 00:25 ~ 00:41: Silent.
	<i><FDQ task prompt> What is the order of the letters on the table?</i> 00:00: The order of the letters on the table is 'LOVE'. 00:01 ~ 00:18: Silent. 00:19: The letters on the table is being shuffled. 00:20 ~ 00:42: Silent. 00:43: The order of the letters on the table is 'VOLE'. 00:44 ~ 01:01: Silent.

Table 9: Case examples of OmniStar (FDQ-2).

Online Task	Case Examples
Frame-level Dense QA	<i><FDQ task prompt> How many times did the objects in the video collide?</i> 00:00: No collision occurred. 00:01 ~ 00:04: Silent. 00:05: One collision occurred. 00:06 ~ 00:10: Silent. 00:11: Two collisions occurred. 00:12 ~ 00:19: Silent.
	<i><FDQ task prompt> How many objects are there in the video?</i> 00:00: There are four objects in the video. 00:01 ~ 00:02: Silent. 00:03: There are five objects in the video. 00:03 ~ 00:07: Silent. 00:08: There are six objects in the video. 00:09 ~ 00:19: Silent.
	<i><FDQ task prompt> How many moving objects are in the video?</i> 00:00: No objects are moving in the video. 00:01 ~ 00:03: Silent. 00:04: Two objects are moving in the video. 00:05 ~ 00:08: Silent. 00:09: Three objects are moving in the video. 00:10 ~ 00:19: Silent.
	<i><FDQ task prompt> What is the scene in the video like?</i> 00:00: The scene in the video shows several people in the interrogation room of the police station. 00:01 ~ 00:15: Silent. 00:16: The scene in the video changes to five women outside in a garden. 00:17 ~ 00:30: Silent.
	<i><FDQ task prompt> How many people are wearing glasses in the video now?</i> 00:00: There is a person wearing glasses in the video. 00:01 ~ 00:08: Silent. 00:09: There are no people wearing glasses in the video. 00:10 ~ 00:15: Silent. 00:16: There are two people wearing glasses in the video. 00:17 ~ 00:21: Silent.
	<i><FDQ task prompt> What actions is the person performing in the video during the cooking preparation?</i> 00:00 ~ 00:03: Silent. 00:04: The person takes out a cucumber wrapped in plastic. 00:05 ~ 00:08: Silent. 00:09: The person takes out a knife. 00:10 ~ 00:15: Silent. 00:16: The person takes out a cutting board. 00:17 ~ 00:22: Silent. 00:23: The person takes out a plate.

E Online Evaluation Metrics

We utilize GPT-4o to evaluate the generated answers in terms of semantic correctness and summary fluency.

E.1 Semantic Correctness

The semantic correctness evaluation is used to assess the ability of Video-LLMs to understand video segments after segmenting the complete video, focusing on whether the model can accurately capture and comprehend all the information in the video segments. When evaluating the semantic correctness of the answers, we input both the model-generated answers for the corresponding video segments

and the ground truth into GPT-4o, and ask GPT-4o to evaluate the model-generated answers from metrics of Semantic Accuracy, Language Quality and Information Completeness, then output scores for these three metrics.

- **Semantic Accuracy.** It evaluates the alignment of core meaning between generated answers and ground truth, evaluating semantic compatibility without errors or hallucinations. The evaluation metric focuses on assessing the overall correctness of the generated answers. The more accurate and closer the generated answer is to the corresponding parts of the ground truth, the higher the score, and vice versa. For this metric, we design five scoring criteria: a score of 10 indicates perfect matching with ground truth and no deviation; a score of 7-9 indicates minor semantic differences but core information intact; a score of 4-6 indicates significant errors or omissions in key details; a score of 1-3 indicates mostly incorrect or irrelevant; a score of 0 indicates completely wrong or unrelated.
- **Language Quality.** It evaluates grammatical correctness, lexical precision, and readability, ensuring clarity and fluency in expression. If the generated answer has less grammatical issues or spelling errors, the score will be higher; otherwise, the score will be lower. For this metric, we design five scoring criteria: a score of 10 indicates flawless grammar, precise wording and no redundancy; a score of 7-9 indicates minor grammatical slips but do not affect understanding; a score of 4-6 indicates obvious grammar issues or unclear phrasing; a score of 1-3 indicates chaotic language and multiple errors hinder comprehension; a score of 0 indicates unintelligible or symbol-filled.
- **Information Completeness.** It evaluates the breadth and depth of information coverage relative to ground truth, including key events and dynamic actions without omissions. This evaluation metric focuses on assessing whether the generated answer is sufficiently comprehensive compared to the ground truth. The more ground truth information the generated answer covers, the higher the score, and vice versa. For this metric, we design five scoring criteria: a score of 10 indicates including all key information from ground truth; a score of 7-9 indicates main points covered but missing minor details; a score of 4-6 indicates partial key information missing but retaining the core content; a score of 1-3 indicates only fragments of information retained; a score of 0 indicates no valid information provided.

E.2 Summarize Fluency

The summary fluency evaluation is used to assess whether the model, after segmenting the complete video, can notice the information contained in previous video segments and thus incorporate this information into its answers. When evaluating the summary fluency of the answers, we concatenate the model-generated responses for all video segments and the corresponding ground truth for different segments, then input them into GPT-4o, and request GPT-4o to evaluate the model-generated answers from metrics of Writing Logicality, Language Fluency, Writing Conciseness, Semantic Consistency and Narrative Completeness, then output scores for these five metrics.

- **Writing Logicality.** It evaluates the clarity and strength of causal, temporal, or inferential relationships between sentences, ensuring coherent progression of ideas. The clearer and more coherent the cause-and-effect logic in the generated answer, the higher the score it receives. Conversely, if the logic is confused, the score will be lower. For this metric, we design five scoring criteria: a score of 10 indicates flawless logic with no gaps or contradictions; a score of 7-9 indicates minor logical jumps but generally coherent; a score of 4-6 indicates weak connections requiring reader inference; a score of 1-3 indicates frequent illogical breaks or contradictions; a score of 0 indicates no logical structure.
- **Language Fluency.** It evaluates precision in wording and avoidance of redundant content. If the generated answer contains sentences with abrupt transitions, the score will be low. If the transitions are smooth, the score will be high. For this metric, we design five scoring criteria: a score of 10 indicates effortless flow with no awkward phrasing; a score of 7-9 indicates smooth overall with rare hiccups; a score of 4-6 indicates choppy transitions and some re-reading needed; a score of 1-3 indicates relies heavily on forced connectors; a score of 0 indicates unreadable due to disjointedness.
- **Writing Conciseness.** It evaluates precision in wording and avoidance of redundant content. If the generated answer contains redundant or repetitive expressions, the score will be low.

If the writing is concise and succinct, the score will be high. For this metric, we design five scoring criteria: a score of 10 indicates precise wording and no redundancy; a score of 7-9 indicates minor repetitions but efficient overall; a score of 4-6 indicates verbose sections needing trimming; a score of 1-3 indicates excessive repetition or filler; a score of 0 indicates entirely redundant or irrelevant.

- **Semantic Consistency.** It evaluates uniformity in terminology, pronouns, and tense usage across the text. If the generated answer contains inconsistencies in terminology or pronoun usage, the score will be low. If the pronouns, tenses, and terminology are consistent throughout the answer, the score will be high. For this metric, we design five scoring criteria: a score of 10 indicates perfect consistency in all elements; a score of 7-9 indicates slight inconsistencies with no impact; a score of 4-6 indicates multiple inconsistencies requiring correction; a score of 1-3 indicates confusing shifts in terms or perspective; a score of 0 indicates incoherent due to inconsistency.
- **Narrative Completeness.** It evaluates coverage of all key information without omissions. This evaluation metric focuses on assessing whether there is any missing information in the overall narrative of the generated answer. The more information is missing, the lower the score; conversely, the less information is missing, the higher the score. For this metric, we design five scoring criteria: a score of 10 indicates including all critical details; a score of 7-9 indicates covering main points but having minor gaps possible; a score of 4-6 indicates missing some key elements; a score of 1-3 indicates only fragmented or irrelevant information; a score of 0 indicates no relevant information.



Figure 5: Comparison of VideoLLM-online, MMDuet, and LiveStar on the RNG task.

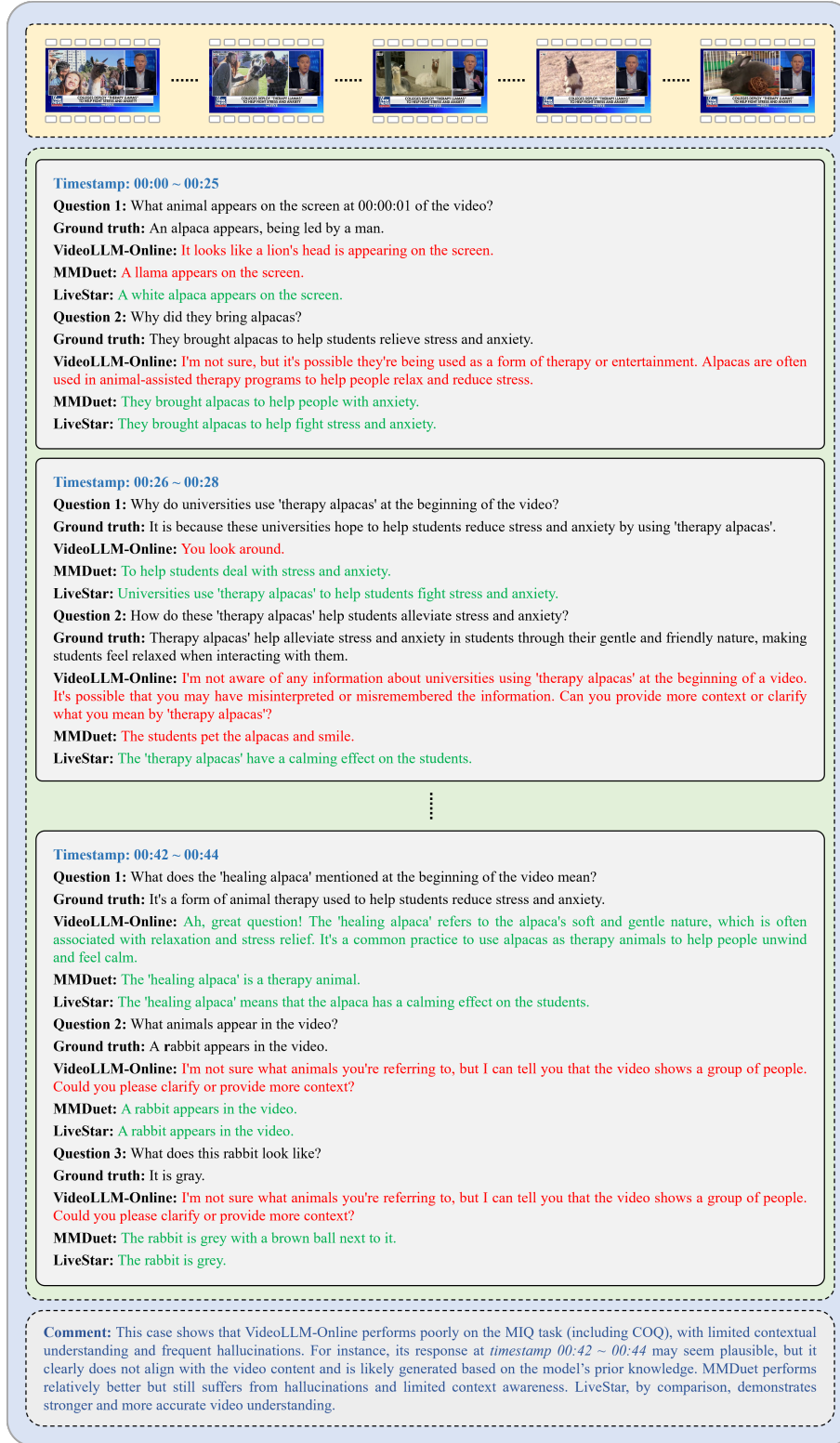


Figure 6: Comparison of VideoLLM-online, MMDuet, and LiveStar on the MIQ task (also representative of the COQ task).

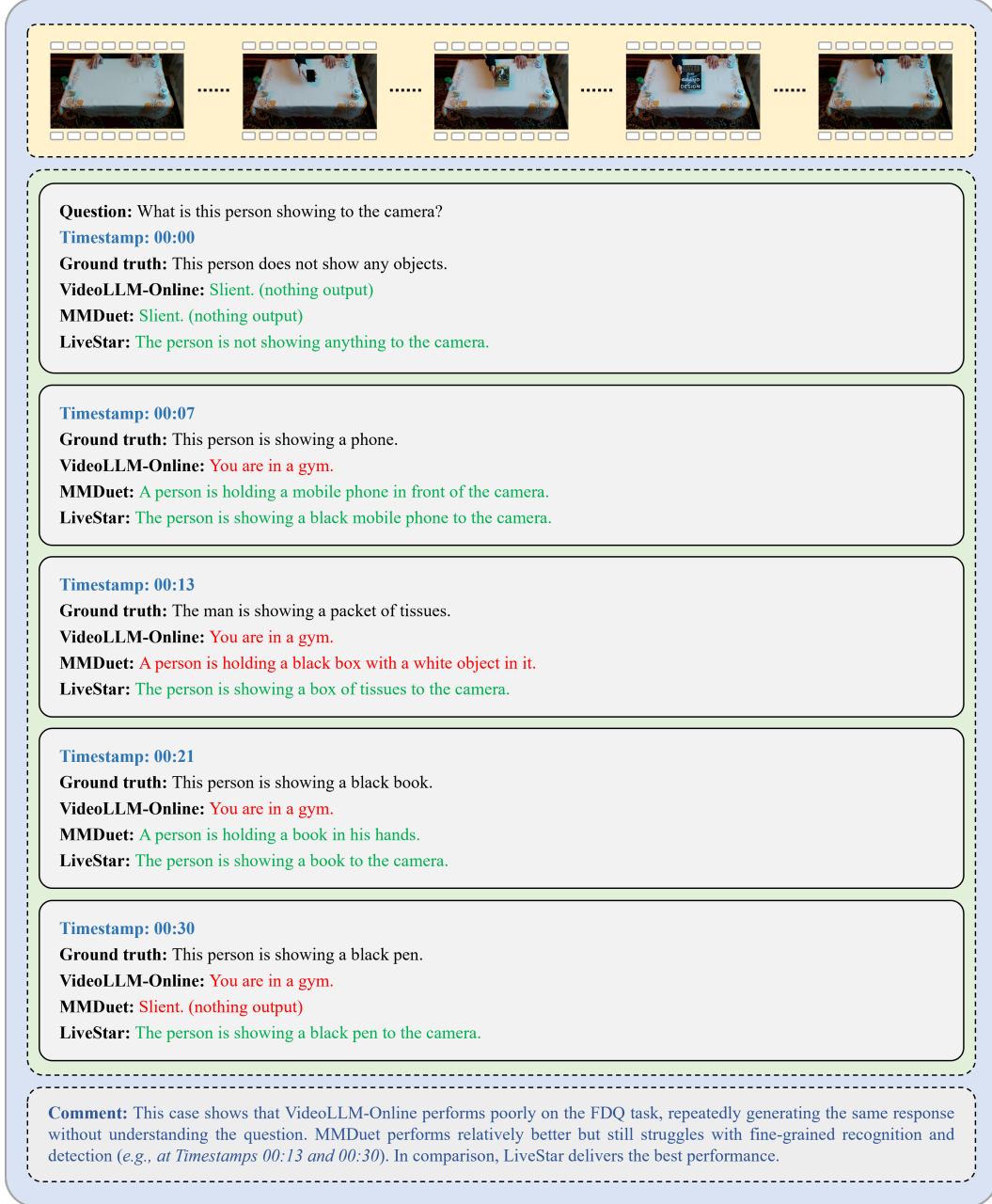


Figure 7: Comparison of VideoLLM-online, MMDuet, and LiveStar on the FDQ task.

F Visualization of LiveStar

We develop a web-based demo for LiveStar using Gradio, which enables users to upload arbitrary videos and perform real-time streaming chat. The interface also allows flexible adjustment of inference parameters (e.g., the tunable scaling factor α) to accommodate various application scenarios. The demo interface is illustrated in Figure 8.

G Impact Statements

Our proposed LiveStar, a pioneering live streaming assistant with temporal decision-making capabilities, presents significant societal implications. On the positive side, LiveStar introduces a paradigm

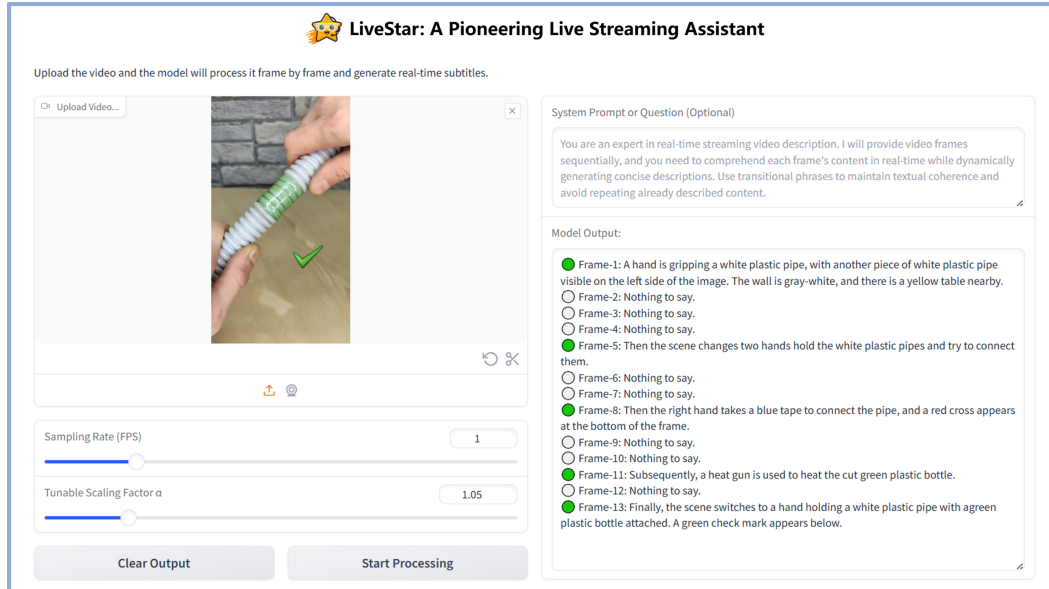


Figure 8: A web demo of LiveStar. Green points indicate generated outputs, while white denotes no output.

shift in online understanding by dynamically determining when and how to generate outputs. This enables efficient processing of real-time online data with reduced latency and computational overhead. This innovation holds promise for applications such as real-time online platforms, where it could enhance viewer engagement through instant content summarization, automated moderation, or adaptive subtitling. Additionally, its ability to prioritize critical events in surveillance systems could improve public safety by enabling faster emergency responses in scenarios like traffic monitoring or crowd management. The technology also has potential in assistive tools for individuals with disabilities, such as real-time visual interpretation or context-aware navigation aids.

However, the technology’s adaptive decision logic also introduces risks. When integrated into drones or public cameras, it raises concerns about mass privacy infringement, particularly if deployed without transparent governance. In conclusion, while LiveStar advances real-time online video understanding, its societal adoption demands robust governance frameworks. We advocate for strict regulatory compliance in privacy-sensitive domains and interdisciplinary collaboration to mitigate misuse risks while maximizing its benefits for human-centric applications.

H Prompt

Prompt to Evaluate Semantic Correctness Using GPT-4o Score

Task Description:

You are an expert in evaluating captions generated for various scenes in videos. I will provide you with the ground truth for each scene caption and the caption generated by the model. You need to score the given caption generated by the model based on following three dimensions and output the respective scores for each dimension:

1. Semantic Accuracy: Measures the match in core meaning between the generated caption and Ground Truth.

Scoring criteria: - 10: Perfect matching with Ground Truth, no deviation. - 7-9: Minor semantic differences but core information intact. - 4-6: Significant errors or omissions in key details. - 1-3: Mostly incorrect or irrelevant. - 0: Completely wrong or unrelated.

2. Language Quality: Evaluates grammatical correctness, word choice, and readability.

Scoring criteria: - 10: Flawless grammar, precise wording and no redundancy. - 7-9: Minor grammatical slips, but do not affect understanding. - 4-6: Obvious grammar issues or unclear phrasing. - 1-3: Chaotic language, and multiple errors hinder comprehension. - 0: Unintelligible or symbol-filled.

3. Information Completeness: Assesses breadth and depth of information coverage relative to Ground Truth.

Scoring criteria: - 10: Including all key information from Ground Truth. - 7-9: Main points covered, but missing minor details. - 4-6: Partial key information missing, but retaining the core content. - 1-3: Only fragments of information retained. - 0: No valid information provided.

Requirements:

1. I will give you the ground truth with "Ground Truth:" and the model-generated caption with "Model Response:".

2. When outputting the score for the Semantic Accuracy dimension, it must start with "Semantic Accuracy:". When outputting the score for the Language Quality dimension, it must start with "Language Quality:". When outputting the score for the Information Completeness dimension, it must start with "Information Completeness:". Apart from the scores for the three dimensions, no other characters are allowed in the output.

Ground Truth: {ground_truth}

Model Response: {model_response}

Prompt to Evaluate Semantic Correctness Using GPT-4o Score

Task Description:

You are an expert in evaluating video content caption generation. I will provide you with the ground truth of video caption and the caption generated by the model. You need to score the given caption generated by the model based on following five dimensions and output the respective scores for each dimension:

1. Writing Logicality: Evaluates causal, temporal, or inferential relationships between sentences.

Scoring criteria: 10: Flawless logic with no gaps or contradictions. 7-9: Minor logical jumps but generally coherent. 4-6: Weak connections requiring reader inference. 1-3: Frequent illogical breaks or contradictions. 0: No logical structure.

2. Language Fluency: Assesses natural transitions between sentences.

Scoring criteria: 10: Effortless flow with no awkward phrasing. 7-9: Smooth overall with rare hiccups. 4-6: Choppy transitions, some re-reading needed. 1-3: Relies heavily on forced connectors. 0: Unreadable due to disjointedness.

3. Writing Conciseness: Measures brevity and avoidance of redundancy.

Scoring criteria: 10: No redundancy; precise wording. 7-9: Minor repetitions but efficient overall. 4-6: Verbose sections needing trimming. 1-3: Excessive repetition or filler. 0: Entirely redundant or irrelevant.

4. Semantic Consistency: Checks uniformity in terminology, pronouns, and tense.

Scoring criteria: 10: Perfect consistency in all elements. 7-9: Slight inconsistencies with no impact. 4-6: Multiple inconsistencies requiring correction. 1-3: Confusing shifts in terms or perspective. 0: Incoherent due to inconsistency.

5. Narrative Completeness: Evaluates whether the caption covers all key information.

Scoring criteria: 10: Including all critical details. 7-9: Covering main points, but having minor gaps possible. 4-6: Missing some key elements. 1-3: Fragmented or irrelevant info only. 0: No relevant information.

Requirements:

1. I will give you the ground truth with "Ground Truth:" and the model-generated caption with "Model Response:".

2. When outputting the score for the Writing Logicality dimension, it must start with "Writing Logicality:". When outputting the score for the Language Fluency dimension, it must start with "Language Fluency:". When outputting the score for the Writing Conciseness dimension, it must start with "Writing Conciseness:". When outputting the score for the Semantic Consistency dimension, it must start with "Semantic Consistency:". When outputting the score for the Narrative Completeness dimension, it must start with "Narrative Completeness:". Apart from the scores for the five dimensions, no other characters are allowed in the output.

Ground Truth: {ground_truth}

Model Response: {model_response}

Prompt to Generate Second-level Caption

You are an expert in generating descriptions of streaming videos. Now, given all the frames of a 1-second video, analyze the visual content and generate complete subtitles.

Input: real-time frames of a video clip or video stream.

Steps: 1. Analyze the video content: identify the main events, characters, scenes, and actions in the video. 2. Extract key information: extract key information from the video that can represent the video content. 3. Generate description: generate a concise and rich text description based on the extracted information. 4. Proofreading and optimization: check the accuracy and fluency of the description to ensure that it conforms to the video content.

Notes: 1. Make sure the description is directly related to the video content. 2. The description is prohibited from using modifiers such as adjectives and adverbs. 3. Avoid verbosity and keep it concise. 4. Pay attention to the logic and coherence of the description. 5. It is forbidden to mention content that is not mentioned in the video.

Output example: "A basketball player (wearing a red jersey) jumps vertically in the middle of the court under the bright court lights, with his arms reaching towards the net. The ball spins clockwise in the air, while two defensive players (in blue uniforms) prepare to grab the rebound."

Prompt to Generate Real-Time Narration

Task: Generate fluid, cinematically-coherent real-time narration that creates continuous visual storytelling. Integrate persistent historical elements with instantaneous observations using temporal reasoning and causal chaining.

Inputs: 1. Historical Context (H): {History}

2. Current Second Data (C): {Current_captions}

3. Current Video Frames

Processing Requirements:

1. Dynamic Temporal Modeling: - Maintain object/interaction state vectors across sliding windows - Calculate kinematic signatures (jerk vectors, path integrals) - Anticipate action sequences using momentum calculus - Differentiate transient vs sustained states ("progressively veering" vs "abrupt swerve")

2. Scene State Engine: - Preserve spatial memory lattice for persistent entities - Refresh object kinematics (pose derivatives, interaction potential) - Implement occlusion resilience through Markovian tracking - Maintain projective geometry constraints for spatial coherence

3. Narrative Flow Synthesis: - Forge event domino sequences with causal integrity - Employ aspectual verb framing (sustaining/commencing/culminating) - Apply filmic continuity principles (eyeline matches, temporal ellipses) - Optimize attention allocation through depth-ordered significance

Output Format: "[XX:XX-XX:XX] [Subject][action] [preposition][environment] while [parallel process], with [situational awareness] (phase indicator)"

Example: Input H: "[00:00-00:05] Red sedan accelerates past stationary truck, establishing eastward movement" Input C: "[00:05-00:06] Vehicle decelerating near crosswalk. [00:06-00:07] Pedestrian initiating crossing sequence"

Output: "[00:05-00:07] The crimson sedan progressively moderates speed approaching a crosswalk, its path converging with a pedestrian now committing to traverse, while the historically logged truck persists in the rear-left zone"

Prompt to Generate Real-Time Narration

Task: As a real-time narration generator, create vivid and coherent video commentary by combining:

Historical context from previous narration: {Insert past narration}

Current video clip

Per-second captions: {List timestamped captions}

Guidelines:

- Temporal continuity: Maintain strict time synchronization between captions (seconds 00:00-00:XX) and narration
- Context weaving: Blend historical context with new developments while avoiding repetition
- Fluency: Use natural transitions between sentences and temporal markers (e.g., “Meanwhile,” “Suddenly”)
- Focus: Highlight key actions/objects from captions while maintaining narrative flow

Output Requirements:

- 1-2 sentences per major event/transition
- Active voice and descriptive verbs
- Neutral observatory tone without speculation

Example Output Format: “[00:05-00:07] The crimson sedan progressively moderates speed approaching a crosswalk, its path converging with a pedestrian now committing to traverse, while the historically logged truck persists in the rear-left zone”