YOU ONLY FORWARD ONCE: RETHINKING VIDEO TEMPORAL GROUNDING IN MLLMS AS SINGLE-PASS PARALLEL BOUNDARY PROBABILITY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video Temporal Grounding (VTG) aims to localize the video segment corresponding to a natural-language query. Existing MLLM-based methods typically generate temporal boundaries autoregressively as textual tokens, creating a mismatch with interval-level localization and incurring sequential decoding overhead. To this end, we present **YOFO-VTG**, which reformulates VTG as parallel boundary-distribution prediction within a single MLLM forward pass. YOFO-VTG extracts video-aligned hidden states at the visual start and end delimiters of each temporal unit and maps them through the native language modeling head to predict start- and end-boundary probability distributions. To overcome the contextual restriction of causal attention, we adopt bidirectional attention, allowing each unit to integrate information from the full video-query sequence. This design retains the native probabilistic prediction mechanism of MLLMs while directly exploiting the complete sequence of video representations. We further introduce a two-stage strategy that first learns coarse boundary distributions with cross-entropy (CE) supervision and then refines the predicted boundaries through joint CE and differentiable IoU optimization. Extensive experiments on TimeLensBench demonstrate that YOFO-VTG achieves competitive localization accuracy while enabling substantially faster inference, highlighting **parallel probability distribution prediction** as an efficient paradigm for MLLM-based VTG.

1 INTRODUCTION

Video Temporal Grounding (VTG) aims to identify the temporal segment in a video that semantically matches a given natural-language query (Wu et al., 2025a; Zhang et al., 2023; Gao et al., 2017; Anne Hendricks et al., 2017). As a fundamental task in fine-grained video understanding, VTG underpins applications including video retrieval (Lin et al., 2023), video question answering (Gupta et al., 2025), and long-video memory retrieval (Yeo et al., 2026). Recent advances in multimodal large language models (MLLMs) have prompted their increasing adoption as VTG backbones, owing to their strong multimodal understanding and reasoning capabilities (Wu et al., 2025a).

Existing MLLM-based approaches improve temporal grounding along three main directions: explicitly injecting temporal cues into the model (Huang et al., 2024; Ren et al., 2024; Li et al., 2026b; Wu et al., 2025b), introducing task-specific post-training objectives (Wang et al., 2026; Feng et al., 2026; Zheng et al., 2026a), and augmenting the model with dedicated localization mechanisms (Fan et al., 2026; Pramanick et al., 2025; Zeng et al., 2025b; Yang et al., 2025). Despite these advances, most methods retain the causal autoregressive decoding paradigm of MLLMs, creating a structural mismatch between interval-level localization and token-by-token generation. Specifically, VTG requires jointly predicting the start and end boundaries that define a temporal interval, whereas standard MLLMs represent these boundaries as discrete tokens and generate them sequentially. Although recent post-training methods introduce localization-aware supervision or rewards (Wang et al., 2026; Feng et al., 2026; Zheng et al., 2026a), they retain the underlying token-generation formulation. Moreover, producing a single interval requires multiple autoregressive decoding steps (Wu et al., 2025b; Zhang et al., 2026; Zheng et al., 2026b), precluding direct one-pass localization. Consequently, existing approaches primarily improve temporal grounding within the generative interface rather than reconsidering how temporal intervals should be represented and predicted.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

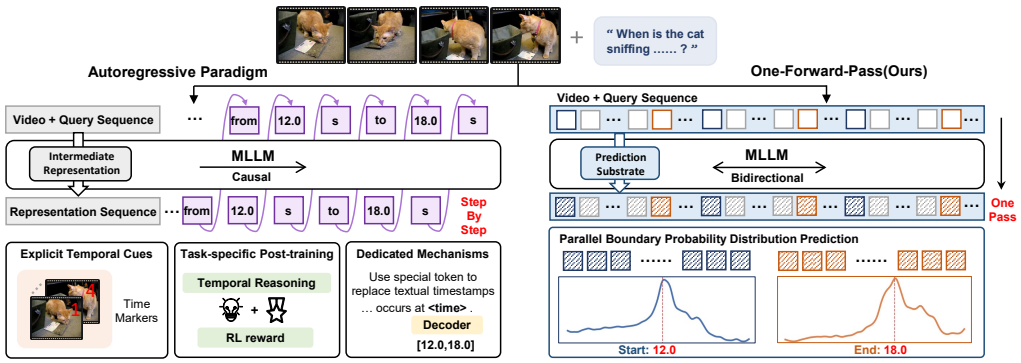


Figure 1: From autoregressive boundary generation to one-forward-pass temporal localization. Existing MLLM-based VTG methods generate temporal boundaries sequentially, optionally enhanced with explicit temporal cues, task-specific post-training, or dedicated mechanisms. In contrast, YOFO-VTG uses bidirectional video-query representation sequence to predict start- and end-boundary distributions in parallel within a single forward pass.

Notably, video-input MLLMs already produce dense contextual representations throughout their multimodal sequences. Rather than using these representations primarily to support language generation, we ask whether they can serve directly as the prediction substrate for temporal grounding. This motivates our central question: *Can VTG be reformulated as sequence-level parallel prediction over native MLLM representations?* We answer this question with **YOFO-VTG**, a new MLLM-based VTG paradigm that repurposes the model's native probabilistic prediction capability for temporal localization. As illustrated in Figure 1, YOFO-VTG extracts temporally-aligned hidden states and simultaneously estimates the likelihood of each temporal position being the start or end of the target segment, naturally yielding two boundary distributions over the entire video sequence. To allow the prediction at each position to leverage information from the entire sequence, we relax the MLLM's causal attention mask to enable bidirectional attention. In this manner, our method reformulates VTG as parallel prediction of start and end probability distributions over the video sequence.

Benefiting from this reformulation, YOFO-VTG can retain the standard SFT training paradigm of MLLMs while incorporating task-specific auxiliary losses to directly optimize temporal overlap. Moreover, since the video representation sequence can be obtained in a single forward pass of the MLLM, our approach directly produces the start and end probability distributions without iterative decoding, substantially reducing inference latency. We evaluate YOFO-VTG under the zero-shot setting on TimeLensBench (Zhang et al., 2026), using models with 2B, 4B, and 8B parameters. Across different scales, YOFO-VTG achieves competitive performance against state-of-the-art methods while maintaining substantially lower inference latency. These results demonstrate the feasibility and effectiveness of reformulating MLLM-based VTG from autoregressive generation into sequence-level parallel prediction of temporal probability distributions.

Our contributions are summarized as follows:

- We propose YOFO-VTG, a new MLLM-based VTG paradigm that revisits the underutilized video representation sequence and naturally inherits the MLLM's native probabilistic prediction capability, transforming token-level iterative generation into sequence-level parallel probability prediction.
- We instantiate this paradigm through bidirectional contextualization of temporally-aligned hidden states and end-to-end training that combines SFT with a differentiable IoU objective, directly aligning model optimization with temporal localization quality.
- Extensive experiments on TimeLensBench demonstrate competitive performance against state-of-the-art methods across 2B, 4B, and 8B MLLMs, while achieving substantially lower single-inference latency through single-forward-pass temporal boundary prediction.

2 Related Work

2.1 MLLM-based Video Temporal Grounding

Video temporal grounding has traditionally been formulated as a direct localization problem over video-text representations, with prior work exploring moment proposals, span prediction, temporal interaction, and proposal-free localization (Gao et al., 2017; Anne Hendricks et al., 2017; Zhang et al., 2020b;a; Mun et al., 2020; Xiao et al., 2021; Nan et al., 2021; Mu et al., 2024; Lei et al., 2021). With the emergence of multimodal large language models (MLLMs), recent approaches increasingly incorporate temporal representations and timestamp knowledge into the language modeling framework (Huang et al., 2024; Ren et al., 2024; Guo et al., 2025a), while others introduce numerical temporal cues and temporal reasoning mechanisms for more precise localization (Li et al., 2026b; Wu et al., 2025b; Guo et al., 2025b; Zeng et al., 2025a). Recent efforts further explore post-training strategies, including reinforcement learning, to enhance temporal grounding (Wang et al., 2026; Feng et al., 2026; Zheng et al., 2026a; Li et al., 2025; Yue et al., 2025), and dedicated localization mechanisms have been developed to provide more explicit temporal representations or fine-grained boundary prediction (Zeng et al., 2025b; Pramanick et al., 2025; Yang et al., 2025; Fan et al., 2026). Despite these advances, MLLM-based VTG methods predominantly retain the next-token prediction paradigm, relying on sequential prediction within the language-modeling pipeline. This motivates an alternative formulation that directly exploits the multimodal representations produced by the MLLM for temporal localization.

2.2 Language Model Representations for Prediction

Beyond autoregressive generation, pretrained language models can also provide contextual representations for downstream prediction. Early pretrained models such as BERT and ELECTRA demonstrate the effectiveness of contextual representations for discriminative language understanding (Devlin et al., 2019; Clark et al., 2020). More recently, LLM2Vec shows that decoder-only language models can be adapted to produce bidirectional representations for non-generative downstream tasks (BehnamGhader et al., 2024), while Qwen3-VL-Embedding extends this representation-centric perspective to multimodal language models, demonstrating that multimodal hidden representations can support retrieval and reranking beyond autoregressive generation (Li et al., 2026a). These studies suggest that language-model representations can serve as effective prediction substrates beyond text generation. Building on this perspective, YOFO-VTG directly performs temporal boundary prediction over dense, temporally-aligned multimodal hidden states, reformulating temporal grounding as a parallel prediction problem rather than sequential timestamp generation.

3 Methodology

3.1 Overall Framework

Formally, given a video V of duration D and a textual query Q, VTG aims to predict a pair of temporal boundaries (s, e) satisfying $0 \leq s \leq e \leq D$. Existing MLLM-based VTG methods typically formulate this task as next-token prediction, autoregressively generating the two boundaries as textual tokens. Although natural for language generation, this formulation is poorly matched to interval localization and incurs additional inference latency. To this end, we propose **YOFO-VTG**, which repurposes the full sequence of video-aligned hidden representations produced by an MLLM to estimate start- and end-boundary distributions in parallel within a single forward pass.

As illustrated in Figure 2, YOFO-VTG comprises a parallel boundary-prediction architecture and a two-stage training pipeline. Following the native video encoding scheme of our base model Qwen3-VL (Bai et al., 2025a), YOFO-VTG serializes temporally ordered video units, marked by timestamps and visual delimiters, together with the textual query into a unified multimodal sequence. A LoRA-adapted MLLM with bidirectional attention jointly encodes the sequence. For each video unit, we extract the hidden states at its <|vision_start|> and <|vision_end|> delimiters as temporally aligned boundary representations. The native language modeling head maps these representations to <|start|> and <|end|> logits, which are normalized across all video units to form two temporal boundary distributions. By predicting both distributions in parallel within a single forward pass, YOFO-VTG retains the probabilistic prediction mechanism of the MLLM while eliminating

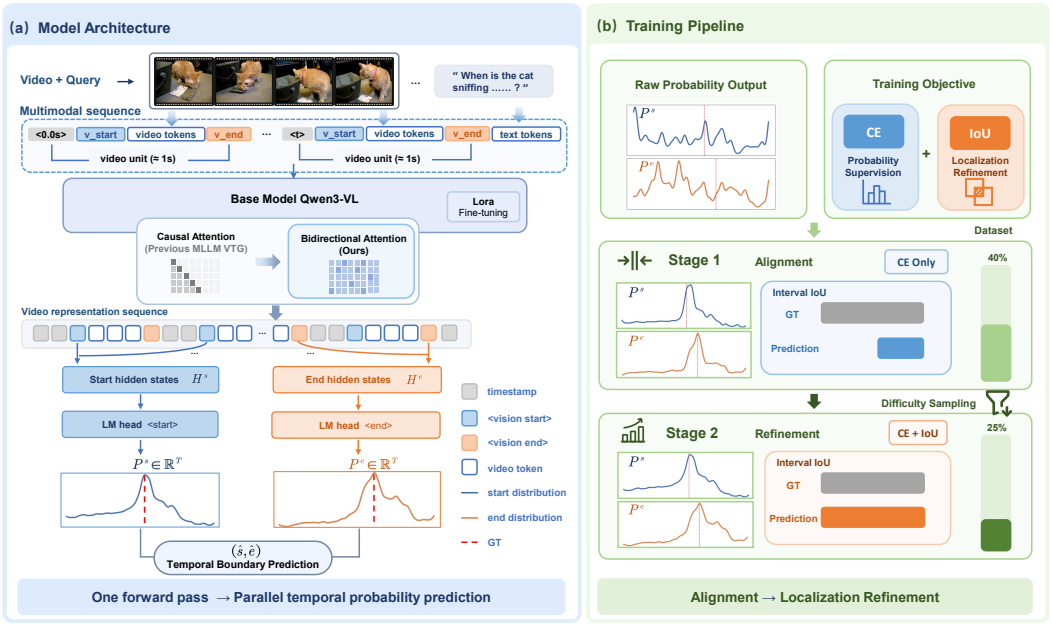


Figure 2: Overview of YOFO-VTG. Left: temporally-aligned video units and the textual query are jointly encoded with bidirectional attention, enabling direct boundary prediction from token representations associated with the special start and end tokens. Right: two-stage training first learns coarse boundaries using CE loss on 40% of the data, then performs difficulty-aware refinement on 25% of the data using joint CE and differentiable IoU supervision.

autoregressive boundary decoding, thereby substantially reducing inference latency. During training, the first stage learns coarse boundary distributions using cross-entropy supervision, while the second stage employs difficulty-aware sampling and jointly optimizes cross-entropy and differentiable IoU objectives for boundary refinement.

3.2 One Forward Pass Paradigm

Temporally-Aligned Video Representation. Our paradigm builds on the native video encoding mechanism of the underlying MLLM. Given a video V, we sample frames at 2 FPS and temporally aggregate every two consecutive frames. The video V is thereby divided into T temporal segments, yielding T temporally ordered visual units $\mathcal{U}_i$, each corresponding to a duration of 1 second. Specifically, the i-th temporal unit is serialized in the multimodal input sequence as

$$\mathcal{U}_i = [t_i \text{ seconds}, <|\texttt{vision_start}|>, \mathcal{V}_i, <|\texttt{vision_end}|>], \qquad i = 1, \ldots, T. \tag{1}$$

where t_i specifies the timestamp of the i-th unit, and $\mathcal{V}_i$ contains its visual tokens. The timestamp anchors each visual unit to its temporal position, while <|vision_start|> and <|vision_end|> delimit the associated visual-token sequence. Such interleaved timestamps are natively used in Qwen3-VL's video encoding mechanism, naturally providing temporal anchors for video events. Since these temporal anchors share the same semantics as the ground-truth boundaries in VTG, their fixed positional relationship with the delimiters naturally establishes a correspondence between temporal positions and delimiter representations. We therefore leverage temporally aligned delimiter positions for subsequent probability distribution prediction. Let $H \in \mathbb{R}^{L \times d}$ denote the output hidden sequence of the MLLM, where L is the sequence length, and d is the hidden dimension. For each temporal unit i, we extract the hidden states at its visual-start and visual-end delimiter positions, $\mathcal{I}_i^s$ and $\mathcal{I}_i^e$, respectively:

$$h_i^s = H[\mathcal{I}_i^s, :], \qquad h_i^e = H[\mathcal{I}_i^e, :], \qquad h_i^s, h_i^e \in \mathbb{R}^d. \tag{2}$$

Stacking these representations across all T temporal units yields

$$H^s = [h_1^s, \ldots, h_T^s], \qquad H^e = [h_1^e, \ldots, h_T^e], \qquad H^s, H^e \in \mathbb{R}^{T \times d}. \tag{3}$$

The resulting H^s and H^e constitute the temporally-aligned video representation obtained in a single MLLM forward pass. Notably, conventional next-token prediction primarily relies on the hidden state at the latest position for subsequent token generation, while the long video representation sequence produced by the MLLM is largely treated as an intermediate state rather than a direct prediction target. Our paradigm instead revisits this underutilized representation sequence, exploiting its rich temporal structure to provide temporally aligned representations for subsequent probability distribution prediction.

Bidirectional Parallel Boundary Prediction. We cast VTG as jointly estimating start- and end-boundary distributions over T temporal units. Accurate boundary estimation requires evaluating each candidate position within the global video and query context. However, the causal attention mechanism inherited from conventional LLMs restricts each hidden representation to preceding sequence elements, limiting its access to subsequent temporal evidence. We therefore replace the causal mask with a bidirectional attention mask, enabling information exchange across the entire multimodal sequence. The resulting contextually enriched representations, $H^s, H^e \in \mathbb{R}^{T\times d}$, are used to estimate the two boundary distributions in parallel.

MLLMs natively predict distributions over discrete tokens, while VTG can analogously be formulated as estimating distributions over temporal start and end positions. Motivated by this structural alignment, we apply the native language modeling head, LMHead, directly to the video representation sequences H^s and H^e. Specifically, for each sequence $H^r, r \in \{s, e\}$, the language modeling head maps its i-th temporal representation, h_i^r, to a logit vector over the vocabulary:

$$z_i^r = \mathrm{LMHead}(h_i^r), \qquad z_i^r \in \mathbb{R}^{|\mathcal{W}|}, \tag{4}$$

where $\mathcal{W}$ denotes the vocabulary. Since VTG requires predicting only the start and end positions of the target event, we retain the logits associated with two semantically aligned tokens, <|start|> and <|end|>. Let w_s and w_e denote their respective indices in $\mathcal{W}$. The start- and end-boundary logits over the T temporal units are computed as

$$Z^s = ([z_i^s]_{w_s})_{i=1}^T, \qquad Z^e = ([z_i^e]_{w_e})_{i=1}^T, \qquad Z^s, Z^e \in \mathbb{R}^T. \tag{5}$$

Here, $[\cdot]_{w_s}$ and $[\cdot]_{w_e}$ select the vocabulary logits associated with the <|start|> and <|end|> tokens, respectively. We then normalize the logits along the temporal dimension to obtain the corresponding boundary distributions:

$$P^s = \mathrm{Softmax}\left(\frac{Z^s}{\tau}\right), \qquad P^e = \mathrm{Softmax}\left(\frac{Z^e}{\tau}\right), \qquad P^s, P^e \in \mathbb{R}^T, \tag{6}$$

where τ is a temperature coefficient that controls the sharpness of the temporal probability distributions, set to 0.05 in all experiments. This formulation scores all candidate positions simultaneously in a single bidirectional MLLM forward pass, enabling direct temporal localization without autoregressive token-by-token decoding.

3.3 Training

Complementary Training Objectives. We optimize YOFO-VTG with LoRA using two complementary objectives: the cross-entropy (CE) loss adapted from the MLLM and a differentiable IoU loss that directly aligns the predicted probability distributions with the temporal localization objective. Given the boundary distributions P^s and P^e, the CE loss supervises the probability mass at the ground-truth temporal positions (i_s, i_e):

$$\mathcal{L}_{\mathrm{CE}} = -\log P_{i_s}^s - \log P_{i_e}^e. \tag{7}$$

This objective encourages the boundary probability distributions to concentrate their probability mass around the ground-truth positions, while inheriting the pretrained MLLM's probability prediction capability. However, CE is not explicitly aligned with the temporal localization metric. A lower $\mathcal{L}_{\mathrm{CE}}$ does not necessarily imply a higher IoU, as it only supervises the probability assigned to the ground-truth position. Fortunately, our parallel probability formulation allows the entire predicted distributions to be directly optimized with respect to IoU in a differentiable manner. Specifically, let

$\mathcal{T} = [t_1, t_2, \ldots, t_T]$ denote the temporal coordinates of the T prediction positions. We obtain differentiable temporal coordinates from the predicted distributions through their probability-weighted expectations:

$$\tilde{s} = \sum_{i=1}^{T} P_i^s t_i, \qquad \tilde{e} = \sum_{i=1}^{T} P_i^e t_i. \tag{8}$$

The soft interval $(\tilde{s}, \tilde{e})$ is differentiable with respect to the predicted probability distributions. To enforce the temporal ordering constraint $\tilde{s} \leq \tilde{e}$, we introduce a ReLU-based ordering penalty:

$$\mathcal{L}_{\text{IoU}} = 1 - \text{IoU}\left((\tilde{s}, \tilde{e}), (s, e)\right) + \text{ReLU}(\tilde{s} - \tilde{e}), \tag{9}$$

Unlike autoregressive VTG, where timestamps are generated as discrete tokens and direct IoU optimization is difficult, our parallel formulation provides continuous temporal estimates directly from the predicted distributions, making metric-aligned optimization straightforward. The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{IoU}}, \tag{10}$$

where λ controls the contribution of the IoU objective. The two objectives provide complementary supervision: CE encourages accurate probability concentration at the ground-truth temporal positions, while the IoU objective considers the resulting temporal extent and directly aligns the predictions with the localization metric.

Two-Stage Optimization. We adopt a two-stage training strategy to progressively optimize the probability distributions. In the first stage, we set $\lambda = 0$ and optimize solely with $\mathcal{L}_{\text{CE}}$, which provides stable supervision for learning accurate start and end distributions and thus establishes a reliable initialization for subsequent IoU optimization. In the second stage, we activate the IoU objective with $\lambda = 5$ and perform difficulty-aware data sampling (Wang et al., 2026; Zhang et al., 2026) based on the pre-computed IoU scores from TimeLens, assigning higher sampling probabilities to samples with lower IoU. The model is then optimized with the overall objective defined above to further refine the temporal probability distributions, particularly on challenging cases. More details of the sampling strategy are provided in the Appendix A.2.

4 Experiments

4.1 Experimental Setting

Datasets and Evaluation Metrics. We train exclusively on TimeLens-100K and conduct zero-shot evaluation on TimeLensBench (Zhang et al., 2026), which is constructed from Charades-STA (Gao et al., 2017), ActivityNet-Captions (Krishna et al., 2017), and QVHighlights (Lei et al., 2021) through systematic data cleaning and standardization. With a single inference, we obtain two candidate intervals from the highest-probability and soft boundaries in Eq. 8, and select the better-performing candidate for evaluation, which also supports practical multi-candidate retrieval without additional inference. Following prior work (Guo et al., 2025a; Huang et al., 2024; Zheng et al., 2026b), we report Recall@1 at IoU $= 0.3$, 0.5, and 0.7 (R@0.3, R@0.5, and R@0.7), along with mean IoU (mIoU). While mIoU measures overall temporal overlap, Recall@1 evaluates localization accuracy under different overlap requirements.

Implementation Details. Experiments are conducted primarily on NVIDIA RTX 4090 and RTX PRO 6000 GPUs. Following TimeLens (Zhang et al., 2026), we use dynamic video resolution and sample videos at 2 FPS, and adopt the dynamic high-resolution setting by default, including for the main results and the difficulty-aware sampling ablation. For computational efficiency, the remaining ablations use the dynamic low-resolution setting, as explicitly noted in their table captions. Detailed configurations are provided in Appendix A.1. With Qwen3-VL (Bai et al., 2025a) as the base MLLM, we adopt two-stage Q-LoRA with batch size 32 over approximately 96K training samples, freezing all parameters except the LoRA modules and language modeling head. Stage 1 is trained with CE loss at a learning rate of 1×10^{-4} and early stopped after $1{,}200$ steps, corresponding to approximately 38K training samples. Stage 2 uses about 25% difficulty-sampled data with joint CE and IoU losses at a learning rate of 5×10^{-5}. Both stages use AdamW with cosine decay and a warm-up ratio of 0.03, with LoRA rank $r = 64$ and scaling factor $\alpha = 128$.

Table 1: Zero-shot evaluation on the TimeLens-Bench.

Model	Charades-TimeLens				ActivityNet-TimeLens				QVHighlights-TimeLens			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
Other VTG methods												
TaRO (Zheng et al., 2026a)	63.5	44.5	23.6	44.8	54.0	40.0	24.8	40.0	69.0	56.7	40.8	53.8
DisTime (Zeng et al., 2025b)	59.9	38.5	21.3	42.5	41.5	28.2	17.1	31.2	61.1	48.0	31.0	45.6
EvoGround (Jung et al., 2026)	58.6	34.3	18.4	40.3	48.5	33.1	20.6	35.8	67.1	54.6	38.9	51.4
Time-R1-7B (Wang et al., 2026)	57.9	32.0	16.9	36.6	44.8	31.0	19.0	33.1	65.8	51.5	36.1	49.2
Qwen2.5-VL-7B (Bai et al., 2025b)	59.7	37.8	16.6	39.3	44.1	31.0	16.1	31.4	41.5	27.8	15.2	31.6
TimeLens-7B (Zhang et al., 2026)	70.5	55.6	28.4	48.8	62.8	51.0	32.6	46.2	74.1	62.7	43.1	56.0
8B-scale models												
Qwen3-VL-8B (Bai et al., 2025a)	69.2	53.4	27.5	48.3	62.1	51.2	34.4	46.8	74.2	64.6	49.3	59.4
TimeLens-8B (Zhang et al., 2026)	76.6	63.0	35.2	55.2	68.9	58.4	40.6	53.2	**80.2**	**71.6**	**55.5**	**65.5**
F2G (Zheng et al., 2026b)	74.2	63.1	36.3	54.9	**69.8**	**59.4**	**42.3**	**54.1**	79.6	70.4	54.5	64.1
Ours-8B	**81.4**	**67.1**	**39.7**	**59.8**	69.1	58.4	40.2	53.9	77.6	65.4	49.8	62.8
4B-scale models												
Qwen3-VL-4B (Bai et al., 2025a)	67.1	47.5	24.1	45.6	62.9	51.0	34.3	47.2	77.6	66.0	50.9	61.7
OMTG-4B (SFT) (Xu et al., 2026)	70.4	56.9	31.2	50.5	59.5	50.2	34.4	45.8	76.5	65.4	50.8	61.5
OMTG-4B (RL) (Xu et al., 2026)	72.0	58.3	32.0	50.5	64.7	54.6	37.0	46.8	**80.9**	**70.4**	**53.5**	62.2
Ours-4B	**80.5**	**65.3**	**39.6**	**58.9**	**68.8**	**57.1**	**38.3**	**52.9**	78.0	65.6	49.9	**63.2**
2B-scale models												
Qwen3-VL-2B (Bai et al., 2025a)	62.7	44.4	23.4	43.4	53.7	41.0	26.5	39.9	67.8	53.4	39.3	52.2
VideoMind-2B (Liu et al., 2025)	35.1	20.1	10.7	28.2	38.1	27.0	17.6	29.1	62.0	50.7	37.5	48.0
Marlin-2B (NemoStation, 2026)	66.3	51.7	27.1	46.5	50.5	40.5	25.3	37.9	60.7	48.5	34.9	46.8
Ours-2B	**75.8**	**62.0**	**37.5**	**56.4**	**66.4**	**54.0**	**37.2**	**51.2**	**76.5**	**64.0**	**48.9**	**62.1**

Baselines. We compare our method with a broad range of recent VTG approaches. For dedicated MLLM-based VTG methods, we include TimeLens (Zhang et al., 2026), F2G (Zheng et al., 2026b), DisTime (Zeng et al., 2025b), TaRO (Zheng et al., 2026a), Time-R1 (Wang et al., 2026), and EvoGround (Jung et al., 2026), which explore different designs for temporal representation, evidence-based grounding, and task-specific post-training. We further include general-purpose or video reasoning MLLMs, including Qwen2.5-VL (Bai et al., 2025b), Qwen3-VL (Bai et al., 2025a), VideoMind (Liu et al., 2025), and Marlin (NemoStation, 2026), as well as OMTG (Xu et al., 2026), a recent temporal grounding model developed for one-to-many localization.

4.2 Main Results

Strong Performance Across Model Scales. As shown in Table 1, our method performs strongly across all model scales. The 8B variant achieves the best results on Charades-TimeLens, with an mIoU of 59.8, and remains highly competitive on the other two benchmarks. At 4B, our model achieves the best results across all four metrics on both Charades-TimeLens and ActivityNet-TimeLens, while also obtaining the highest mIoU on QVHighlights-TimeLens. Most notably, the 2B variant achieves the best results across all four metrics on all three benchmarks, outperforming both the Qwen3-VL backbone and several larger models. These results demonstrate the strong VTG performance of our approach across a wide range of model scales.

Robust Scaling Behavior With Reduced Model Size. Beyond absolute performance, our method scales more robustly to smaller models. As shown in Figure 3, scaling Qwen3-VL from 8B to 2B reduces mIoU by 6.9 and 7.2 points on ActivityNet-TL and QVHighlights-TL, respectively, whereas our model drops by only 2.7 and 0.7 points and consistently outperforms the corresponding backbone across all three scales. This robustness suggests that our paradigm preserves stronger VTG capabilities under limited parameter budgets; together with single-forward-pass inference, it highlights a favorable accuracy–efficiency trade-off further analyzed in Sec. 4.3.

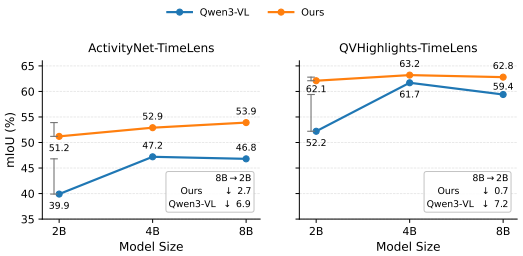


Figure 3: Scaling behavior across model sizes.

4.3 Scaling and Efficiency Analysis

We evaluate inference latency to demonstrate the efficiency advantage of our paradigm. Since existing VTG methods adopt different video preprocessing and resolution strategies, resulting in substantially different visual sequence lengths, we standardize the input setting across models and measure single-inference latency under identical conditions. Detailed evaluation settings are provided in Appendix A.6. As shown in Figure 4, our method achieves a favorable accuracy–latency trade-off on Charades-TL. Conventional autoregressive VTG methods incur substantial latency from token-by-token decoding. In contrast, our single-forward-pass paradigm directly predicts temporal boundaries without iterative decoding. This advantage is particularly evident for the 2B model, which achieves the highest mIoU and lowest latency among the compared methods, highlighting the scalability and efficiency of our one-forward design.

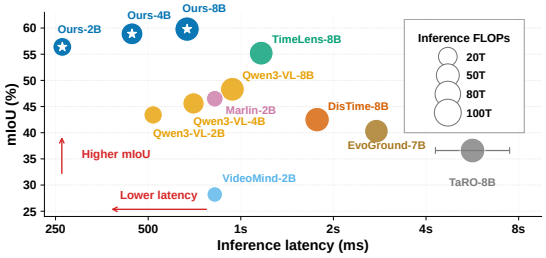


Figure 4: Accuracy–efficiency trade-off among VTG methods. The x-axis and y-axis report inference latency and mIoU, respectively, while bubble size represents inference FLOPs.

4.4 Ablation Study

Data Filtering Stabilizes SFT. As shown in Table 2, increasing training steps on the full dataset yields inconsistent gains, improving performance on Charades-TL but degrading it on QVHighlights-TL. Motivated by prior findings on difficulty-based sampling for VTG (Zhang et al., 2026) RL, we investigate whether the same principle benefits our SFT paradigm. Following these, we estimate sample difficulty from offline inference, with lower IoU indicating more challenging samples. We first train with CE and then focus subsequent training on difficult samples for joint CE and IoU optimization. Despite using less training data and fewer steps, difficulty sampling improves both R@0.5 and mIoU on Charades-TL over the full-data setting, while largely recovering the degradation on QVHighlights-TL. These results suggest that, under our SFT paradigm, selecting informative samples can be more effective than simply increasing the training budget.

Table 2: Effect of difficulty-aware sampling with Qwen3-VL-8B.

Setting	Training	Charades-TL		QVHighlights-TL	
		R@0.5	mIoU	R@0.5	mIoU
Full Data	1200	61.7	56.2	**65.8**	**63.6**
Full Data	2400	65.5 (+3.8)	58.4 (+2.2)	63.9 (-1.9)	62.4 (-1.2)
Ours	**1200+600**	**67.1** (+5.4)	**59.8** (+3.6)	65.4 (-0.4)	62.8 (-0.8)

Table 3: Main component ablation with Qwen3-VL-4B under the dynamic low-resolution setting.

Attention	Representation	Charades-TL		ActivityNet-TL		QVHighlights-TL	
		R@0.5	mIoU	R@0.5	mIoU	R@0.5	mIoU
Causal	Delimiter	53.3	49.3	40.7	39.7	49.6	48.3
Bidirectional	Video Token	53.3	49.3	47.3	44.4	58.9	56.4
Bidirectional	**Delimiter**	**64.6**	**57.7**	**51.8**	**48.8**	**62.0**	**60.4**

Bidirectional Attention and Delimiter Representations. Our single-forward-pass localization paradigm raises two questions: *(Q1) Does parallel boundary prediction require bidirectional attention?* and *(Q2) Which hidden representations are better suited to boundary prediction, video-token states or delimiter states?* For Q1, we prepend the query to the video sequence to construct a compatible causal-attention baseline, where each temporal position attends to the query and preceding, but not subsequent, video content. As shown in Table 3, causal attention reduces mIoU by 8.4, 9.1, and 12.1 points across the three benchmarks, demonstrating the importance of both preceding and subsequent temporal evidence for boundary prediction. For Q2, we compare hidden states at video-token positions with delimiter representations. Despite occupying only a small fraction of the multimodal sequence, delimiter representations consistently outperform video-token representations, improving mIoU by 8.4, 4.4, and 4.0 points across the three benchmarks. This shows that visual delimiters, although introduced as structural markers in the native MLLM sequence, provide a more effective interface for video-aligned representation extraction and temporal boundary prediction. In the Appendix A.3and A.4, we provide ablation details and a preliminary analysis of why delimiter representations are better suited to this task.

Table 4: Ablation of two-stage training and loss design under the dynamic low-resolution setting.

Model	Stage 1	Stage 2	Charades-TimeLens				QVHighlights-TimeLens			
			R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
Qwen3-VL-4B	$\mathcal{L}_{\text{CE}}$	$\mathcal{L}_{\text{CE}}$	77.4	62.5	34.7	55.2	72.4	58.9	44.1	57.8
	$\mathcal{L}_{\text{IoU}}$	$\mathcal{L}_{\text{IoU}}$	20.0	9.0	3.6	19.5	20.4	12.7	9.5	21.0
	$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{IoU}}$	$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{IoU}}$	78.5	63.7	**38.6**	57.5	75.2	**62.2**	45.9	59.9
	$\mathcal{L}_{\text{CE}}$	$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{IoU}}$	**79.0**	**64.6**	38.4	**57.7**	**75.6**	62.0	**47.2**	**60.4**

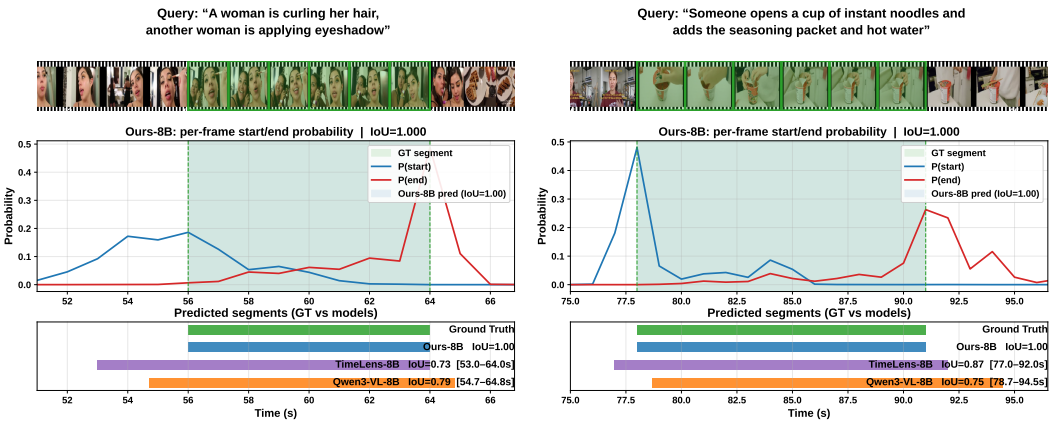


Figure 5: Qualitative visualization of temporal grounding. Each example shows the per-frame start/end probability distributions and the predicted temporal segments.

Staged Loss Optimization Improves Localization. We ablate the loss design of our two-stage training strategy using the 4B model with low-dynamic-resolution videos for both training and evaluation. As shown in Table 4, neither loss alone achieves the best performance. CE provides stable supervision for learning the MLLM's probability predictions but lacks direct guidance toward the IoU objective, while IoU alone makes training difficult to converge and leads to substantially lower performance. Joint optimization improves over either loss alone but remains inferior to our staged strategy, which first learns coarse boundary distributions with CE and then introduces IoU while retaining CE for boundary refinement. These results highlight the complementary roles of CE as the primary probability supervision and IoU as a task-specific objective for temporal boundary refinement, with staged optimization yielding more effective localization.

4.5 Visualization

Figure 5 compares our method with Qwen3-VL-8B and TimeLens-8B on two examples. In the first example (left), where the query involves two simultaneous actions, both baselines struggle to identify the queried actions and their temporal segments, whereas our model correctly grounds both. In the second example (right), both baselines produce inaccurate event boundaries and fail to capture the target event's temporal extent, while our model precisely localizes the boundaries. These results demonstrate effective semantic grounding and temporal localization through probability distribution prediction over temporal positions.

5 Conclusion

We presented YOFO-VTG, which reformulates MLLM-based Video Temporal Grounding from token-level autoregressive prediction to sequence-level parallel localization. By predicting temporal boundary probability distributions directly from the MLLM's native video representations in a single forward pass, our approach achieves competitive localization accuracy with substantially lower inference cost. These results show that the MLLM's native sequence representations can serve as direct localization signals, offering a simple and efficient alternative to MLLM-based VTG.

AI USE STATEMENT

In this work, we used generative AI tools to aid in and polish the writing of the paper. We did not use generative AI tools for research ideation, methodology or experiment design, implementation, data generation or processing, mathematical reasoning, result interpretation, or any other required research tasks. We have reviewed all AI-assisted work and take responsibility for the final content of this work.

ETHICS STATEMENT

This study focuses on investigating a new modeling paradigm for the video temporal grounding (VTG) task and does not involve human subjects, personal or sensitive information, or the generation or promotion of harmful content. All datasets used in our experiments are publicly available and widely adopted by the research community. Moreover, our LLM-based paradigm does not generate textual content; it directly predicts probability distributions of temporal boundaries. Therefore, we believe that this work does not raise additional ethical concerns and is consistent with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the datasets and evaluation metrics used in our experiments, together with detailed implementation settings in the experimental setup section 4.1. The complete training configurations are provided in Appendix A.1, the difficulty-aware sampling strategy is described in Appendix A.2, and the prompt templates are provided in Appendix A.5.

REFERENCES

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pp. 5803–5812, 2017.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv.org*, 2025b. doi: 10.48550/arXiv.2502.13923.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. URL `https://openreview.net/pdf?id=r1xMH1BtvB`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Rong Fan, Kaiyan Xiao, Minghao Zhu, Liuyi Wang, Kai Dai, and Zhao Yang. Groundvts: Visual token sampling in multimodal large language models for video temporal grounding. *arXiv preprint arXiv:2604.02093*, 2026.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5277–5285. IEEE, 2017.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3302–3310, 2025a.

Yongxin Guo, Jingyu Liu, Mingda Li, Qingbin Liu, Xi Chen, and Xiaoying Tang. Trace: Temporal grounding video llm via causal event modeling. In *International Conference on Learning Representations*, 2025b. doi: 10.48550/arXiv.2410.05643.

Ayush Gupta, Anirban Roy, Rama Chellappa, Nathaniel D Bastian, Alvaro Velasquez, and Susmit Jha. Toga: Temporally grounded open-ended video qa with weak supervision. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23593–23603. IEEE, 2025.

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

Minjoon Jung, Byoung-Tak Zhang, and Lorenzo Torresani. Evoground: Self-evolving video agents for video temporal grounding. *arXiv preprint arXiv:2605.13803*, 2026.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

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

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

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026a.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. *Advances in Neural Information Processing Systems*, 38:64426–64455, 2026b.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2782–2792. IEEE, 2023.

Ye Liu, Kevin Qinghong Lin, Chang Wen Chen, and Mike Zheng Shou. Videomind: A chain-of-lora agent for long video reasoning. *arXiv.org*, 2025. doi: 10.48550/arXiv.2503.13444.

Fangzhou Mu, Sicheng Mo, and Yin Li. Snag: Scalable and accurate video grounding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18930–18940. IEEE, 2024.

Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *Computer Vision and Pattern Recognition*, pp. 10807–10816. IEEE, 2020. doi: 10.1109/CVPR42600.2020.01082.

Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, Hao Zhang, and Wei Lu. Interventional video grounding with dual contrastive learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2764–2774. IEEE, 2021.

NemoStation. Marlin: A tiny vlm to extract structured information from videos. Hugging Face Model Repository, 2026. URL https://huggingface.co/NemoStation/Marlin-2B.

Shraman Pramanick, Effrosyni Mavroudi, Yale Song, Rama Chellappa, Lorenzo Torresani, and Triantafyllos Afouras. Enrich and detect: Video temporal grounding with multimodal llms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24297–24308. IEEE, 2025.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323. IEEE, 2024.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-r1: Post-training large vision language model for temporal video grounding. *Advances in Neural Information Processing Systems*, 38:83330–83364, 2026.

Jianlong Wu, Wei Liu, Ye Liu, Meng Liu, Liqiang Nie, Zhouchen Lin, and Chang Wen Chen. A survey on video temporal grounding with multimodal large language model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(2):1521–1541, 2025a. doi: 10.1109/TPAMI.2025.3615586.

Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. Number it: Temporal grounding videos like flipping manga. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13754–13765. IEEE, 2025b.

Shaoning Xiao, Long Chen, Jian Shao, Yueting Zhuang, and Jun Xiao. Natural language video localization with learnable moment proposals. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4008–4017, 2021.

Qi Xu, Yue Tan, Shihao Chen, Jiahao Meng, Anna Wang, Shunping Ji, Hao Fei, and Jason Li. Towards one-to-many temporal grounding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2606.06294.

Zuhao Yang, Yingchen Yu, Yunqing Zhao, Shijian Lu, and Song Bai. Timeexpert: An expert-guided video llm for video temporal grounding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24286–24296. IEEE, 2025.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25599–25609, 2026.

Feng Yue, Zhaoxing Zhang, Junming Jiao, Zhengyu Liang, Shiwen Cao, Feifei Zhang, and Rong Shen. Tempo-r0: A video-mllm for temporal video grounding through efficient temporal sensing reinforcement learning. *arXiv preprint arXiv:2507.04702*, 2025.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. Timesuite: Improving mllms for long video understanding via grounded tuning. In *International Conference on Learning Representations*, volume 2025, pp. 38057–38081, 2025a.

Yingsen Zeng, Zepeng Huang, Yujie Zhong, Chengjian Feng, Jie Hu, Lin Ma, and Yang Liu. Distime: Distribution-based time representation for video large language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21961–21971. IEEE, 2025b.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 6543–6554, 2020a.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Temporal sentence grounding in videos: A survey and future directions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10443–10465, 2023.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. Timelens: Rethinking video temporal grounding with multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10419–10429, 2026.

Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12870–12877, 2020b.

Minghang Zheng, Zihao Yin, Yi Yang, Yuxin Peng, and Yang Liu. Temporal-aware reasoning optimization for video temporal grounding. *arXiv preprint arXiv:2606.09248*, 2026a.

Zelin Zheng, Xinyan Liu, Ruixin Li, Antoni B Chan, Guorong Li, Qingming Huang, and Laiyun Qing. Foresee-to-ground: From predictive temporal perception to evidence-driven reasoning for video temporal grounding. *arXiv preprint arXiv:2605.21973*, 2026b.

Appendix

A Experimental Details

A.1 Training Configuration

All experiments are trained on the TimeLens-100K dataset, following the video preprocessing pipeline of TimeLens (Zhang et al., 2026). The main results and difficulty-sampling ablation in the paper use dynamic high-resolution video inputs. Specifically, the min and total pixel budgets are set to $64 \times 32 \times 32$ and $14{,}336 \times 32 \times 32$, corresponding to approximately 64 and 14,336 visual tokens, respectively. When the number of video frames is large, the min pixel budget is used, which may result in a total token count exceeding the nominal total budget; when the number of frames is small, the total pixel budget determines the resolution. Due to computational resource constraints, the remaining ablations in the paper and the supplementary experiments in the appendix use dynamic low-resolution video inputs, with min and total pixel budgets of $16 \times 32 \times 32$ and $3{,}584 \times 32 \times 32$, corresponding to approximately 16 and 3,584 visual tokens, respectively.

A.2 Difficulty-Aware Subset Selection

Difficulty Estimation. We construct the Stage 2 training subset using an offline difficulty signal obtained from TimeLens (Zhang et al., 2026). Specifically, TimeLens performs inference on each $(\text{video}, \text{query})$ pair in the TimeLens-100K dataset and predicts a temporal interval. We use the IoU between the TimeLens prediction and the ground-truth interval as the difficulty signal: lower IoU indicates a more challenging sample. We verified that the recorded IoU exactly matches $\mathrm{IoU}(\texttt{span}, \texttt{pred})$ for all $96{,}270$ TimeLens records. After matching these annotations to the $96{,}171$ samples in our training split, $95{,}856$ samples (99.67%) receive a valid difficulty label. Since these predictions are generated independently of YOFO-VTG, the resulting IoU scores provide a fixed offline reference for subset selection.

Difficulty-Aware Sampling. To avoid over-concentrating samples from particular video-duration ranges, we partition the training samples with valid difficulty labels into nine duration buckets, $[0, 30], [30, 60], \ldots, [210, 240], [240, \infty)$, and impose a per-bucket quota K. Within each bucket $\mathcal{B}_b$, where b indexes the duration bucket, we assign each sample i an IoU-dependent Gaussian weight:

$$w_i^{(b)} = \exp\left(-\frac{1}{2}\left(\frac{\mathrm{IoU}_i - \mu}{\sigma}\right)^2\right), \tag{11}$$

where IoU_i is the TimeLens IoU of sample i, and μ and σ control the center and spread of the Gaussian weighting function, respectively.

Table 5: Sampling configuration.

Parameter	Setting
Sampling mode	Bucketed
Duration buckets	9
Bucket width	30s + tail
K	3333
μ	0.2
σ	0.2
Inverse density	$M = 20$
RNG seed	42

Table 6: Per-bucket sampling statistics.

Duration	Available	Selected	Rate
$[0, 30]$	7,404	3,333	45.0%
$[30, 60]$	18,645	3,333	17.9%
$[60, 90]$	17,090	3,333	19.5%
$[90, 120]$	13,491	3,333	24.7%
$[120, 150]$	17,214	3,333	19.4%
$[150, 180]$	16,761	3,333	19.9%
$[180, 210]$	1,637	1,637	100%
$[210, 240]$	1,046	1,046	100%
$[240, \infty)$	2,568	2,568	100%
Total	**95,856**	**25,249**	**26.3%**

We then correct these weights using the empirical IoU density within each bucket. Specifically, the IoU range is divided into $M = 20$ bins, where $j(i)$ denotes the bin containing sample i, and $c_b^{(j)}$ denotes the number of samples in the j-th IoU bin of bucket b. The corrected weight is

$$\tilde{w}_i^{(b)} = \frac{w_i^{(b)}}{c_b^{(j(i))} + 10^{-6}}. \tag{12}$$

This inverse-density correction reduces the dominance of densely populated IoU regions, allowing relatively scarce IoU regions to retain sampling probability while preserving diversity across duration ranges. We then sample without replacement from each bucket according to the normalized corrected weights:

$$S_b \sim \mathrm{WRS}\left(\mathcal{B}_b, n_b, \hat{w}^{(b)}\right), \qquad n_b = \min(K, N_b), \tag{13}$$

where S_b is the selected subset from bucket b, $\mathcal{B}_b$ is the set of training samples in that bucket, $N_b = |\mathcal{B}_b|$ is its sample count, n_b is the number of samples selected from the bucket, and $\hat{w}^{(b)}$ denotes the normalized version of $\tilde{w}^{(b)}$. Here, WRS denotes weighted random sampling without replacement.

Configuration and Integration. For the main model, we set $K = 3333$, $\mu = 0.2$, and $\sigma = 0.2$, with inverse-density correction using $M = 20$ bins and random seed 42. The resulting subset contains $25{,}249$ samples, with a mean IoU of 0.376, compared with 0.615 for the full training pool. Difficulty-aware sampling is applied only to Stage 2, where the $25{,}249$ selected samples are used for training. Stage 1 is trained with CE loss and early stopping after $1{,}200$ steps, corresponding to approximately $38{,}000$ training samples. The selected indices and sampling order for Stage 2 are fixed by deterministic seeds to ensure reproducibility. Detailed sampling configurations and per-bucket statistics are provided in Tables 5 and 6.

A.3 More Details on Model Component Ablations

We further report the complete results of the model component ablations presented in the main paper. All models use Qwen3-VL-4B-Instruct as the backbone and share the same training data, batch size, optimizer, learning-rate schedule, two-stage training procedure, and evaluation protocol. Each ablation changes only the component under investigation. The complete results on the three TimeLens-Bench subsets, evaluated using $R@0.3$, $R@0.5$, $R@0.7$, and mIoU, are provided in Table 7.

Causal Attention Ablation. We investigate whether bidirectional attention is necessary for boundary prediction by restoring the native causal attention mask of Qwen3-VL. Because the default input layout places the video before the query, directly applying causal attention would prevent the video representations from accessing the query. We therefore move the query before the video in the causal setting:

$$\text{bidirectional}: \ [V_1, \ldots, V_N, query], \qquad \text{causal}: \ [query, V_1, \ldots, V_N]. \tag{14}$$

This preserves access to the query while retaining the causal dependency structure of the original MLLM. All other components and training settings remain unchanged.

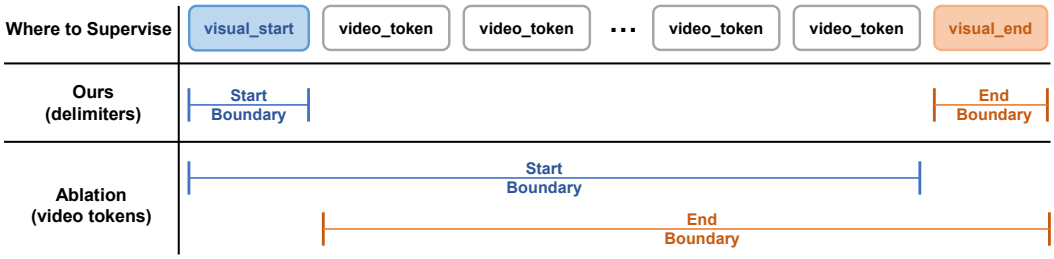


Figure 6: Comparison of boundary supervision positions. The delimiter-based baseline supervises the two delimiter representations, while the video-token variant supervises the start delimiter together with all video tokens for the start boundary, and all video tokens together with the end delimiter for the end boundary.

Table 7: Complete results of the main component ablation with Qwen3-VL-4B under the dynamic low-resolution setting.

Attention	Representation	Charades-TL				ActivityNet-TL				QVHighlights-TL			
		R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
Causal	Delimiter	70.9	53.3	26.6	49.3	53.5	40.7	24.2	39.7	68.7	49.6	29.9	48.3
Bidirectional	Video	71.5	53.3	27.4	49.3	59.7	47.3	29.3	44.4	72.1	58.9	42.6	56.4
Bidirectional	**Delimiter**	**79.0**	**64.6**	**38.4**	**57.7**	**63.0**	**51.8**	**35.1**	**48.8**	**75.6**	**62.0**	**47.2**	**60.4**

Representation Position Ablation. We investigate where to apply boundary supervision within each video unit. As illustrated in Fig. 6, our baseline supervises the two delimiter representations, with the start delimiter used for start-boundary prediction and the end delimiter used for end-boundary prediction. In contrast, the video-token variant distributes boundary supervision over the representations associated with each boundary: the start boundary is supervised using the start delimiter and all video-token representations within the unit, while the end boundary is supervised using all video-token representations and the end delimiter.

Specifically, let t denote the video unit containing the ground-truth boundary. The delimiter-based baseline applies single-point CE supervision independently to the start and end boundaries:

$$\mathcal{L}_{\text{point}} = -\log p_t^s - \log p_t^e, \tag{15}$$

where p_t^s and p_t^e denote the predicted start and end probabilities at the start- and end-delimiter positions of the ground-truth unit, respectively.

For the video-token variant, let

$$\mathcal{V}_t = \{i : \text{unit}(i) = t\} \tag{16}$$

denote all video-token positions within the ground-truth unit. We define

$$\mathcal{V}_t^s = \{\text{start}\} \cup \mathcal{V}_t, \qquad \mathcal{V}_t^e = \mathcal{V}_t \cup \{\text{end}\}, \tag{17}$$

where $\mathcal{V}_t^s$ contains the start delimiter and all video-token positions, while $\mathcal{V}_t^e$ contains all video-token positions and the end delimiter. We then uniformly distribute the corresponding boundary supervision over these positions:

$$\mathcal{L}_{\text{video-token}} = -\frac{1}{|\mathcal{V}_t^s|} \sum_{i \in \mathcal{V}_t^s} \log p_i^s - \frac{1}{|\mathcal{V}_t^e|} \sum_{i \in \mathcal{V}_t^e} \log p_i^e. \tag{18}$$

All other components and training settings remain unchanged.

A.4 Preliminary Analysis of Representation Plasticity

To better understand why delimiter representations are more effective than video-token representations for boundary prediction, we analyze their representation reorganization during fine-tuning. We compare the hidden states of the pretrained and fine-tuned Qwen3-VL-4B-Instruct models under the same inputs, and measure their similarity using linear CKA (Kornblith et al., 2019). Given

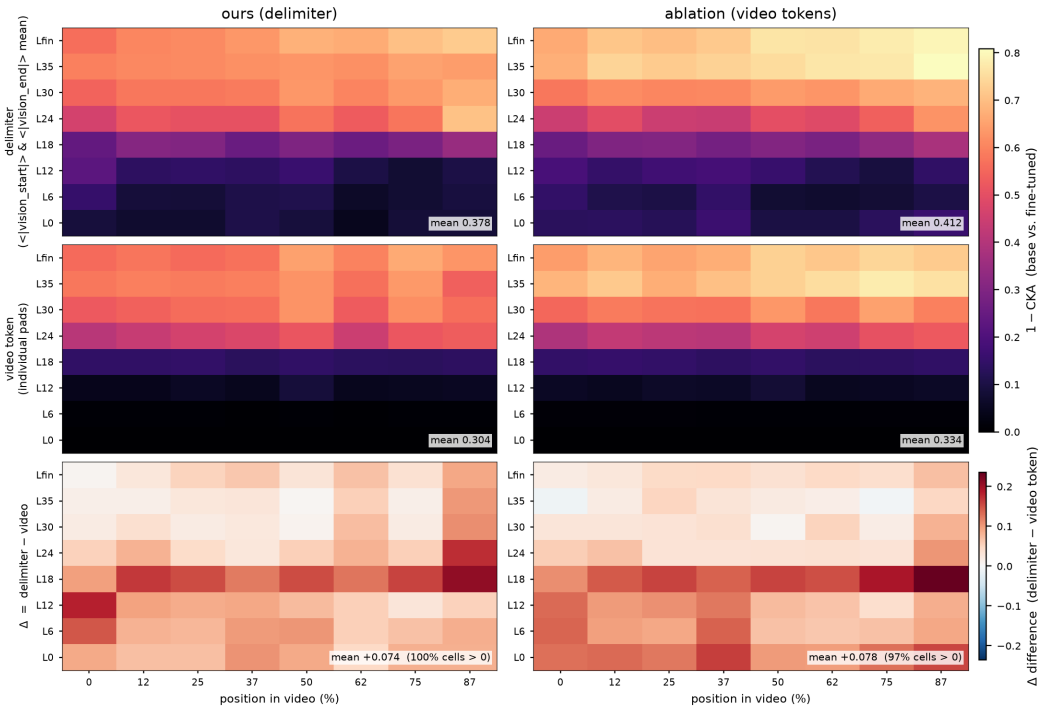


Figure 7: **Representation reorganization during fine-tuning.** We compare delimiter and video-token representations across LLM layers and normalized temporal positions using $1-\mathrm{CKA}$ between pretrained and fine-tuned hidden states.

hidden-state matrices $X, Y \in \mathbb{R}^{n\times d}$, linear CKA is defined as

$$\mathrm{CKA}(X,Y) = \frac{\|Y_c^\top X_c\|_F^2}{\|X_c^\top X_c\|_F \, \|Y_c^\top Y_c\|_F}, \tag{19}$$

where X_c and Y_c denote the centered representations. We use $1-\mathrm{CKA}$ as a measure of representation reorganization, where a larger value indicates a larger change in the internal representation geometry between the pretrained and fine-tuned models. Figure 7 compares the representation changes induced by fine-tuning for delimiter and video-token representations across LLM layers. In the left column, delimiter representations undergo substantially larger changes than video-token representations, as reflected by the consistently positive difference in the bottom row (mean $+0.074$). This is expected, since only the delimiter representations receive direct task supervision in this setting. More interestingly, the same pattern persists in the video-token supervision variant (right), where video tokens also receive direct task supervision, with a mean difference of $+0.078$ and 97% of cells showing a positive difference. This suggests that the greater reorganization of delimiter representations cannot be explained solely by the locations receiving the training loss. Instead, delimiter positions may provide a more plastic interface for task-specific adaptation, allowing their representations to be substantially reorganized to incorporate boundary-related information while video-token representations retain more of their pretrained visual content. This observation provides a possible explanation for the empirical advantage of delimiter representations, although the underlying causal mechanism remains an interesting direction for future study.

A.5 Prompt

We use the following prompt format for temporal grounding.

Table 8: Example of the input prompt for temporal grounding.

Field	Prompt
Video sequence	<0.0 seconds><|vision_start|><|video_pad|> ···<|video_pad|><|vision_end|> <1.0 seconds><|vision_start|><|video_pad|> ···<|video_pad|><|vision_end|> ⋮ <20.0 seconds><|vision_start|><|video_pad|> ···<|video_pad|><|vision_end|>
Instruction	You are a video-text retrieval assistant. Retrieve the video segment that matches the following description. Focus on actions, objects, and scenes and predict the corresponding start and end markers after the corresponding frame.
Query	Description: A person opens the refrigerator and takes out a bottle.

Table 9: Complete accuracy and efficiency results corresponding to the trade-off shown in Fig. 4. "Gen. Tok." denotes the number of generated tokens per sample.

Model	Backbone	Inference Mode	Visual Patch	LLM Seq. Len.	Gen. Tok.	Latency (ms)	FLOPs (T)	Peak Mem. (GiB)
Ours-2B	Qwen3-VL-2B	**Single forward pass**	13,888	3,802	–	**263**	**23.8**	4.7
Ours-4B	Qwen3-VL-4B	**Single forward pass**	13,888	3,802	–	**443**	**45.9**	9.3
Ours-8B	Qwen3-VL-8B	**Single forward pass**	13,888	3,802	–	**669**	**74.7**	16.3
Qwen3-VL-2B base	Qwen3-VL-2B	Generative	13,888	3,808	10–13	519	23.8	4.7
Qwen3-VL-4B base	Qwen3-VL-4B	Generative	13,888	3,808	10–11	702	46.1	9.3
Qwen3-VL-8B base	Qwen3-VL-8B	Generative	13,888	3,808	10–11	939	74.9	17.5
Marlin	Qwen3.5-2B	Generative	10,416	2,926	12–14	823	15.5	4.6
TimeLens	Qwen3-VL-8B	Generative	13,888	3,798	17–18	1,165	74.9	17.5
DisTime	InternVL2.5-8B	Generative	16 frames	4,275	9	1,770	84.4	10.9 (4 GPUs)
EvoGround	Qwen2.5-VL-7B	Generative	13,888	3,617	81–116	2,759	72.0	16.2
VideoMind	Qwen2-VL-2B	Generative	3,840	1,094	9	823	11.0	7.3
TaRO	Qwen3-VL-8B	Generative	14,880	4,322	113–212	6,173	88.4	17.6

A.6 Additional Details on Speed and Efficiency Comparison

We provide the measurement details for the speed and efficiency comparison in Table 9. All models are evaluated on the Charades split of TimeLens-Bench. For the unified video setting, we sample 31 frames, yielding approximately 13,888 visual patches and 3,802 LLM sequence tokens per sample. Latency is measured as the mean wall-clock time of a complete inference call, including visual encoding and LLM computation, after warm-up and over multiple runs. FLOPs and peak GPU memory are measured under the same inference configuration. Unless otherwise specified, all measurements use `bf16`, SDPA attention, and a single 24 GB GPU. For autoregressive methods, latency and FLOPs include the full token generation process, whereas our method performs boundary prediction in a single forward pass. DisTime uses 4-GPU model parallelism due to its memory requirements.

B More Experiments

In the appendix, we further analyze the effects of the key hyperparameters involved in our training strategy. All experiments use Qwen3-VL-4B as the backbone and adopt the dynamic low-resolution setting. The training configuration is otherwise kept consistent with the baseline.

Effect of Difficulty Sampling Hyperparameters. We first investigate the effect of the difficulty coefficient μ used to construct the Stage 2 training subset. To isolate the impact of sample difficulty, all settings use the same number of training samples and follow the same training configuration as the baseline, with only the difficulty distribution of the selected subset being varied. As shown in Table 11, the baseline setting of $\mu = 0.2$ consistently outperforms $\mu = 0.5$ across all metrics on the three benchmarks, indicating that emphasizing harder samples can benefit Stage 2 fine-tuning under a fixed data budget. This suggests that the composition of the training subset can be more important than simply increasing its size when the number of samples is fixed. Comparing $\mu = 0.2$ with the more aggressive setting of $\mu = 0.05$, the performance differences are relatively small. As shown in

Table 10: Statistics of the subsets selected with different μ settings.

μ	Subset Size	IoU Mean	IoU Std.	IoU < 0.1	IoU < 0.5	Duration Mean
0.05	25,249	0.543	0.249	4.5%	38.0%	124s
0.2	25,249	0.376	0.279	19.5%	66.3%	124s
0.5	25,249	0.309	0.293	32.0%	74.8%	124s

Table 11: Effect of the difficulty sampling parameter μ under the dynamic low-resolution setting.

μ	Charades-TL				ActivityNet-TL				QVHighlights-TL			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
0.2	**79.0**	**64.6**	**38.4**	**57.7**	**63.0**	51.8	35.1	**48.8**	**75.6**	62.0	**47.2**	**60.4**
0.05	78.0	64.0	37.5	57.0	62.9	**51.9**	**35.4**	48.8	75.5	**62.4**	47.0	60.3
0.5	78.3	64.1	37.9	57.2	62.5	50.4	33.7	47.6	73.9	60.6	46.1	59.5

Table 10, this is likely because the pool of highly difficult samples is inherently limited, resulting in similar average IoU and difficulty distributions after sampling. Therefore, further increasing the emphasis on hard samples provides only marginal additional benefit.

Effect of the Temperature Coefficient τ. We further investigate the effect of the temperature coefficient τ, which is applied to the logits before the final softmax operation. Specifically, dividing the logits by a smaller τ increases their relative differences and consequently produces a sharper probability distribution over the video representation sequence. As shown in Table 12, smaller values of τ consistently lead to better performance across the three benchmarks, with $\tau = 0.05$ achieving the strongest overall results. A possible interpretation is that temperature scaling helps bridge the distributional difference between conventional next-token prediction and our boundary probability prediction. In standard language modeling, the output distribution is learned over a large vocabulary and is not necessarily required to be highly concentrated. In contrast, boundary prediction assigns probability mass over temporal positions, where the supervision is typically concentrated around the ground-truth boundary. This may favor sharper temporal probability distributions. From this perspective, reducing τ can be viewed as explicitly encouraging a sharper output distribution, which may better match the characteristics of boundary prediction and facilitate more precise localization. We emphasize that this provides only a qualitative interpretation of the observed effect; a more rigorous analysis of the relationship between temperature scaling and the learned boundary distributions is left for future work.

Effect of Video Resolution. Due to computational constraints, we adopt dynamic low-resolution sampling for the ablation and supplementary experiments. We therefore further report the complete results under both dynamic high- and low-resolution settings across three model scales in Table 13. Across all model scales, dynamic high-resolution sampling consistently achieves better performance than low-resolution sampling, demonstrating that video resolution is also an important factor for MLLM-based VTG. The performance improvement comes at a substantial computational cost. Higher-resolution inputs produce longer video representation sequences, leading to increased training memory and computational overhead. In our experiments, even the smallest 2B model incurs substantial memory consumption from the longer sequences under dynamic high-resolution sampling. A single-model LoRA training setup cannot be accommodated on an RTX 4090 with 24GB memory, whereas an RTX PRO 6000 with 96GB memory is required to support the same configuration. These results highlight the practical trade-off between localization performance and computational cost, suggesting that the choice of video resolution should be considered together with the available computational resources and application requirements.

Table 12: Effect of the temperature coefficient τ under the dynamic low-resolution setting.

τ	Charades-TL				ActivityNet-TL				QVHighlights-TL			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
0.05	**79.0**	**64.6**	**38.4**	**57.7**	**63.0**	**51.8**	**35.1**	**48.8**	**75.6**	62.0	47.2	60.4
0.2	75.8	61.1	35.8	55.5	60.6	50.7	35.4	47.9	73.0	**63.0**	**49.3**	**60.5**
1.0	72.8	59.5	33.6	53.6	58.6	49.7	34.6	46.5	71.4	61.3	47.9	59.3

Table 13: Effect of video resolution under the same model scale.

Model	Resolution	Charades-TL				ActivityNet-TL				QVHighlights-TL			
		R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
Ours-8B	High	81.4	67.1	39.7	59.8	69.1	58.4	40.2	53.9	77.6	65.4	49.8	62.8
	Low	79.3	64.6	37.1	57.4	63.5	53.4	37.0	50.1	76.5	62.4	47.3	61.2
Ours-4B	High	80.5	65.3	39.6	58.9	68.8	57.1	38.3	52.9	78.0	65.6	49.9	63.2
	Low	79.0	64.6	38.4	57.7	63.0	51.8	35.1	48.8	75.6	62.0	47.2	60.4
Ours-2B	High	75.8	62.0	37.5	56.4	66.4	54.0	37.2	51.2	76.5	64.0	48.9	62.1
	Low	73.3	58.1	35.6	54.1	61.5	50.1	34.6	47.8	73.1	59.2	45.2	58.3