Modularized Dynamic-Granularity Video LLM for Multi-Event Long Video Understanding

Wei Feng[1], Xin Wang[1,2,*], Yu-Wei Zhan[1], Yuwei Zhou[1], Wenwu Zhu[1,2,*]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Beijing National Research Center for Information Science and Technology, Tsinghua University
{fw22,zhou-yw21}@mails.tsinghua.edu.cn,zhanyuweilif@gmail.com,{xin_wang,wwzhu}@tsinghua.edu.cn

arXiv:2607.15778v1 [cs.CV] 17 Jul 2026

***Abstract*—Video Large Language Models (Video LLMs) have made significant advancements in various video understanding tasks. However, long-video scenarios remain challenging due to the inherent tension between limited visual token budgets and the need to capture multiple key events. Existing approaches typically process long videos in two stages, i.e., i) select keyframes and ii) perform detailed perception, which exhibit significant limitations: they lack a modular mechanism for adaptive capacity allocation and self-correction, resulting in unreliable modeling. To tackle these challenges, we propose MoD-VLLM, a novel Modularized Dynamic-Granularity Video LLM framework for multi-event long video understanding, which seamlessly unifies temporal grounding and semantic understanding in an iterative, self-reflective manner. Specifically, we propose a Positive-Negative Video Segments Grounding module and a Modularized Dynamic-Granularity Reflection module, which form a closed loop to progressively localize the question-related video segments. The grounding module instructs a Video LLM to distinguish relevant from irrelevant video segments based on the video question. The reflection module employs a modularized scheduler that dynamically selects fine-grained encoding for relevant positive segments to capture detailed perception and coarse-grained encoding for negative segments to efficiently maintain global context, thereby enabling adaptive granularity allocation. We further propose a dynamic-granularity reinforcement learning strategy, allowing MoD-VLLM to learn optimal grounding policies and dynamic granularity visual representation jointly. Moreover, we propose MEventBench, a challenging Multi-Event Long Video Benchmark to evaluate models for complex long video understanding and reasoning. Extensive experiments on several long video understanding benchmarks and our MEventBench demonstrate that the proposed MoD-VLLM is able to significantly outperform state-of-the-art baselines.**

***Index Terms*—Long Video Understanding, Video LLM**

I. Introduction

Recently, Video Large Language Models (Video LLMs) have made significant advancements. By aligning Large Language Models with vision encoders through instruction tuning, video LLMs are able to perform various video understanding tasks within a unified framework [1], [2].

Despite these advances, existing Video LLMs still exhibit fundamental limitations in long video scenarios. Unlike short videos, long videos commonly contain multiple events that are sparsely and discontinuously distributed over time. In essence, multi-event long video understanding presents a fundamental dilemma: the tension between limited visual token budgets and the need to comprehensively capture multiple events. Existing long-video methods fail to resolve this challenge.

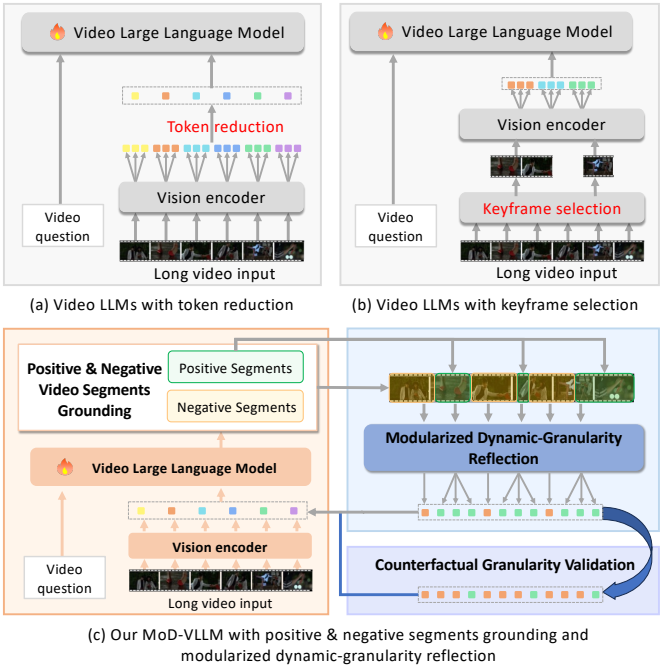

Fig. 1. Conceptual comparison of different video LLM paradigms. Token reduction methods cause critical detail loss when processing long videos. Keyframe selection suffers from irreversible error propagation when localizing wrong segments. Our MoD-VLLM overcomes these issues via iterative, self-corrective grounding and modularized encoding with dynamic granularity.

Specifically, approaches shown in Figure 1(a) employ token pruning or frame sampling strategies to reduce the visual tokens [3], [4]. Such static visual modeling strategies overlook the semantic importance differences among segments and fail to capture fine-grained details that are critical for answering the query. Other methods shown in Figure 1(b) adopt two-stage strategies [5], [6], where keyframes are first localized and then processed for detailed understanding. These approaches lack effective self-reflection and error-correction during localization, causing errors to accumulate once initial predictions are biased. As a result, existing methods lack a mechanism to adaptively allocate modeling capacity based on segment importance and fail to perform effective self-correction, leading to unreliable modeling of multiple key events.

*Corresponding authors: Xin Wang and Wenwu Zhu.

To address these challenges, as shown in Figure 1(c), we propose **MoD-VLLM**, a novel **Mo**dularized **D**ynamic-Granularity **V**ideo **LLM** for multi-event long video understanding. Specifically, our MoD-VLLM framework consists of i) a Positive-Negative Video Segments Grounding module, which first instructs the video LLM to identify and distinguish relevant and irrelevant video segments regarding the multi-event question, and ii) a Modularized Dynamic-Granularity Reflection module, which adopts a modular scheduling mechanism to adaptively allocate dynamic granularity across video frames. It performs fine-grained encoding on relevant segments to capture detailed semantics, while applying coarse-grained encoding to irrelevant segments to efficiently maintain global visual context. The reflection module provides a closed-loop feedback mechanism, where counterfactual granularity verification serves as a basic reflection process to mitigate localization errors, progressively guiding the model toward more accurate video segment localization. To optimize the proposed MoD-VLLM framework, we further propose a dynamic-granularity reinforcement learning strategy for optimal grounding policies generation and dynamic granularity representation learning. In addition, we construct and propose **MEventBench**, a **M**ulti-**Event Bench**mark for long video understanding, which contains long videos with questions about several key video segments and temporal dependencies. Extensive experiments on several long video understanding benchmarks and our MEventBench show that the proposed MoD-VLLM framework is able to significantly outperform existing state-of-the-art baselines.

In summary, we make the following contributions:

- We propose MoD-VLLM, a modularized dynamic-granularity video LLM framework for accurate multi-event long video understanding, which is capable of conducting dynamic granularity iteration through positive-negative video segments grounding and modularized dynamic-granularity reflection.
- We propose a dynamic-granularity reinforcement learning strategy to optimize the MoD-VLLM framework.
- We propose MEventBench, a multi-event benchmark for long video understanding.
- We conduct extensive experiments on several long video understanding benchmarks and our MEventBench, which demonstrate that MoD-VLLM can outperform state-of-the-art baseline methods.

II. Related Work

Recently, advances in Large Language Models [7] have led to the development of Video LLMs for temporal video understanding through cross-modal alignment [8]. Models such as Video-LLaMA [9] typically employ visual encoders to extract frame-level features and project them into the LLM's space, sharing ideas with image-based LLMs [10]. However, when dealing with long videos, these models face a critical challenge: processing all frames at high sampling rates exceeds token limits, while reducing sampling rates risks losing essential temporal information and visual details.

To address the token limitation in long video understanding, some methods focus on compressing visual tokens. Video LLMs like LongVU [4] reduce token counts, while MovieChat [3] uses memory mechanisms for efficient compression. However, these approaches often discard fine-grained spatial-temporal details needed for complex video reasoning. Other methods adopt a two-stage, coarse-to-fine strategy. Approaches like VideoTree [6] and Adaptive Keyframe Selection(AKS) [5] first identify key segments coarsely, then analyze them in detail. While this strategy suffers from error propagation if initial localization fails, and the two stages are typically trained separately, leading to suboptimal alignment between localization and understanding.

III. Method

We introduce our MoD-VLLM framework in this section. Inspired by the application of video grounding for different video tasks [11], [12] and modularization designs for the multimodal large language models [13], we introduce similar approaches to advance multi-event understanding in long videos. As shown in Figure 2, MoD-VLLM designs a Positive-Negative Video Segments Grounding module and a Modularized Dynamic-Granularity Reflection module, which operate iteratively for multi-event long video understanding. Given a long video input, the grounding module first instructs the video LLM to identify video segments relevant to the question (positive segments) and treat the remaining as irrelevant (negative segments). Based on this guidance, the reflection module then employs a modularized encoding scheduler that dynamically selects from a set of pre-defined granularity modules (varying in per-frame token counts and sampling rates). It represents positive segments with fine-grained tokens for detailed perception, while encoding negative segments coarsely to preserve global context. This dynamic granularity representation replaces the original uniform-granularity tokens and is fed back to the grounding module for iterative refinement. Over several iterations, the video LLM progressively refines both the grounding policy and the final answer to the video question.

A. Positive-Negative Video Segments Grounding

Our grounding module uses two branches: uni-granularity representation and question-frame similarity. The video LLM then identifies question-relevant segments as positive and the rest as negative.

Uni-granularity representation. Given a long video $v \in R^{T\times H\times W\times C}$, we uniformly sample N frames $\widetilde{v} \in R^{N\times H\times W\times C}$. Each frame is encoded by a vision transformer:

$$\{v_i^{cls}, v_i^1, v_i^2, \ldots, v_i^{patch}\} = ViT(\widetilde{v}_i), \quad i = 1, \ldots, N, \quad (1)$$

and the global feature v_i^{cls} is projected into visual tokens:

$$\begin{aligned} z_i &= f(v_i^{cls}), \quad i = 1, \ldots, N, \\ Z &= \{z_i\} \in R^{N\times L\times d}, \end{aligned} \quad (2)$$

where Z is the visual token sequence, d is the hidden dimension, and L is the tokens per frame.

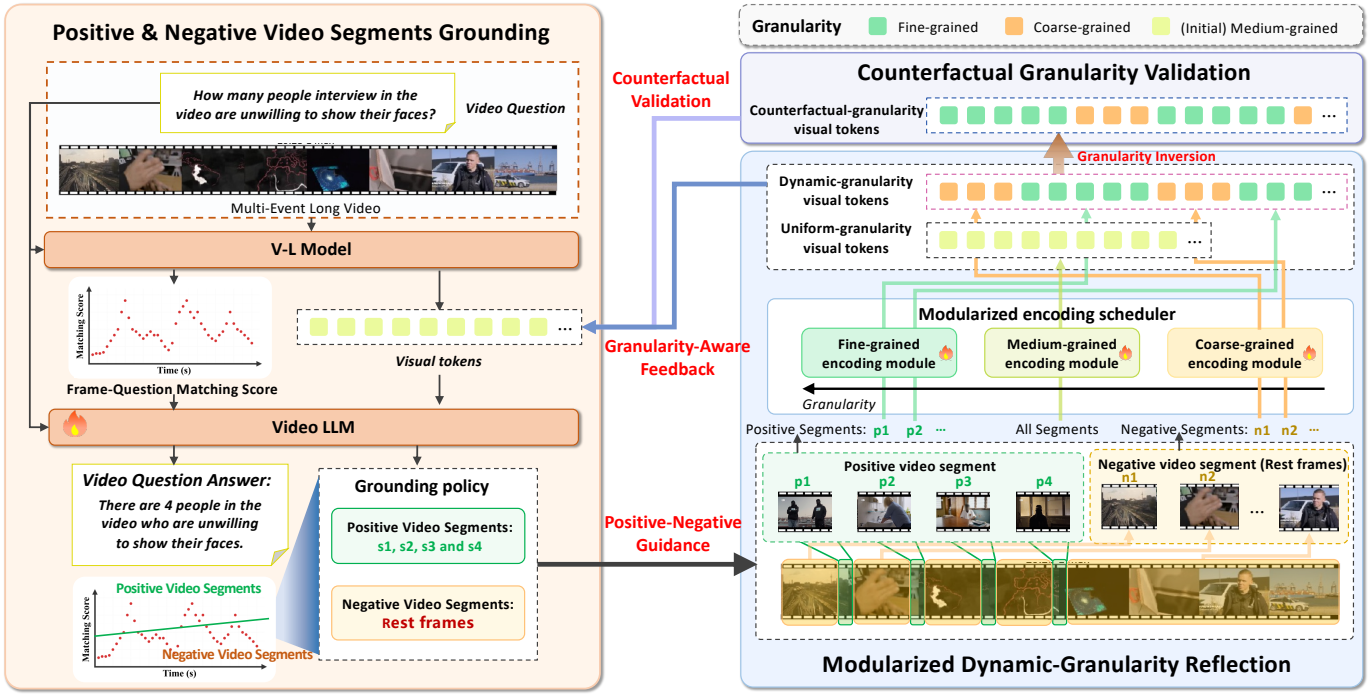

Fig. 2. Our MoD-VLLM framework for multi-event long video understanding. The positive-negative video segments grounding module instructs the video LLM to generate a grounding policy that identifies question-relevant segments. The modularized dynamic-granularity reflection module then differentiates encoding granularity: positive segments are encoded in a fine-grained manner for detail, while negative segments are encoded coarsely to retain global context. Through iterative updates of the visual token sequence, the model refines grounding policies and answers. An optional counterfactual granularity sequence (with reversed granularity assignments) is used for validation to mitigate error propagation.

Question-frame similarity representation. Using the same frames and features, we compute similarity between v_i^{cls} and question embedding q:

$$sim_i = \frac{v_i^{cls}}{\|v_i^{cls}\|} \frac{q}{\|q\|}, \\ Sim = \{sim_i\} \in R^N. \tag{3}$$

Multi-event video grounding. To combine the two obtained branches' visual information into the video LLM, we design a structured input paradigm that systematically integrates visual tokens with the question-frame similarity guidance. Specifically, we formulate this through a multi-modal prompt template:

$$input = Concat(p_{inst}, Z, p_{Sim}, p_{task}), \tag{4}$$

where p_{inst} denotes the instruction prompt 'Based on the following visual tokens and frame-query similarity scores:', p_{Sim} denotes the textual embeddings of the above question-frame similarity sequence, and p_{task} denotes the task prompt 'Identify which parts of the video segments are relevant to the question query q'. The complete prompt also includes some other examples of outputs and the necessary description for the input video information, such as video duration. This design enables the video LLM to simultaneously process visual semantics and similarity guidance. The question-frame similarities with higher scores would directly guide the video LLM to pay attention to the fact that these video segments are semantically close to the long video question. The final segments' grounding output would be responded to by the video LLM in a restricted JSON format:

$$output = VidLLM(input) \\ = [[s_1, e_1], ..., [s_j, e_j]], \tag{5}$$

where s and e denote the start time and end time for one grounding segment, and the number of j is irregular depending on the related video events to the question. These video segments are currently considered positive video segments more related to the video question, while the rest are considered negative video segments.

B. Modularized Dynamic-Granularity Reflection

Based on the grounding results, we apply dynamic granularity encoding to differentiate positive and negative segments.

Modularized encoding modules with dynamic granularity. We build a set of encoding modules $\mathcal{E} = \{\phi_1, \ldots, \phi_{b-1}, \phi_b, \phi_{b+1}, \ldots, \phi_K\}$ by fine-tuning the projection layers between the visual encoder and the LLM. A shared ViT backbone extracts features, while switchable projections produce tokens at different granularities. By combining modules with different per-frame token budgets and sampling rates, we can flexibly tailor the information density of the visual representation to meet diverse video understanding requirements. $\phi_m = \phi_b$ represents the initial uniform encoding granularity on the video segments grounding. Coarser module $\phi_c = \phi_{b-1}$ reduces tokens per frame for compression and finer modules $\phi_f = \phi_{b+1}$ increases tokens for detail. $\{\phi_i, i < b-1\}$ denotes the coarse-grained encoding module applying a lower frame sampling rate compared to the initial setting, while

$\{\phi_i, i > b+1\}$ denotes the fine-grained encoding module applying a higher frame sampling rate, and their sampling rates are sorted from low to high. Based on the constructed group of encoding modules, we can assert that if $p > n$, then encoding module ϕ_p would contain richer visual information than ϕ_n.

Modularized encoding scheduler. For detailed perception on the key video segments, we aim to encode positive segments as fine-grained as possible without exceeding the LLM's token limit L_{max}. Let r_k be the token generation rate (tokens/frame) of module ϕ_k. After grounding, we obtain positive frame count T_p and negative frame count T_n, with proportion $\rho = T_p/T$. Granularity levels for negative and positive segments are computed as:

$$\begin{aligned} \text{GranularityLevel}_{neg}(\rho) &= b - \lfloor (b-1)\cdot \rho^{\alpha} \rfloor, \\ \text{GranularityLevel}_{pos}(\rho) &= b + \lfloor (K-b)\cdot (1-\rho^{\alpha}) \rfloor, \end{aligned} \tag{6}$$

where $\alpha \in [1.5, 2.0]$ is an adjustable factor. The corresponding encoding modules are selected as:

$$\begin{aligned} \phi_n &= \phi_{\max(1,\min(\text{GranularityLevel}_{neg}(\rho),b))}, \\ \phi_p &= \phi_{\min(K,\max(\text{GranularityLevel}_{pos}(\rho),b))}. \end{aligned} \tag{7}$$

The total token budget must satisfy:

$$T_p \cdot r_p + T_n \cdot r_n \le L_{max}. \tag{8}$$

If the budget is exceeded, we first downgrade ϕ_n (negative segments) to a coarser module, then ϕ_p if needed. Conversely, if tokens are far below L_{max}, we upgrade ϕ_p first. This can be summarized as the optimization:

$$\begin{aligned} \underset{\phi_p,\phi_n \in \mathcal{E}}{\text{maximize}} \quad & w_p \cdot r_p - w_n \cdot r_n, \\ \text{subject to} \quad & T_p \cdot r_p + T_n \cdot r_n \le L_{\text{max}}, \\ & r_p \ge r(\phi_{\text{base}}) \ge r_n, \\ & \phi_p \in \{\phi_b, \phi_{b+1}, \ldots, \phi_K\}, \\ & \phi_n \in \{\phi_1, \phi_2, \ldots, \phi_b\}, \end{aligned} \tag{9}$$

where w_p, w_n are preference weights for positive/negative segments.

Given the selected modules, we encode each continuous video segment s_i with its assigned module $\phi_i \in \{\phi_p, \phi_n\}$, producing the final dynamic granularity token sequence:

$$Z' = \{\phi_1(s_1), \phi_2(s_2), \ldots, \phi_t(s_t)\}, \tag{10}$$

which replaces the original uniform token sequence Z and is fed back to the grounding module for iterative refinement.

C. Dynamic Granularity Iteration with RL

We combine the grounding and reflection modules into an iterative framework, and optimize the MoD-VLLM via a dynamic granularity reinforcement learning strategy.

Dynamic granularity iteration. To enable multi-event understanding, we iteratively alternate between grounding and reflection. After the first iteration, we replace the original uniform token sequence Z with the dynamic granularity tokens Z' for subsequent grounding. Two multimodal inputs are constructed:

$$\begin{aligned} input_{grounding} &= Concat(p_{inst}, Z', p_{Sim}, p_{task}), \\ input_{answering} &= Concat(Z', q), \end{aligned} \tag{11}$$

where $input_{grounding}$ follows the same structure as Eq. 4, and q is the original question. Both inputs include textual instructions about the temporal arrangement of the interleaved tokens. The answer is obtained by feeding $input_{answering}$ to the video LLM.

To mitigate error propagation from missed relevant segments (which would be encoded coarsely), we introduce a counterfactual validation in the first reflection step. In addition to Z', we generate a counterfactual token sequence:

$$Z'' = \{\phi'_1(s_1), \phi'_2(s_2), \ldots, \phi'_t(s_t)\}, \tag{12}$$

where ϕ'_i is assigned by the scheduler under reversed positive-negative classification. Z'' is used only in the first iteration. During the second grounding step, the model processes both Z' and Z'' and merges the two grounding outputs to reduce error propagation.

Dynamic granularity reinforcement learning. To improve grounding accuracy, we optimize our MoD-VLLM framework using direct preference optimization (DPO) within our iterative framework. During grounding, the model is instructed to generate multiple candidate policies $\{p_i\}$ (each in the JSON format of Eq. 20). Each policy leads to a dynamic token sequence Z_i via the scheduler.

A high-quality token sequence should represent question-relevant frames in detail while compressing irrelevant parts. Therefore, feeding a well-structured Z_i to the video LLM should produce an answer closer to the ground truth, having a lower cross-entropy loss. Conversely, a poor sequence would increase loss. Inspired by RLHF [20], we apply DPO [21], [22] using the cross-entropy scores as implicit preferences. Let p_w be the policy with the smallest loss and p_l a policy with a larger loss. The DPO objective is:

$$\begin{aligned} &L_{DPO}(\pi_\theta; \pi_{ref}) = \\ &- E_{(q,v,p_w,p_l)\sim\mathcal{D}}[\log \sigma(\beta \log \frac{\pi_\theta(p_w|q,v)}{\pi_{ref}(p_w|q,v)} - \beta \log \frac{\pi_\theta(p_l|q,v)}{\pi_{ref}(p_l|q,v)})], \end{aligned} \tag{13}$$

where π_θ is the trainable video LLM, π_{ref} is a fixed reference video LLM, σ is the sigmoid function, β is a scaling parameter, and $\mathcal{D}$ is the training dataset.

IV. Experiment

Implementations. We use LLaVA-Video(7B) [19] as the video LLM backbone. Fine-tuned encoding modules ϕ_{coarse}, ϕ_{medium}, ϕ_{fine} produce 36, 64, and 169 tokens per frame respectively. For initial grounding, we compute similarity scores with CLIP [23] and sample 128 frames using the medium-grained encoder (64 tokens/frame), with initial sampling rate $N_T = 128/T_{video}$. The full encoding set includes $\{(0.2N_T, \phi_{coarse}),\ (0.5N_T, \phi_{coarse}),\ (N_T, \phi_{coarse}),\ (N_T, \phi_{medium}),\ (N_T, \phi_{fine}),\ (2N_T, \phi_{fine}),\ (3N_T, \phi_{fine})\}$.

TABLE I
PERFORMANCE COMPARISON ON LONG VIDEO UNDERSTANDING BENCHMARKS. * DENOTES APPLYING GPT-4 AS LLM FOR THE MAIN RESULT.

Models	Size	VideoMME				Lvbench	MLVU Dev	MEventBench
		Short	Medium	Long	Overall			
Duration(min)		$\leq$2	4$\sim$15	30$\sim$60	1$\sim$60	30$\sim$140	3$\sim$120	15$\sim$100
Video-RAG [14]	7B	66.4	60.2	59.8	62.1	39.3	65.2	52.5
LongVU [4]	7B	64.7	58.2	59.5	60.9	38.3	65.4	51.2
Video-XL [15]	7B	67.4	60.7	54.9	61.0	37.7	64.9	49.3
Video-XL2 [16]	8B	73.7	65.9	60.2	66.6	<u>48.4</u>	74.8	53.8
VITA 1.5 [17]	7B	67.0	54.2	47.1	56.1	32.1	60.2	45.9
AKS [5]	7B	-	-	-	65.4	46.8	70.1	55.7
VideoTree [6]	*	-	-	54.2	-	-	-	55.4
Qwen2.5-VL [18]	7B	**81.4**	<u>70.8</u>	<u>62.6</u>	<u>71.6</u>	45.3	<u>75.5</u>	<u>60.8</u>
LLaVA-Video [19]	7B	78.0	69.3	61.8	69.7	43.2	71.4	59.2
Ours-MoD-VLLM w/ LLaVA-Video	7B	<u>80.3</u>	**72.4**	**66.9**	**73.2**	**49.6**	**78.2**	**64.8**

We run three dynamic granularity iterations per video. Further parameters and training details are in the supplement.

Evaluation. We evaluate on long-video benchmarks: VideoMME [24], Lvbench [25], and MLVU [26], reporting accuracy on multiple-choice questions. To assess complex multi-event reasoning, we construct **MEventBench**, containing 1200 video-question pairs from VideoMME, Longvideobench [27], InfiniBench [28], and CG-Bench [29]. Each video includes at least three question-related segments dispersed in time. In summary, we classify the type of our selected multi-event data into multi-event counting, ordering, and reasoning. More details are provided in the supplement.

A. Main Result on Long Video Understanding

Table I reports results on long-video benchmarks. Our MoD-VLLM achieves higher average accuracy on longer videos (¿2min), outperforming state-of-the-art methods of similar scale. These include open-source video LLMs (LongVU [4], Video-XL2 [16], Qwen2.5-VL [18]) and two-stage coarse-to-fine approaches (Video-RAG [14], AKS [5], VideoTree [6]). On our MEventBench, MoD-VLLM also surpasses the strongest baseline, validating the effectiveness of dynamic granularity iteration for complex multi-event reasoning.

B. Ablation Study

In this section, we provide ablation results about our concerned questions as follows.

TABLE II
ABLATION STUDY ON DIFFERENT MULTI-EVENT TASKS.

Method	MEventBench			
	Counting	Ordering	Reasoning	Overall
Qwen2.5-VL	64.2	61.1	57.2	60.8
LLaVA-Video	62.0	59.4	56.3	59.2
MoD-VLLM	69.4	64.2	62.1	64.8

Performance on different multi-event tasks. Table II shows results across task categories on MEventBench. MoD-VLLM outperforms strong video LLM baselines in all categories, achieving an overall accuracy of 64.8%, which is 4.0% and 5.6% higher than Qwen2.5-VL and LLaVA-Video, respectively. The largest gain appears in Counting (69.4%, +5.2% over Qwen2.5-VL), highlighting our dynamic granularity iteration's ability to localize sparse, distributed events. Improvements in Ordering (64.2%) and Reasoning (62.1%) further confirm the framework's strength in modeling temporal relations and complex inference, enabled by iterative refinement and positive-negative context modeling. These results demonstrate that allocating fine-grained representation to critical segments while maintaining global context is key to comprehensive long-video understanding.

TABLE III
ABLATION STUDY ON INPUT MODALITIES FOR GROUNDING.

Method	VideoMME	MEventBench		
		Counting	Ordering	Reasoning
MoD-VLLM w/ V+S	73.2	69.4	64.2	62.1
MoD-VLLM w/ V	67.1	65.8	60.3	57.7
MoD-VLLM w/ S	62.5	59.6	55.0	51.8

Why apply both visual representation and similarity sequence for grounding instruction? We evaluate the contribution of visual tokens (V) and similarity scores (S) to grounding. As Table III shows, using both (V+S) achieves the best performance across VideoMME and all MEventBench tasks. Relying only on visual tokens (V) or only similarity (S) leads to clear drops, with similarity-alone performing weakest.

The advantage of V+S stems from two factors. First, CLIP-based similarity, trained on short image-text pairs, does not fully capture the alignment between long questions and video frames. Second, without similarity as a prior, the video LLM initially trained for QA, struggles to ground multiple segments under DPO, due to the cold start in a more complex task. The visual tokens provide dense semantics while the similarity scores offer a direct question-frame relevance signal, enabling robust mutual verification and improving grounding accuracy.

Case Analysis. To demonstrate MoD-VLLM's complex video reasoning, Figure 3 visualizes a multi-event counting example with the question: "How many items did the woman take out of her bag?" The task is challenging because the key segments are distributed over time and require detailed perception (e.g., two phones are taken simultaneously in one segment). Through dynamic granularity iteration, our model is able to progressively refine its attention and produce a granularity encoding that highlights all relevant segments where items are removed.

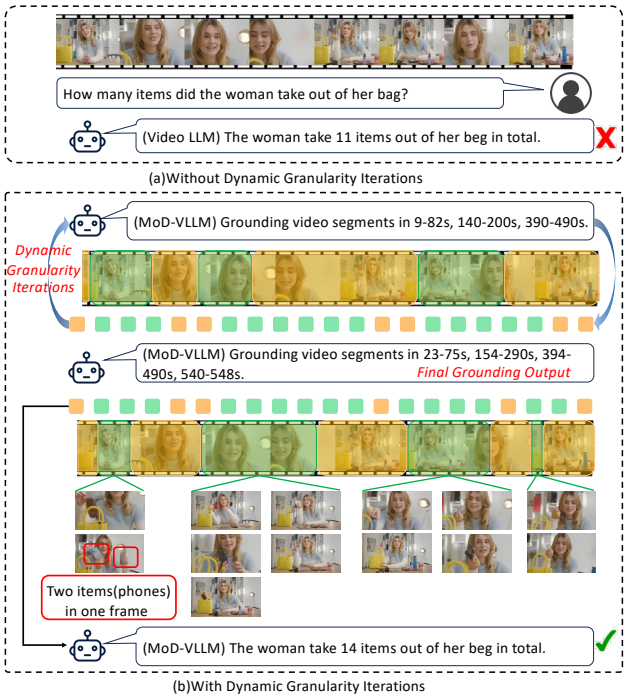

Fig. 3. A qualitative example of multi-event long video understanding. The duration is about 10 minutes long.

V. Conclusion

In this paper, we propose MoD-VLLM, a novel modularized dynamic-granularity video LLM framework with reflection for multi-event long video understanding. Specifically, we design the Positive-Negative Video Segments Grounding module and the Modularized Dynamic-Granularity Reflection module to enable dynamic granularity iteration on complex long videos. We further propose a dynamic-granularity reinforcement learning strategy to optimize MoD-VLLM with dynamic granularity representation. We construct and propose MEventBench, a multi-event benchmark for long video understanding. Extensive experiments on several long video understanding benchmarks and our MEventBench demonstrate that the proposed MoD-VLLM framework outperforms existing state-of-the-art baselines on processing complex long videos with multiple question-relevant segments.

References

[1] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, vol. 36, 2023, pp. 34 892–34 916.

[2] B. Huang, X. Wang, H. Chen, Z. Song, and W. Zhu, "Vtimellm: Empower llm to grasp video moments," in *CVPR*, 2024, pp. 14 271–14 280.

[3] 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 *CVPR*, 2024, pp. 18 221–18 232.

[4] X. Shen, Y. Xiong, C. Zhao, L. Wu, J. Chen, C. Zhu, Z. Liu, F. Xiao, B. Varadarajan, F. Bordes *et al.*, "Longvu: Spatiotemporal adaptive compression for long video-language understanding," in *ICML*, 2025.

[5] X. Tang, J. Qiu, L. Xie, Y. Tian, J. Jiao, and Q. Ye, "Adaptive keyframe sampling for long video understanding," in *CVPR*, 2025, pp. 29 118–29 128.

[6] Z. Wang, S. Yu, E. Stengel-Eskin, J. Yoon, F. Cheng, G. Bertasius, and M. Bansal, "Videotree: Adaptive tree-based video representation for llm reasoning on long videos," in *CVPR*, 2025, pp. 3272–3283.

[7] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[8] X. Wang, Y. Zhou, B. Huang, H. Chen, and W. Zhu, "Multi-modal generative ai: Multi-modal llms, diffusions and the unification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[9] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," in *EMNLP*, 2023, pp. 543–553.

[10] R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, and S. Taşırlar, "Fuyu-8b: A multimodal architecture for ai agents," 2023.

[11] X. Wang, X. Lan, and W. Zhu, *Video Grounding and Its Generalization: From ID and Task-specific Models to OOD and Large Foundation Models*. Springer, 2025.

[12] W. Feng, X. Wang, H. Chen, Z. Zhang, and W. Zhu, "Multi-sentence video grounding for long video generation," in *ICME*. IEEE, 2025, pp. 1–6.

[13] Y.-W. Zhan, X. Wang, P. Mao, T. Feng, R. Wang, and W. Zhu, "Modularagent: A task-aware modular framework for joint optimization of multimodal large language models and world models," in *CVPR*, 2026.

[14] Y. Luo, X. Zheng, X. Yang, G. Li, H. Lin, J. Huang, J. Ji, F. Chao, J. Luo, and R. Ji, "Video-rag: Visually-aligned retrieval-augmented long video comprehension," *arXiv preprint arXiv:2411.13093*, 2024.

[15] Y. Shu, Z. Liu, P. Zhang, M. Qin, J. Zhou, Z. Liang, T. Huang, and B. Zhao, "Video-xl: Extra-long vision language model for hour-scale video understanding," in *CVPR*, 2025, pp. 26 160–26 169.

[16] M. Qin, X. Liu, Z. Liang, Y. Shu, H. Yuan, J. Zhou, S. Xiao, B. Zhao, and Z. Liu, "Video-xl-2: Towards very long-video understanding through task-aware kv sparsification," *arXiv preprint arXiv:2506.19225*, 2025.

[17] C. Fu, H. Lin, X. Wang, Y.-F. Zhang, Y. Shen, X. Liu, H. Cao, Z. Long, H. Gao, K. Li *et al.*, "Vita-1.5: Towards gpt-4o level real-time vision and speech interaction," *arXiv preprint arXiv:2501.01957*, 2025.

[18] 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.

[19] Y. Zhang, J. Wu, W. Li, B. Li, Z. MA, Z. Liu, and C. Li, "Llava-video: Video instruction tuning with synthetic data," *Transactions on Machine Learning Research*.

[20] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[21] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *NeurIPS*, vol. 36, pp. 53 728–53 741, 2023.

[22] Z. Song, X. Wang, Z. Qian, H. Chen, L. Huang, H. Xue, and W. Zhu, "Modularized self-reflected video reasoner for multimodal llm with application to video question answering," in *ICML*, 2025.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*. PMLR, 2021, pp. 8748–8763.

[24] C. Fu, Y. Dai, Y. Luo, L. Li, S. Ren, R. Zhang, Z. Wang, C. Zhou, Y. Shen, M. Zhang *et al.*, "Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis," in *CVPR*, 2025, pp. 24 108–24 118.

[25] W. Wang, Z. He, W. Hong, Y. Cheng, X. Zhang, J. Qi, M. Ding, X. Gu, S. Huang, B. Xu *et al.*, "Lvbench: An extreme long video understanding benchmark," in *ICCV*, 2025, pp. 22 958–22 967.

[26] J. Zhou, Y. Shu, B. Zhao, B. Wu, Z. Liang, S. Xiao, M. Qin, X. Yang, Y. Xiong, B. Zhang *et al.*, "Mlvu: Benchmarking multi-task long video understanding," in *CVPR*, 2025, pp. 13 691–13 701.

[27] H. Wu, D. Li, B. Chen, and J. Li, "Longvideobench: A benchmark for long-context interleaved video-language understanding," *NeurIPS*, vol. 37, pp. 28 828–28 857, 2024.

[28] K. Ataallah, C. Gou, E. Abdelrahman, K. Pahwa, J. Ding, and M. Elhoseiny, "Infinibench: A comprehensive benchmark for large multimodal models in very long video understanding," *arXiv preprint arXiv:2406.19875*, 2024.

[29] G. Chen, Y. Liu, Y. Huang, B. Pei, J. Xu, Y. He, T. Lu, Y. Wang, and L. Wang, "Cg-bench: Clue-grounded question answering benchmark for long video understanding," in *ICLR*, 2025.

[30] S. Wang, G. Chen, D.-a. Huang, Z. Li, M. Li, G. Li, J. M. Alvarez, L. Zhang, and Z. Yu, "Videoitg: Multimodal video understanding with instructed temporal grounding," *arXiv preprint arXiv:2507.13353*, 2025.

SUPPLEMENT

A. Training details

The overall training of our framework is conducted under 8 NVIDIA A100-40GB GPUs. Our framework is built upon the LLaVA-Video architecture as the backbone model. To achieve dynamic granularity control, we introduce additional adaptive pooling modules in the projection layers between the visual encoder and the LLM, which systematically regulate the number of tokens per video frame. This allows us to construct three distinct encoding modules operating at different granularity levels: 36 tokens per frame (coarse), 64 tokens per frame (medium), and 169 tokens per frame (fine), representing progressively increasing spatial detail. For the dynamic granularity encoding scheduler, we set the adjustable factor α to 1.5.

Module training details. For training the encoding modules in the first stage, we employ the LLaVA-Video-178K [19] dataset to finetune the video LLM. During the reinforcement learning stage, we employ the VideoITG dataset [30], which contains 40K videos and 500K instruction-guided annotations.

TABLE IV
IMPLEMENTATION DETAILS

Config	Finetuning	Reinforcement Learning
Video LLM	LLaVA-Video-7B	
Optimizer	AdamW	
Epochs	10	5
Warmup ratio	0.05	
Learning rate	2e-5	1e-5
Batch size	1	
Gradient accumulation steps	16	
Number of Tokens per Frame	36 64 169	mixed
Training cost	380h in total	800h

1) Algorithm details: **Modularized encoding scheduler.** Compared to the original uni-granularity representation, we hope to represent the positive video segment as fine-grained as possible with a higher ratio of sampled frames, but not exceeding the token limitations L_{max}[1] of the video LLM. Defining r_k is the token generation rate with the number of frames (tokens/frame) for the encoding module ϕ_k, since we are aware that the initial video segments grounding is executable with the medium-grained encoding module, so we have:

$$T \cdot r_{base} \lesssim L_{max}, \tag{14}$$

which represents the number of visual tokens is less than, but very close to the limit of LLM. Based on the grounding guidance, we can acquire the total number of positive video frames T_p and negative video frames T_n. Considering $\rho = \frac{T_p}{T}$ as the proportion of positive sample frames, we explore the granularity level as:

$$\begin{aligned} \text{GranularityLevel}_{neg}(\rho) &= b - \lfloor (b-1) \cdot \rho^{\alpha} \rfloor, \\ \text{GranularityLevel}_{pos}(\rho) &= b + \lfloor (K-b) \cdot (1-\rho^{\alpha}) \rfloor, \end{aligned} \tag{15}$$

[1]In actual implementation, L_{max} would be smaller than the max token limitations of LLM, to preserve part of space for processing long text prompt.

where α is an adjustable factor ranging from 1.5 to 2.0. According to the granularity level, we select the encoding module for positive segments and negative segments separately:

$$\begin{aligned} \phi_n &= \phi_{max(1, min(\text{GranularityLevel}_{neg}(\rho), b))}, \\ \phi_p &= \phi_{min(K, max(\text{GranularityLevel}_{pos}(\rho), b))}. \end{aligned} \tag{16}$$

Since the total token limitations should be fulfilled as:

$$T_p \cdot r_p + T_n \cdot r_n \leq L_{max}. \tag{17}$$

And we design our scheduler algorithm as shown in Algorithm 1. When we notice the total number of required tokens L_{total} higher than L_{max}, we would first try to downgrade the encoding module for negative segments to a more coarse-grained one, and if necessary downgrade the encoding module for positive segments, utill $L_{total} \leq L_{max}$. Meanwhile, if the L_{total} is far less than L_{max}, we will try to upgrade the granularity for the positive encoding module first.

Algorithm 1 Dynamic Granularity Encoding Scheduler

Require: $T_p, T_n, \mathcal{E}, L_{\text{max}}, \phi_p, \phi_n, \phi_{\text{base}} = \phi_b = \phi_{medium}$
Ensure: ϕ_p^*, ϕ_n^*

$\phi_p^* \leftarrow \phi_p, \phi_n^* \leftarrow \phi_n$
$L \leftarrow T_p \cdot r(\phi_p^*) + T_n \cdot r(\phi_n^*)$
while $L > L_{\text{max}}$ **do**
  if $\phi_n^* \neq \phi_1$ **then**
    $\phi_n^* \leftarrow \text{coarser}(\phi_n^*)$
  else if $\phi_p^* \neq \phi_{\text{base}}$ **then**
    $\phi_p^* \leftarrow \text{coarser}(\phi_p^*)$
  else
    break
  end if
  $L \leftarrow T_p \cdot r(\phi_p^*) + T_n \cdot r(\phi_n^*)$
end while
while $L \ll L_{\text{max}}$ **do**
  $\phi_p^{\text{temp}} \leftarrow \phi_p^*, \phi_n^{\text{temp}} \leftarrow \phi_n^*$
  if $\phi_p^{\text{temp}} \neq \phi_K$ **then**
    $\phi_p^{\text{temp}} \leftarrow \text{finer}(\phi_p^{\text{temp}})$
  else if $\phi_n^{\text{temp}} \neq \phi_{base}$ **then**
    $\phi_n^{\text{temp}} \leftarrow \text{finer}(\phi_n^{\text{temp}})$
  else
    break
  end if
  $L_{\text{new}} \leftarrow T_p \cdot r(\phi_p^{\text{temp}}) + T_n \cdot r(\phi_n^{\text{temp}})$
  if $L_{\text{new}} \leq L_{\text{max}}$ **then**
    $\phi_p^* \leftarrow \phi_p^{\text{temp}}, \phi_n^* \leftarrow \phi_n^{\text{temp}}, L \leftarrow L_{\text{new}}$
  else
    break
  end if
end while
return ϕ_p^*, ϕ_n^*

Overall, the selection based on the ratio of positive segments

could be summarized as the following optimization problem:

$$\begin{aligned} \underset{\phi_p,\phi_n\in\mathcal{E}}{\text{maximize}} \quad & w_p \cdot r_p - w_n \cdot r_n, \\ \text{subject to} \quad & T_p \cdot r_p + T_n \cdot r_n \leq L_{\max}, \\ & r_p \geq r(\phi_{\text{base}}) \geq r_n, \\ & \forall r_k = r(\phi_k), \phi_{base} = \phi_b, \\ & \phi_p \in \{\phi_b, \phi_{b+1}, \ldots, \phi_K\}, \\ & \phi_n \in \{\phi_1, \phi_2, \ldots, \phi_b\}, \end{aligned} \tag{18}$$

where w_p and w_n are the weight coefficients representing the preference for the positive video segments.

Obtaining the suitable encoding modules for positive video segments and negative ones, our dynamic granularity encoding would process video frames through interleaved fine-grained and coarse-grained representations according to the temporal distribution of positive and negative video segments. Given the partitioned and continus video segments $\{s_1, s_2, ..., s_t\}$ with corresponding granularity assignments $\{\phi_1, \phi_2, ..., \phi_t\}$ where $\phi_i \in \{\phi_p, \phi_n\}$, we form the final dynamic encoding visual tokens:

$$Z' = \{\phi_1(s_1), \phi_2(s_2), ..., \phi_t(s_t)\}, \tag{19}$$

where Z' would serve as granularity-aware feedback reflected to the positive-negative video segments grounding module, replacing the original input visual sequence Z.

Dynamic granularity reinforcement learning. Since the applied video LLM is not trained with localizing multiple related events in long videos, it is difficult to ensure the accuracy of the grounding policy through in-context prompt learning alone. To address that, we design a dynamic granularity reinforcement learning method to optimize the MoD-VLLM framework through our dynamic granularity iteration framework.

To begin with, during the positive-negative video segments grounding strategy, we prompt the video LLM to generate several multi-event grounding policies $\{p_i\}$, each policy p_i is in the same JSON format as $output$ in equation 20. Each policy corresponds to a group of positive video segments and negative ones, which would inform the dynamic granularity encoding scheduler to generate a sequence of dynamic encoding visual tokens Z_i. To ensure policy diversity and avoid convergence to a local optimum, we implement three key strategies: (1) we iteratively prompt the LLM with the instruction 'try another grounding policy different from the history ones' to generate distinct temporal segments; (2) we employ a counterfactual augmentation technique where the inverse of a generated policy (i.e., treating all non-selected segments as the positive result) is included as a valid policy candidate; and (3) we randomly generate several policies based on the video length and vary random seeds during each training sample. This multifaceted approach to policy generation prevents the DPO optimization from collapsing into a limited set of predictions and encourages the model to explore a broader and more robust solution space for segment grounding.

$$\begin{aligned} output &= VidLLM(input) \\ &= [[s_1, e_1], ..., [s_j, e_j]], \end{aligned} \tag{20}$$

Generally, compared to the token sequence representing irrelevant segments in fine-grained or relevant segments in coarse-grained, we can assert that a dynamic visual tokens sequence correctly represents all the question-related key video frames in a fine-grained manner and other irrelevant video frames only occupy a small amount of visual tokens through coarse-grained encoding, would be more helpful for the video LLM to understand a multi-event long video, since it provides detailed spatial-temporal visual context precisely where it is needed for detailed understanding and temporal reasoning, while decreases the noise from the global video information by compressing the irrelevant segments.

Therefore, if we input the video question with each sequence of dynamic encoding visual tokens to the video LLM, the correct sequence of visual tokens would help the video LLM to generate an answer closer to the ground truth of the video question, and the cross-entropy loss between them would be smaller. While an incorrect sequence of visual tokens tends to mislead the video LLM, resulting in an incorrect answer response and an increase in loss.

Inspired by the reinforcement learning from human feedback [20](RLHF), we apply direct preference optimization [21](DPO) based on the different cross-entropy scores of the visual token sequence inputs from the grounding policies. This method optimizes our MoD-VLLM framework without explicitly rewarding models and formulates the objective as:

$$\begin{aligned} &L_{DPO}(\pi_\theta; \pi_{ref}) = \\ &- E_{(q,v,p_w,p_l)\sim\mathcal{D}}[\log\sigma(\beta\log\frac{\pi_\theta(p_w|q)}{\pi_{ref}(p_w|q)} - \beta\log\frac{\pi_\theta(p_l|q)}{\pi_{ref}(p_l|q)})], \end{aligned} \tag{21}$$

where p_w represents the positive grounding policy leading to the smaller cross-entropy loss, and p_l denotes the negative policy with the larger loss. π_θ represents the video LLM to be optimized in this stage, and π_{ref} is a reference model initialized with the same video LLM but remains frozen.σ is the sigmoid function, and β is a controlling parameter.

As shown in Algorithm 2, our dynamic granularity reinforcement learning alternates between the DPO of grounding policy generation and the SFT of dynamic representation learning. We first employ direct preference optimization to enable the video LLM for more accurate grounding policy generation, and then adopt supervised fine-tuning on the video LLM to adapt the dynamic granularity visual token sequence input.

B. MEventBench.

Here, we provide the detailed data distribution of our proposed MEventBench dataset in Figure 4. We construct the multi-event long-video dataset as follows. First, we filter existing video benchmarks to retain only videos longer than 10 minutes. From these, we select samples whose original type-annotations include keywords such as "ordering," "counting,"

Algorithm 2 Dynamic Granularity Reinforcement Learning

Require: Video LLM $VidLLM$, Dynamic Granularity Module D, Positive-Negative Grounding Module G, dataset $\mathcal{D} = \{(v_i, q_i, y_i)\}_{i=1}^{N}$, training steps $Step$, gradient accumulate step s, numbers of policies per data n

Activate: $VidLLM$
for $t = 1$ **to** $Step$ **do**
  for $m = 1$ **to** s **do**
    initialize $\pi_\theta = VidLLM, \pi_{ref} = VidLLM$
    $i \leftarrow ((t-1)s + m - 1)\%N + 1$
    Prepare data (v_i, q_i, y_i) from $\mathcal{D}$
    Generate policies $p_1, p_2, ...p_n = G(\pi_{ref}, q_i, v_i)$
    for $j = 1$ **to** n **do**
      Grounding: $S_j = G(v_i, p_j)$
      Dynamic granularity encoding: $v_j = D(S_j)$
      Forward propagation: $\widehat{y} = VidLLM(q_i, v_j)$
      Compute $L_{CEj} = -y \log(\widehat{y}(q, v_j))$
    end for
    $p_w \leftarrow \arg\min_{\{p_j\}} L_{CE}$
    $p_l \leftarrow \{p_j, p \neq p_w\}$
    Optimize π_θ **with DPO loss**
  end for
  $VidLLM \leftarrow \pi_\theta$
  for $m = 1$ **to** s **do**
    $i \leftarrow ((t-1)s + m - 1)\%N + 1$
    Prepare data (v_i, q_i, y_i) from $\mathcal{D}$
    Generate single policy $p = G(VidLLM, q_i, v_i)$
    Grounding: $S_j = G(v_i, p_j)$
    Dynamic granularity encoding: $v_j = D(S_j)$
    Forward propagation: $\widehat{y} = VidLLM(q_i, v_j)$
    Optimize $VidLLM$ **with cross-entropy loss**
  end for
end for

or "reasoning." For the remaining data, we search the question texts for indicative phrases like "how many," "what order," and "why." The union of these two subsets yields several thousand candidate video–question pairs. We then manually inspect each candidate to ensure that the question indeed refers to multiple, temporally dispersed segments within the video rather than a single key video segment. The final dataset is assembled from the pairs that satisfy this multi-segment requirement. In summary, we classify the type of our selected multi-event data into multi-event counting, ordering, and reasoning.

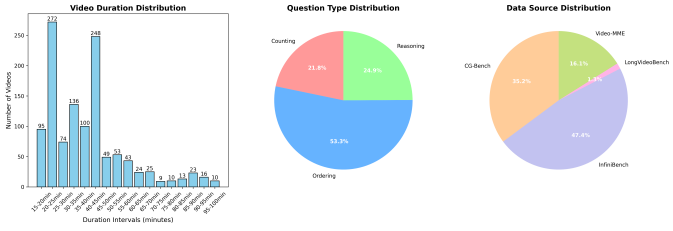

Fig. 4. Data distribution of MEventBench.

Counting tasks. These data require models to identify and enumerate multiple instances of specific events or objects distributed sparsely throughout long videos. The key challenge lies in accurately detecting and tallying occurrences that may be temporally dispersed, visually similar, or partially occluded, demanding robust temporal localization and fine-grained visual discrimination.

Ordering tasks. These data focus on temporal ordering by requiring models to arrange multiple events based on their occurrence patterns chronologically. The complexity arises from events with ambiguous temporal boundaries, overlapping timelines, or subtle contextual cues that determine their sequential relationships, necessitating precise temporal understanding.

Reasoning tasks. These data emphasize causal and logical inference by challenging models to identify underlying relationships between multiple events, such as cause-effect chains, prerequisite conditions, or intentional motivations. The difficulty stems from the need to integrate disparate visual evidence across long temporal spans and construct coherent narrative structures from sparse, distributed events.

Together, these three task types form a comprehensive evaluation framework that assesses models' capabilities in quantitative analysis (counting), temporal understanding (ordering), and higher-level cognitive processing (reasoning), covering the essential dimensions of complex multi-event video comprehension.