# THMAP : TEMPORAL HEATMAP FOR TEXT-TO-VIDEO INTELLIGENCE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A typical video contains more visual evidence than vision language models (VLMs) can process effectively, making query-relevant events easy to miss. Existing approaches either expand the visual context of VLMs at high training and inference cost, or train an external model to predict query-conditioned temporal regions as a preprocessing step, often requiring task-specific supervision on large training datasets. We study whether relevance is already encoded by the VLM that interprets the selected frames, and introduce THMAP. THMAP extracts a query-conditioned temporal relevance heatmap from text-to-vision token interactions that occur within a small set of attention heads grounding the user query in visual context. We decompose the outputs of these heads in a mechanistically interpretable manner to measure token-level relevance, and aggregate these signals into a temporal relevance heatmap at the desired granularity. To sharpen this heatmap, we learn a lightweight low-rank adaptation using a differentiable ranking objective and a modest amount of labeled data, requiring one to two orders of magnitude less supervision than typical existing approaches. The same heatmap scores clips for highlight detection and selects visual context for temporal grounding and video question answering. On highlight detection, THMAP improves over the previous state of the art by relative gains of $7.2\%$ in mAP and $6.9\%$ in Hit@1. For temporal grounding, it improves mIoU by upto $29.4\%$ over a strong trained selector under an eight-frame budget, while remaining competitive across two VLMs for video question answering.

## 1 INTRODUCTION

Vision language models (VLMs) can now answer questions about images and videos, reason about temporal and spatial content, and follow open-ended multimodal instructions (Bai et al., 2025; Wang et al., 2025; Li et al., 2025; Zohar et al., 2025). However, videos expose a fundamental bottleneck: they contain far more visual information than a VLM can process at useful resolution, and performance drops as relevant evidence becomes sparse in long videos (Fu et al., 2025; Wu et al., 2024). Existing VLMs often address this by expanding the visual context (Chen et al., 2025; Shu et al., 2025), but this requires costly long-context training and expensive inference. Moreover, a larger nominal window does not guarantee a larger effective context: relevant evidence can still be overlooked among distractors (Liu et al., 2024a; Hsieh et al., 2024). This is especially acute in video, where the query-relevant event may occupy only a small portion of the timeline.

These limitations have motivated query-conditioned frame selection, which allocates a VLM's visual context to only those frames that are likely to be relevant to the user's query. Training-free methods balance image–text similarity with temporal coverage (Liu et al., 2025; Tang et al., 2025; Zhang et al., 2025), whereas training-based methods learn relevance from task-specific supervision. For video highlight detection (HD) task, FlashVTG achieves state-of-the-art performance by training on the full training splits of benchmark datasets (Cao et al., 2025). For video question answering (VQA), VideoITG (Wang et al., 2026) trains a query-conditioned frame selector on substantial purpose-built supervision. Together, these methods establish active context selection as a compelling alternative to indiscriminately expanding the visual context. However, they also raise a complementary question: must relevance be derived from an external similarity model or a separately trained task-specific frame selector, or does the VLM that interprets the selected evidence frames already contain the useful relevance signal?



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.

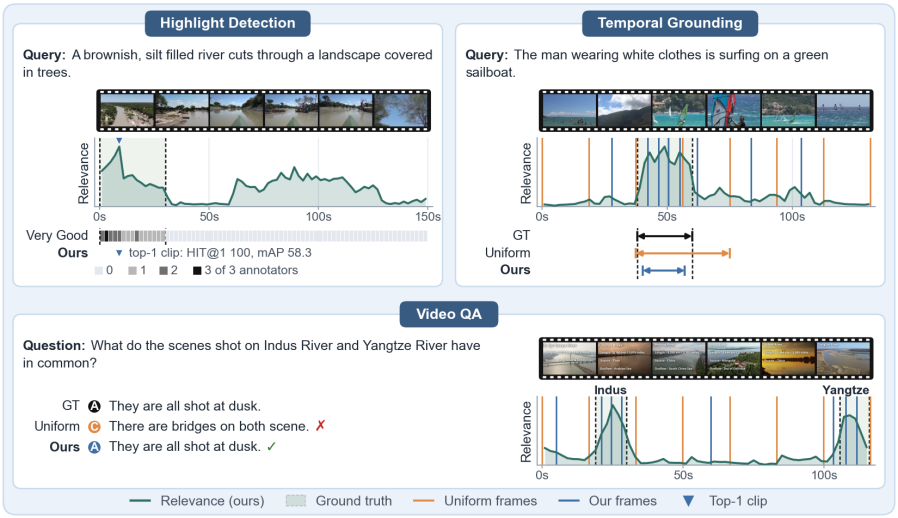



Figure 1: THMAP turns the VLM's own text-to-vision attention into a single query-conditioned relevance curve that serves highlight detection, temporal grounding, and video QA without any task-specific head.

We study this question across three tasks that are usually treated separately. *Highlight detection* ranks clips by their relevance to a query, *temporal grounding* localizes the queried event with a time interval(s), and *video question answering* answers the query from visual evidence. Prior work has connected highlight detection and moment retrieval through joint datasets and architectures (Lei et al., 2021; Liu et al., 2022; Lin et al., 2023; Xiao et al., 2024), while temporal-grounding and video-question-answering systems have largely evolved as separate model families (Zhang et al., 2026; Fu et al., 2025; Wu et al., 2024). Despite their different outputs, all three require identifying query-relevant temporal regions: a temporal relevance curve (a) scores clips directly for highlight detection, (b) identifies concentrated support for temporal grounding, and (c) supplies frame samples as compact visual context for question answering. This shared structure motivates a single reusable relevance mechanism instead of different ones for every task (Figure 1).

We observe that the required signal is already latent in a pretrained VLM's internal computation. Prior studies on LLMs show that the activations of a small subset of attention heads, mostly situated in the intermediate layers, can accurately perform in-context source attribution for text grounding tasks (Cohen-Wang et al., 2025; Todd et al., 2024; Mahapatra et al., 2025). Extrapolating these findings to VLMs, when the model jointly processes a video and a query (prefill step), query tokens retrieve visual information through text-to-vision attention. However, raw attention weights alone are an unreliable relevance score: they indicate where a query token reads from, but ignore the contributions of value vectors (Sundaresan et al., 2026). We turn these observations into a video-frame selector. Given a video and a query, we prefill the VLM once, measure the token level contribution of the vision tokens' value vectors in contextualizing the query tokens, identify query-specific grounding heads, and aggregate their contributions into a *temporal relevance heatmap* at the desired level of granularity. We utilize the same method of obtaining the temporal relevance heatmap from the VLM's internal activation signals across all three tasks.

The internal signal is informative but noisy, so we sharpen it instead of learning a separate task-specific relevance model from scratch. We freeze the base VLM and adapt its attention projections with a low-rank adapter (LoRA)(Hu et al., 2022). Because the temporal heatmap is a differentiable readout of the model's own computation, a ranking objective based on Smooth Average Precision (SAP) (Brown et al., 2020), directly trains the model to make relevant temporal regions outscore the distracting and redundant ones. With modest supervision, this lightweight adaptation refines cross-modal associations acquired during pretraining, yielding a parameter- and sample-efficient, query-conditioned, temporal heatmap scorer.

We evaluate the resulting temporal relevance heatmap on highlight detection, temporal grounding, and video question answering, spanning multiple datasets, frame budgets, grounding models, and

answering VLMs. The heatmap is a strong highlight predictor without a task-specific detection head, and its frame selections improve downstream grounding and answering models under restricted visual budgets. The gains are largest when frames are scarce, precisely where uniform sampling is most likely to miss the relevant event. Moreover, the same selector transfers across downstream VLMs, demonstrating the generality of our approach.

Our contributions are fourfold:

- We unify highlight detection, temporal grounding, and video question answering as consumers of a shared query-conditioned temporal relevance heatmap.
- We extract this heatmap in a single VLM prefill from value-projected text-to-vision interactions and query-specific grounding heads, without an external similarity model or task-specific scoring head.
- We sharpen the latent relevance signal with a $15.3$M-parameter LoRA adapter ($0.19\%$ of an $8$B backbone) trained on only around $5$K query–video pairs, yielding a parameter- and sample-efficient selector shared across tasks.
- We establish state-of-the-art QVHighlights highlight detection with $47.33$ mAP and $77.94$ Hit@1, corresponding to relative gains of $7.2\%$ and $6.9\%$ over the strongest FlashVTG baseline, respectively. At an eight-frame budget, our selections improve QVHighlights-TimeLens grounding from VideoITG's $32.3$ to $41.8$ mIoU ($p < 0.01$), and transfer competitively to video QA, reaching $66.4\%$ accuracy versus VideoITG's $66.1\%$ on VideoMME-Medium with Qwen3-VL-8B at a 32-frame budget.

*Codebase will be released upon acceptance.*

## 2 Related Work

**Vision language models and video downstream tasks.** Recent vision language models, including Qwen3-VL (Bai et al., 2025), InternVL3.5 (Wang et al., 2025), and LLaVA-OneVision (Li et al., 2025), have become general-purpose backbones for video understanding by encoding sampled video frames as visual tokens and conditioning a language model on them. This has enabled progress across query-conditioned video tasks such as highlight detection, temporal grounding, and video question answering. However, most VLM pipelines still allocate visual tokens through query-independent sampling, such as uniform or fixed-rate frame selection, so the evidence available to the model may be chosen before the query is fully used.

For highlight detection and temporal grounding, prior work mainly learns task-specific temporal models over video features or trains VLMs for localization. Moment-DETR (Lei et al., 2021) introduced an end-to-end DETR formulation that jointly handles moment retrieval and highlight scoring. CG-DETR (Moon et al., 2026) strengthens query–moment alignment by modeling clip-wise cross-modal interactions, LLMEPET (Jiang et al., 2024) injects LLM-based encoders into moment retrieval architectures, $R^2$-Tuning (Liu et al., 2024b) transfers image-pretrained representations to video for temporal grounding and FlashVTG (Cao et al., 2025) strengthens temporal prediction by feature layering and adaptive score refinement. TimeLens (Zhang et al., 2026) instead improves temporal grounding with stronger VLM training and high-quality temporal annotations. BLIP-ITM (Li et al., 2022) provides a pretrained image–text matching baseline for scoring frames or clips, but does not explicitly model long-range temporal context.

For VQA, recent work has focused both on stronger video-language backbones and on selecting better visual evidence before answering. Apollo (Zohar et al., 2025) studies VLM design for broad short- and long-video understanding, while Q-Frame (Zhang et al., 2025) identifies query-relevant frames under limited visual budgets. Frame selection methods such as BOLT (Liu et al., 2025) use CLIP-based query–frame similarity with inverse-transform sampling, AKS (Tang et al., 2025) optimizes prompt relevance and temporal coverage using pretrained frame-level features such as BLIP/CLIP, and VideoITG (Wang et al., 2026) trains an instruction-conditioned temporal grounding model for selecting frames useful to downstream VLM understanding. These works show that VQA performance depends strongly on which frames are exposed to the answerer, especially when the frame budget is small.

**Model internals, attribution, and head selection.** A growing line of work studies useful task signals from a model's internal computation rather than only from its final output. Layer by Layer (Skean et al., 2025) shows that intermediate model representations can provide strong features for downstream tasks, motivating the use of hidden computations as reusable signals. For attribution, Attr-Kit (Sundaresan et al., 2026) decomposes model behavior through value-vector contributions, while AT2 (Cohen-Wang et al., 2025) learns how to effectively use attention weights for attribution. RCStat (Mahapatra et al., 2025) introduces relative contextualization as a statistical measure of how one token group conditions another, while (Phukan et al., 2024) use hidden states for token-level attribution in QA. These works suggest that internal activations, value vectors, and selected attention heads can expose where a model obtains task-relevant evidence.

# 3 METHOD

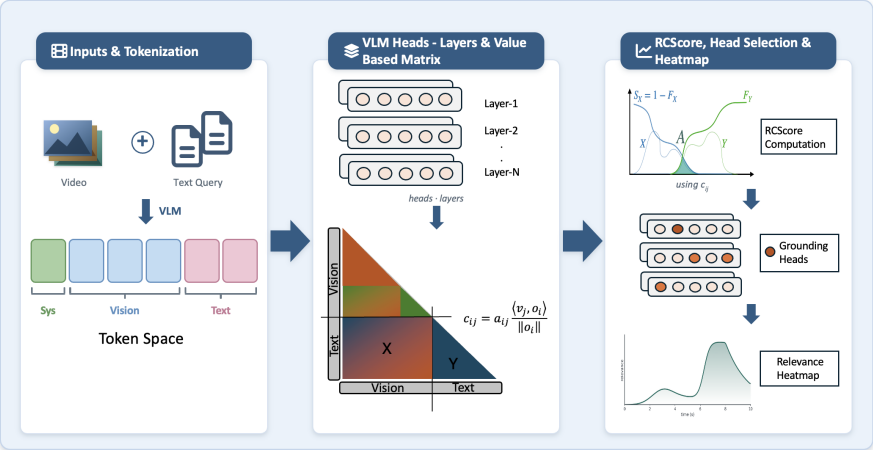



Figure 2: System overview of the proposed relevance heatmap extraction. A video and text query are tokenized into a single VLM input sequence, with visual tokens before the query tokens (§ 3.1). For each layer and head, we compute value-projection contributions $c_{ij}$ from query to visual tokens (§ 3.2). We then use the RCScore criterion to select query-conditioned grounding heads (§ 3.3) and aggregate their token-level contributions into a chunk-level temporal relevance heatmap (§ 3.4). The heatmap is used directly for highlight detection or as a sampler for downstream VLMs (§ 5).

THMAP builds on value-projection attribution from Attr-Kit (Sundaresan et al., 2026) and relative contextualization from RCStat (Mahapatra et al., 2025), but extends them from text-only source attribution to video-language relevance. Instead of explaining fixed text-token spans, we use cross-modal text-to-vision contributions to produce a trainable temporal relevance signal over video chunks.

Our goal is to extract query-conditioned visual relevance from the *internals* of a VLM, without autoregressive generation. Given a video and a text query, we prefill the VLM once and read its internal text-to-vision interactions to score each visual chunk. We first represent the video as ordered chunks and place the query after the visual tokens, allowing query tokens to attend over the full video (§ 3.1). We then compute value-projection contributions from visual tokens to query tokens (§ 3.2), select query-conditioned grounding heads (§ 3.3), and aggregate their contributions into a chunk-level relevance curve (§ 3.4). This curve is used directly for highlight detection, or converted into a fixed-budget frame selection for temporal grounding and video question answering (§ 3.5, 5).

## 3.1 SETUP AND NOTATIONS

We represent a video as an ordered sequence of $N$ visual chunks $V = (e_1, \ldots, e_N)$, where each chunk is either a single frame or a short temporally ordered clip. The chunks are mutually exclusive and cover distinct time spans; since the scoring method is agnostic to this choice, we use "chunk" throughout. We construct the VLM input as a system prompt, followed by the visual chunks in temporal order and then the text query $\mathcal{T}$. Each chunk $e_n$ is encoded as a contiguous block of vision

tokens with positions $P_n$, and $P = \bigcup_{n=1}^{N} P_n$ denotes all vision-token positions. The query occupies text-token positions $R$, and layers and heads are indexed by $\ell$ and $h$.

For a fixed head $(\ell, h)$, each token $i$ has query, key, and value vectors $q_i, k_i, v_i \in \mathbb{R}^d$. Under the causal mask, token $i$ attends only to positions $j \leq i$, with attention weights $a_{ij} = \mathrm{softmax}_{j \leq i}\big(\langle q_i, k_j \rangle / \sqrt{d}\big)$, where $\sum_{j \leq i} a_{ij} = 1$. Because the text query appears after the video, every token of the text query can attend to every vision token. This text-to-vision attention path is the source of the relevance signal used below.

### 3.2 Value Projection for Measuring Token-level Contributions

Attention weights alone indicate where a token reads from, but not how each value vector contributes to the head output. We therefore use a value-based attribution. For head $(\ell, h)$, the contextualized output at token $i$ is the attention-weighted value read-out $o_i = \sum_{j \leq i} a_{ij}\, v_j$. Following Attr-Kit (Sundaresan et al., 2026), we measure the contribution of token $j$ by projecting its value vector onto the output direction $\hat{o}_i = o_i / \|o_i\|$ and weighting this projection by attention:

$$c_{ij} \;=\; a_{ij} \left\langle v_j,\, \hat{o}_i \right\rangle \;=\; a_{ij}\, \frac{\langle v_j,\, o_i \rangle}{\|o_i\|}. \tag{1}$$

The resulting signed contributions exactly decompose the output magnitude, $\sum_{j \leq i} c_{ij} = \|o_i\|$. Equivalently, the read-out can be expressed along its output direction as $o_i = \hat{o}_i \sum_{j \leq i} c_{ij}$, so value projection maps the heterogeneous value vectors $\{v_j\}$ into scalar token-level contributions measured on a common direction. Thus, $c_{ij}$ measures the share of query token $i$'s head output attributable to token $j$.

### 3.3 Query-Conditioned Head Selection

Not every attention head provides useful grounding. Following RCStat (Mahapatra et al., 2025), we observe that many heads contextualize the query mostly through nearby text tokens, while only a small subset routes the query through the video. We therefore select grounding heads separately for each video–text pair. For a head $(\ell, h)$, we compare its text-to-vision value projections with its query-to-text value projections by forming the empirical variables $X \sim \{c_{ij} : i \in R, j \in P\}$ and $Y \sim \{c_{ij} : i \in R, j \in R, j \leq i\}$. Here, $X$ measures how much the text query representation reads from visual tokens, while $Y$ measures how much it reads from query-side text.

Following RCStat (Mahapatra et al., 2025), we score a head by the relative contextualization of vision over text, $\mathbb{E}[\max(X - Y, 0)]$: heads receive high scores when query-to-vision contributions tend to exceed query-to-text contributions. In practice, we compute the CDF-based RCScore estimator described in Appendix § A. We rank all layer–head pairs by this score and keep the top $K$ heads as the query-conditioned grounding-head set $\mathcal{H}$. This set is recomputed for every video–text pair, making head selection example-specific rather than fixed across the dataset.

### 3.4 Chunk-Level Relevance Scoring

Given the selected grounding heads, we convert token-level value projections into one relevance score per visual chunk. For each head $(\ell, h) \in \mathcal{H}$ and chunk $e_n$, we average the value projections from all text query tokens ($R$) to all vision tokens belonging to that chunk ($P_n$):

$$s_n^{(\ell,h)} = \frac{1}{|R|\,|P_n|} \sum_{i \in R} \sum_{j \in P_n} c_{ij}. \tag{2}$$

This produces a head-specific relevance curve over the video. Because heads can differ in scale, we min–max normalize each head's scores across chunks, obtaining $\tilde{s}_n^{(\ell,h)} \in [0, 1]$, and then average across the selected grounding-head set $\mathcal{H}$ as $s_n = \frac{1}{|\mathcal{H}|} \sum_{(\ell,h) \in \mathcal{H}} \tilde{s}_n^{(\ell,h)}$. The final output is the relevance curve $\{s_n\}_{n=1}^{N}$, where higher values indicate chunks that contribute more strongly to the query representation through the selected grounding heads.

### 3.5 FRAME SAMPLING

Some downstream tasks consume a fixed budget of $k$ frames rather than a dense relevance curve. For these tasks, we apply a sampling function $\phi : (\{s_n\}, k) \mapsto \mathcal{S}$, where $\mathcal{S}$ is a set of $k$ selected frame locations on the video timeline. Scoring tasks, such as highlight detection, bypass this step and use $\{s_n\}$ directly.

Our default sampler is inverse-transform sampling over the temporal relevance curve. We treat the normalized scores as a density over chunk times, with CDF $F(t) = \sum_{n:\, t_n \le t} s_n \big/ \sum_n s_n$, and choose chunks at evenly spaced quantiles $u_1, \ldots, u_k$:

$$\phi_{\text{cdf}}(\{s_n\}, k) = \left\{ F^{-1}(u_m) \right\}_{m=1}^{k}. \tag{3}$$

This concentrates selections in high-relevance regions while preserving temporal coverage. Since the heatmap defines relevance over the video timeline, CDF sampling can select frames between the initially sampled chunks rather than only returning a subset of the chunk set. The selected frames are then passed to the downstream task model.

## 4 TRAINING LOW-RANK ADAPTER

**LoRA adaptation.** The relevance heatmap in § 3 is computed from the VLM's internal text-to-vision value projections. To sharpen this signal, we train a lightweight LoRA adapter while keeping the base VLM, value-projection readout, grounding-head selection procedure, and chunk-level scoring function fixed. LoRA is inserted into the attention projections $W_q, W_k, W_v, W_o$ of each layer, and only the LoRA parameters $\{A, B\}$ are optimized. Thus, training changes the projections used to compute $c_{ij}$, but does not introduce a separate scoring head.

**Supervision and gradients.** Each training example consists of a video $V$, a text query $\mathcal{T}$, and a relevance label $\mathrm{RL}$ with one label per chunk (Appendix § B.1). The labels may be binary or graded, and supervise the relative ordering of chunks rather than calibrated absolute scores. Given $(V, \mathcal{T})$, we prefill the VLM and compute chunk scores $\{s_i\}_{i=1}^{N}$ (§ 3). The grounding-head set $\mathcal{H}$ is selected by top-$K$ RCScore ranking (§ 3.3) and detached during optimization, so gradients flow through the selected heads' value projections and chunk scores, but not through the discrete head-selection step.

**Objective.** Because the heatmap is used as a ranked relevance signal, we optimize a ranking loss. For one training example, let $\hat{s}_i$ be the score $s_i$ standardized within the video, and let $\mathrm{RL}_i \in \mathcal{L}$ be the relevance label of chunk $i$; for graded labels, $\mathcal{L} = \{0, \ldots, 4\}$ . We use Smooth-AP (Brown et al., 2020), replacing hard rank comparisons with soft pairwise comparisons $G_{ij} = \sigma_\tau(\hat{s}_j - \hat{s}_i)$, which indicates whether chunk $j$ is ranked above chunk $i$. For graded labels, we evaluate ranking at valid thresholds $\mathcal{M} = \{m \in \{2, 3, 4\} : 1 \le |\mathcal{I}_m| < N\}$, where $\mathcal{I}_m = \{i : \mathrm{RL}_i \ge m\}$ is the set of chunks relevant at threshold $m$. The soft rank of chunk $i$ among all chunks is $r_i = 1 + \sum_{j \ne i} G_{ij}$, and its soft rank among relevant chunks is $r^+_{i,m} = 1 + \sum_{j \in \mathcal{I}_m, j \ne i} G_{ij}$.

The Smooth-AP loss is

$$\widehat{\mathrm{AP}}_m = \frac{1}{|\mathcal{I}_m|} \sum_{i \in \mathcal{I}_m} \frac{r^+_{i,m}}{r_i}, \qquad \mathcal{L}_{\mathrm{SAP}} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \left(1 - \widehat{\mathrm{AP}}_m\right). \tag{4}$$

Smooth-AP encourages relevant chunks to appear early in the ranking, but it can still concentrate the heatmap on only a few chunks from a long relevant interval. We therefore add a soft coverage loss. For each threshold $m$, let $\mu^-_m$ be the mean standardized score of irrelevant chunks. The coverage term rewards relevant chunks whose scores exceed this negative baseline:

$$\rho_m = \frac{1}{|\mathcal{I}_m|} \sum_{i \in \mathcal{I}_m} \sigma_\tau(\hat{s}_i - \mu^-_m), \qquad \mathcal{L}_{\mathrm{cov}} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} (1 - \rho_m). \tag{5}$$

The final objective averages the per-example loss over the training set,

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{(V, \mathcal{T}, \mathrm{RL}) \in \mathcal{D}} \left(\mathcal{L}_{\mathrm{SAP}} + \lambda_{\mathrm{cov}} \mathcal{L}_{\mathrm{cov}}\right), \tag{6}$$

and is optimized only over the LoRA parameters $\{A, B\}$. Concrete temperatures, loss weights, data mixtures, and optimization settings are given in § 6.1. More details in Appendix § B.2.

# 5 VIDEO DOWNSTREAM TASKS

For each video–query pair, our method produces a chunk-level relevance curve $\{s_n\}_{n=1}^{N}$ over the video. We evaluate it on three query-conditioned video tasks (Figure 1). In highlight detection (HD), the curve is used directly: chunks are scored and ranked by saliency. In temporal grounding (TG) and video question answering (VQA), the curve is used as a frame *sampler*: given a budget $k$, it selects $k$ frames from high-relevance regions while maintaining temporal coverage. For TG, selected frames are passed to a span-predictor VLM, or *grounder*, which outputs the queried event's start–end interval. For VQA, selected frames are passed to an answering VLM, or *answerer*, which predicts the answer. Thus, HD evaluates the heatmap as a standalone relevance signal, while TG and VQA evaluate it as a front-end sampler for downstream VLMs. Task-specific datasets, metrics, and baselines are given in § 6.2, § 6.3, and § 6.4; prompts are in Appendix §D.

# 6 EXPERIMENTS

We evaluate whether the proposed relevance curve provides useful query-conditioned visual evidence across the three downstream tasks introduced in § 5. The experiments are organized as follows. We first describe the common training recipe used to obtain the heatmap extractor (§ 6.1). We then evaluate the heatmap directly on HD (§ 6.2), where no downstream VLM is required. Finally, we evaluate it as a frame sampler for TG and VQA (§ 6.3 and 6.4), where sampled frames are passed to a grounder or answerer under a fixed frame budget.

## 6.1 COMMON TRAINING RECIPE

We train the heatmap extractor in two stages using the objective from § 4. Stage 1 trains the LoRA adapter on the HD/TG mixture from QVHighlights-train and Charades-train, totaling $3{,}973$ query–video pairs, and this checkpoint is used for highlight detection and temporal grounding. Stage 2 warm-starts from Stage 1 and fine-tunes on $1{,}340$ VQA pairs from a subset of VideoITG-40K, and this checkpoint is used for VQA. All training is performed at the 2 s chunk level, with labels supervising the relative ranking of chunks within each video. We ensure that there is no overlap between text queries and videos used for training with evaluation data. We keep $K = 5$ grounding heads and set $\lambda_{\text{cov}} = 0.4$ across experiments; full details are in Appendix §B. We also train Stage 1 adapters with roughly 1K, 2K, 3K, and 4K HD/TG pairs, showing strong performance with modest supervision and steady gains with more data (Table 5).

## 6.2 HIGHLIGHT DETECTION

**Datasets.** We evaluate on QVHighlights (Lei et al., 2021), a benchmark for query-based highlight detection. We use its graded 0–4 saliency annotations and report results on full val and test splits.

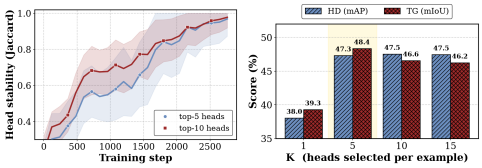
**Metrics.** Following the official evaluator (Lei et al., 2021), we report mean Average Precision (mAP) and Hit@1 at the "$\geq$Very-Good" threshold. mAP evaluates the full ranked list of clips, while Hit@1 measures whether the top-ranked clip is genuinely salient.

**Baselines.** We compare against three groups of baselines. The first group includes task-specific temporal models built on pre-extracted video features: Moment-DETR (Lei et al., 2021), UniVTG (Lin et al., 2023), CG-DETR (Moon et al., 2026), LLMEPET (Jiang et al., 2024), $R^2$-Tuning (Liu et al., 2024b), and FlashVTG (Cao et al., 2025). The second group uses pretrained image–text matching: BLIP-ITM (Li et al., 2022) scores frames or clips with respect to the query. The third group is VLM-based temporal relevance: VideoITG (Wang et al., 2026) uses an 8B instruction-tuned VLM to select query-relevant frames. Further details are in Appendix § C.1.

**Results.** Table 1 shows that our method achieves the best performance on QVHighlights. Compared with the strongest FlashVTG baseline using InternVideo2 6B features, our method achieves relative improvement of $7.2\%$ in validation mAP and $6.9\%$ in Hit@1. BLIP-ITM is limited by frame-level image–text matching without long-range temporal context, while FlashVTG relies on pre-extracted video features and a task-specific prediction head. These results indicate that internal text-to-vision signals provide a strong relevance curve for highlight detection.

Table 1: Highlight detection on QVHighlights (mAP / Hit@1 at ≥Very-Good).

| Method | Backbone / feat. | test | | val | |
|---|---|---|---|---|---|
| | | mAP | Hit@1 | mAP | Hit@1 |
| VideoITG | VideoITG-8B | 36.06 | 51.04 | 35.58 | 49.29 |
| Moment-DETR | CLIP+SF | 35.69 | 55.60 | 35.65 | 55.55 |
| UniVTG | CLIP+SF | 38.20 | 60.96 | 38.83 | 61.81 |
| $R^2$-Tuning | CLIP | 40.75 | 64.20 | 39.45 | 64.13 |
| LLMEPET | CLIP+SF | 40.33 | 65.69 | 40.52 | 65.03 |
| CG-DETR | CLIP+SF | 40.33 | 66.21 | 40.79 | 66.71 |
| FlashVTG | CLIP+SF | 41.07 | 66.15 | 41.39 | 67.61 |
| BLIP-ITM | BLIP (446M) | 42.08 | 69.91 | 41.6 | 69.7 |
| FlashVTG | InternVideo2 | 44.09 | 71.01 | 44.15 | 72.90 |
| **Ours** | Qwen3-VL-8B | **47.16** | **77.69** | **47.33** | **77.94** |




(a) Head convergence. (b) Effect of $K$.

Figure 3: Analysis of query-conditioned head selection. Selected heads stabilize during LoRA training, and $K$=5 gives the best tradeoff across HD and TG. More details for the experiments are given in Appendix § F.1.

Table 2: Temporal grounding results (mIoU / R@0.5, %) on Charades-TimeLens, QVHighlights-TimeLens, and ActivityNet-TimeLens at $k = \{8, 16\}$. **Bold** and underline denote best and second-best per column; our method is shaded. Significance is tested against the second-best method on common examples using a paired $t$-test for mIoU and McNemar's test for R@0.5. ($^{**}p < 0.01$, $^{*}p < 0.05$).

| Grounder | Sampler | Charades-TimeLens | | | | QVHighlights-TimeLens | | | | ActivityNet-TimeLens | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $k=8$ | | $k=16$ | | $k=8$ | | $k=16$ | | $k=8$ | | $k=16$ | |
| | | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 |
| TimeLens | Uniform | 26.1 | 18.1 | 36.5 | 31.5 | 25.2 | 20.8 | 33.7 | 31.6 | 20.0 | 17.8 | 24.9 | 22.0 |
| | BOLT | 29.4 | 28.4 | 38.0 | 38.1 | 33.0 | 33.7 | 42.7 | 44.7 | 20.4 | 20.5 | 28.4 | 27.9 |
| | AKS | 25.6 | 19.8 | 35.8 | 33.8 | 32.6 | 32.4 | 36.0 | 35.0 | 21.8 | 20.7 | 25.3 | 22.2 |
| | VideoITG | 32.7 | 26.8 | 42.2 | 39.3 | 32.3 | 27.0 | 42.7 | 41.5 | 26.9 | 24.4 | 34.6 | 32.8 |
| | **Ours** | **39.0**$^{**}$ | **38.3**$^{**}$ | **45.5**$^{**}$ | **46.4**$^{**}$ | **41.8**$^{**}$ | **45.6**$^{**}$ | **50.8**$^{**}$ | **55.9**$^{**}$ | **28.1** | **27.2** | **37.5**$^{*}$ | **39.8**$^{**}$ |
| Qwen3-VL | Uniform | 31.6 | 28.4 | 37.5 | 35.4 | 29.1 | 27.4 | 39.4 | 41.5 | 21.2 | 18.0 | 27.3 | 24.7 |
| | BOLT | 32.1 | 34.0 | 37.5 | 41.0 | 41.4 | 46.4 | 51.7 | 58.0 | 25.1 | 26.2 | 33.6 | 33.7 |
| | AKS | 32.0 | 31.7 | 40.1 | 44.0 | 38.3 | 41.0 | 41.5 | 44.3 | 27.7 | 26.6 | 30.7 | 28.9 |
| | VideoITG | 35.2 | **35.7** | **42.1**$^{*}$ | **44.7** | 34.8 | 32.2 | 47.0 | 48.9 | 29.9 | 30.0 | 38.3 | **41.0** |
| | **Ours** | **36.0** | 35.4 | 40.2 | 40.9 | **47.8**$^{**}$ | **53.3**$^{**}$ | **54.7**$^{**}$ | **60.4**$^{*}$ | **32.9**$^{*}$ | **33.8** | **39.0** | 39.0 |

### 6.3 Temporal Grounding

**Datasets.** We evaluate on Charades-TimeLens, QVHighlights-TimeLens, and ActivityNet-TimeLens using the TimeLens-cleaned annotations (Zhang et al., 2026). For Charades-TimeLens and ActivityNet-TimeLens, we use 1,000 randomly sampled query–video pairs each; for QVHighlights-TimeLens, we use the full validation split of 1,541 pairs.

**Metrics.** Following (Zhang et al., 2026), we report mIoU (mean temporal IoU between predicted and ground-truth intervals) and Recall@0.5 (fraction of examples with IoU $> 0.5$).

**Baselines and grounders.** We compare against four frame samplers under budgets $k = 8$ and $k = 16$. Uniform samples frames at fixed temporal intervals; BOLT (Liu et al., 2025) uses CLIP-based query–frame similarity with inverse-transform sampling; AKS (Tang et al., 2025) selects keyframes by optimizing prompt relevance and temporal coverage using pretrained BLIP features; and VideoITG (Wang et al., 2026) is an 8B instruction-conditioned frame selector trained with large-scale temporal grounding supervision. For each sampler, the selected frames are passed to the same span-predicting VLM grounder, evaluated with TimeLens-8B (Zhang et al., 2026) and Qwen3-VL-8B (Bai et al., 2025) (details in Appendix § C.2).

**Results.** Table 2 shows that our sampler improves temporal grounding in most settings across datasets, budgets, and grounders. BOLT and AKS rely on external frame-level relevance or keyframe heuristics, which can miss query-specific evidence; VideoITG is trained with about 500K data points, but its frame-selection supervision is designed for downstream VQA task and does not directly optimize TimeLens-style span prediction. In contrast, our adapter uses only about 4K training pairs while reading relevance from the selector VLM's internal text-to-vision signal.

Table 3: Effect of different frame samplers on video question answering (accuracy %). VideoMME short/medium/long and LongVideoBench at $k = \{16, 32\}$. Every sampler's frames are answered by the same VLM with the identical options-conditioned prompt; only the frame selection differs. **Bold** = best, underline = second-best per column; our method is shaded. Stars mark a paired McNemar test of best vs. second-best on common examples ($^{**}\ p < 0.01$, $^{*}\ p < 0.05$).

| Answerer | Sampler | VideoMME-Short | | VideoMME-Medium | | VideoMME-Long | | LongVideoBench | |
|---|---|---|---|---|---|---|---|---|---|
| | | $k$=16 | $k$=32 | $k$=16 | $k$=32 | $k$=16 | $k$=32 | $k$=16 | $k$=32 |
| Qwen3-VL-8B | Uniform | 69.1 | 75.2 | 55.7 | 58.8 | 52.5 | 54.6 | 54.9 | 58.3 |
| | BOLT | 73.3 | 76.7 | 58.4 | 63.5 | 52.4 | 55.6 | 56.7 | 60.7 |
| | AKS | 70.8 | 76.1 | 58.3 | 61.2 | 52.5 | 55.2 | 57.5 | 58.0 |
| | VideoITG | **76.7** | 79.1 | **64.3**$^{**}$ | 66.1 | **56.0** | **58.7** | **62.6** | **64.6** |
| | **Ours** | 75.1 | **79.8** | 60.2 | **66.4** | 53.2 | 56.2 | 62.4 | 63.9 |
| InternVL3.5-8B | Uniform | 74.1 | 76.5 | 59.4 | 61.8 | 51.5 | 52.7 | 59.5 | 60.3 |
| | BOLT | 76.8 | 77.1 | 63.2 | 66.1 | 51.2 | 54.3 | 58.5 | 61.6 |
| | AKS | 74.0 | 76.8 | 59.3 | 61.5 | 52.8 | 55.4 | 59.4 | 60.1 |
| | VideoITG | **77.0** | 77.6 | 62.0 | 63.9 | **54.5** | **58.2** | **65.4** | 65.3 |
| | **Ours** | 77.0 | **77.9** | **63.8** | **66.6** | 51.2 | 55.7 | 63.2 | **65.9** |

### 6.4 Video Question Answering

**Datasets.** We evaluate on VideoMME (Fu et al., 2025) and LongVideoBench (Wu et al., 2024), two multiple-choice video QA benchmarks. For VideoMME, we report results separately on the short, medium, and long duration splits using the full dataset under each category. For LongVideoBench, we evaluate on a 1,000-question subset.

**Metrics.** We report multiple-choice accuracy, defined as the fraction of questions for which the predicted option letter matches the ground-truth answer.

**Baselines and answerers.** We compare our sampler against Uniform sampling, BOLT (Liu et al., 2025), AKS (Tang et al., 2025), and VideoITG (Wang et al., 2026) at both frame budgets. All samplers within a comparison block use the same answering VLM and the same prompt; only the selected frames differ. We evaluate with two answering VLMs, Qwen3-VL-8B (Bai et al., 2025) and InternVL3.5-8B (Wang et al., 2025), to test whether the effect transfers across different answerers.

**Results.** Table 3 shows that our sampler is competitive in most settings across answerers and budgets, despite being trained on only about 5K examples. BOLT and AKS rely on external frame-level similarity or keyframe heuristics, which can miss answer-specific evidence when reasoning requires multiple dispersed moments. VideoITG benefits from much larger supervision, about 500K temporal grounding annotations, and is stronger on VideoMME-Long; our smaller training mixture contains fewer long videos, which likely limits long-range QA. Overall, the results show that the proposed heatmap selects useful visual evidence for VQA while requiring substantially less task supervision.

## 7 Conclusion

We presented a unified view of highlight detection, temporal grounding, and video question answering through a shared query-conditioned temporal relevance heatmap. By decomposing token interactions within a pretrained VLM, and identifying the attention heads that ground the user query in the visual context, our method extracts this latent relevance signal. We sharpen the heatmap further with lightweight and sample-efficient LoRA adaptation. Across the three tasks, the resulting heatmap supports both direct relevance prediction and the selection of compact visual evidence for downstream models. A current limitation is that our method does not process all of the content available in a video: it operates only on the visual modality and therefore cannot exploit complementary information from speech, audio, or subtitles. However, because our method is defined over token interactions rather than vision-specific features, its underlying formulation is general enough to extend to other modalities. Future work can investigate multimodal relevance heatmaps that jointly identify and select evidence across visual, auditory, and textual streams.

## AI USE STATEMENT

We used AI-based writing assistants to help with writing, organization, grammar, and phrasing. They were not used to generate research ideas, experimental results, or scientific claims, and all content was reviewed and verified by the authors.

## ETHICS STATEMENT

We have reviewed and comply with the conference Code of Ethics. To the best of our knowledge, this manuscript is original, and all related work is properly cited. The authors take responsibility for the content of the paper. Large vision language models are used as an integral and disclosed component of the proposed method. All experiments use publicly available academic benchmarks for non-commercial research; we collect no new human-subjects data and perform no person identification or tracking.

## REPRODUCIBILITY STATEMENT

**Code and Assets.** We use publicly available models and datasets, listed below; source code will be released upon acceptance.

- **Models:**
  - Qwen3-VL-8B-Instruct — https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct
  - TimeLens-8B — https://huggingface.co/TencentARC/TimeLens-8B
  - InternVL3.5-8B — https://huggingface.co/OpenGVLab/InternVL3_5-8B-HF
  - Nemotron-Nano-12B-v2-VL — https://huggingface.co/nvidia/NVIDIA-Nemotron-Nano-12B-v2-VL-BF16
  - CLIP ViT-L/14 — https://huggingface.co/openai/clip-vit-large-patch14
  - VideoITG-8B — https://huggingface.co/nvidia/VideoITG-8B
- **Datasets:**
  - QVHighlights — https://github.com/jayleicn/moment_detr
  - Charades-STA — https://github.com/jiyanggao/TALL
  - ActivityNet Captions — https://cs.stanford.edu/people/ranjaykrishna/densevid/
  - Video-MME — https://huggingface.co/datasets/lmms-eval/Video-MME
  - LongVideoBench — https://huggingface.co/datasets/longvideobench/LongVideoBench
  - TimeLens-Bench (Charades-TimeLens, QVHighlights-TimeLens, ActivityNet-TimeLens) — https://huggingface.co/datasets/TencentARC/TimeLens-Bench

## REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *European conference on computer vision*, pp. 677–694. Springer, 2020.

Zhuo Cao, Bingqing Zhang, Heming Du, Xin Yu, Xue Li, and Sen Wang. Flashvtg: Feature layering and adaptive score handling network for video temporal grounding. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 9226–9236. IEEE, 2025.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, volume 2025, pp. 18227–18246, 2025.

Benjamin Cohen-Wang, Yung-Sung Chuang, and Aleksander Madry. Learning to attribute with attention. *arXiv preprint arXiv:2504.13752*, 2025.

Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Guo Chen, et al. Nvidia nemotron nano v2 vl. *arXiv preprint arXiv:2511.03929*, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yiyang Jiang, Wengyu Zhang, Xulu Zhang, Xiao-Yong Wei, Chang Wen Chen, and Qing Li. Prior knowledge integration via llm encoding and pseudo event regulation for video moment retrieval. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7249–7258, 2024.

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.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

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.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024a.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3318–3327. IEEE, 2025.

Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3042–3051, 2022.

Ye Liu, Jixuan He, Wanhua Li, Junsik Kim, Donglai Wei, Hanspeter Pfister, and Chang Wen Chen. R 2-tuning: Efficient image-to-video transfer learning for video temporal grounding. In *European conference on computer vision*, pp. 421–438. Springer, 2024b.

Debabrata Mahapatra, Shubham Agarwal, Apoorv Saxena, and Subrata Mitra. Rcstat: A statistical framework for using relative contextualization in transformers. *arXiv preprint arXiv:2506.19549*, 2025.

WonJun Moon, Sangeek Hyun, Subeen Lee, and Jae-Pil Heo. Correlation-guided calibration of query dependency for video temporal grounding. *Pattern Recognition*, 174:112984, 2026.

Anirudh Phukan, Shwetha Somasundaram, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. Peering into the mind of language models: An approach for attribution in contextual question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11481–11495, 2024.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26160–26169. IEEE, 2025.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

Sai Sundaresan, Archit Gupta, and Debabrata Mahapatra. Attr-kit: An efficient toolkit for no-decode source attribution. In *Third Conference on Language Modeling*, 2026. URL `https://openreview.net/forum?id=WzmyBk85GP`.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In *International conference on learning representations*, volume 2024, pp. 17282–17333, 2024.

Shihao Wang, Guo Chen, De-an Huang, Zhiqi Li, Minghan Li, Guilin Liu, Jan Kautz, Jose M Alvarez, Lei Zhang, and Zhiding Yu. Videoitg: Multimodal video understanding with instructed temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24640–24650, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Yicheng Xiao, Zhuoyan Luo, Yong Liu, Yue Ma, Hengwei Bian, Yatai Ji, Yujiu Yang, and Xiu Li. Bridging the gap: A unified video comprehension framework for moment retrieval and highlight detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18709–18719. IEEE, 2024.

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.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22056–22065, 2025.

Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18891–18901. IEEE, 2025.

# A Details of RCScore Head Selection

This section gives the full form of the head-selection score used in Sec. 3.3. For each layer–head pair $(\ell, h)$, we compare two sets of value-projection contributions. The first contains query-to-vision contributions,

$$X \sim \{\, c_{ij} : i \in R,\ j \in P \,\}, \tag{7}$$

and the second contains query-to-text contributions within the query side,

$$Y \sim \{\, c_{ij} : i \in R,\ j \in R,\ j \le i \,\}. \tag{8}$$

Intuitively, a visually grounded head should assign larger value-projection contributions to visual tokens than to nearby text tokens when forming the query representation.

The ideal relative-contextualization score is

$$\mathbb{E}[\max(X - Y, 0)]\,, \tag{9}$$

which measures how much query-to-vision contributions exceed query-to-text contributions. Directly estimating this quantity requires the joint distribution of $X$ and $Y$. Following RCStat (Mahapatra et al., 2025), we instead compute an upper-bound score from the marginal empirical CDFs. Let $F_X$ and $F_Y$ denote the empirical CDFs of $X$ and $Y$, and let $S_X(t) = 1 - F_X(t)$ be the survival function of $X$. We define

$$\mathrm{RCScore}(\ell, h) = \int_{-\infty}^{\infty} \min\big(F_Y(t), S_X(t)\big)\,\mathrm{d}t = \int_{-\infty}^{\infty} \min\big(F_Y(t), 1 - F_X(t)\big)\,\mathrm{d}t. \tag{10}$$

This score upper bounds the expected relative contextualization:

$$\mathbb{E}[\max(X - Y, 0)] \le \mathrm{RCScore}(\ell, h). \tag{11}$$

We compute $\mathrm{RCScore}(\ell, h)$ for every layer–head pair and choose the top $K$ heads:

$$\mathcal{H}(P, R) = \underset{(\ell,h)}{\mathrm{topK}}\, \mathrm{RCScore}(\ell, h). \tag{12}$$

We refer to $\mathcal{H}$ as the query-conditioned grounding-head set, since these heads are the ones whose query representations are most strongly contextualized by visual tokens relative to query-side text.

# B Training Details

## B.1 Training Data Details

We train the LoRA adapter in two stages. In Stage 1 we train on a highlight-detection and temporal-grounding (HD/TG) mixture; in Stage 2 we warm-start from the Stage 1 checkpoint and fine-tune on video question answering (VQA) data. Both stages share the same architecture and objective and differ only in the training data and a few optimization hyperparameters. During training, the visual entity is a 2 s clip, so the relevance labels and the predicted scores $\{s_i\}$ (corresponding to $s_n$ of § 3.4) are defined per clip, matching the highlight-detection granularity.

**Training data.** Table 4 lists the datasets and their sizes. Stage 1 combines the QVHighlights and Charades training splits (3,973 query–video pairs); Stage 2 uses the VideoITG-40K subset (1,340 pairs). All splits are disjoint from every evaluation split used in § 6.

**Relevance labels (RL).** For every training pair we build a per-clip label vector $\mathrm{RL}$, where $\mathrm{RL}_i$ denotes the relevance label of clip $i$ and clips not marked relevant receive 0. The saliency source depends on the dataset:

- **QVHighlights (HD).** We use the native graded 0–4 saliency annotations (3 annotators per clip) and set $\mathrm{RL}_i$ to their mean.
- **Charades (TG).** We synthesize saliency from the ground-truth start–end span: clips overlapping the span receive the maximum grade (4), partial overlapping clips receive scores between 1 and 3 while non-overlapping receive 0.

Table 4: Training data. Stage 1 trains the HD/TG checkpoint; Stage 2 fine-tunes it into the VQA checkpoint. Counts are training query–video pairs (queries) and the distinct source videos/clips they cover.

| Stage | Dataset | Videos / clips | Queries | Label |
|---|---|---|---|---|
| Stage 1 (HD/TG) | QVHighlights-train | 3,110 | 3,173 | graded 0–4 |
| | Charades-train | 421 | 800 | graded 0–4 |
| | **Combined** | 3,531 | **3,973** | – |
| Stage 2 (VQA) | VideoITG-40K subset (Charades + NExT-QA) | 1,340 | 1,340 | graded $\{0, 2, 4\}$ |
| | **Combined** | 1,340 | **1,340** | – |

- **VideoITG-40K (VQA).** VideoITG-40K annotates query-relevant frames at two granularities. A 2 s clip receives grade 4 if its bin contains a frame-level relevant second and grade 2 if it falls only within a coarser relevant clip, giving a two-level graded label.

In all cases the labels supervise the *relative ranking* of clips within a video, not absolute scores.

### B.2 Training objective

We train the heatmap as a ranked relevance signal. For each training pair $(V, \mathcal{T}, \mathrm{RL})$ with $N$ chunks, let $\{s_i\}_{i=1}^{N}$ be the predicted chunk scores and let $\mathrm{RL}_i \in \mathcal{L}$ be the relevance label of chunk $i$; in our graded setting $\mathcal{L} = \{0, \ldots, 4\}$. Since only the ordering of chunks matters, we optimize a ranking loss rather than a pointwise regression loss.

We use Smooth-AP (Brown et al., 2020), a differentiable surrogate for average precision. First, we standardize the scores within each video,

$$\hat{s}_i = \frac{s_i - \mu_s}{\sigma_s + \varepsilon}, \qquad \mu_s = \frac{1}{N}\sum_{i=1}^{N} s_i, \qquad \sigma_s = \mathrm{std}_i(s_i). \tag{13}$$

We then replace each hard pairwise rank comparison with a soft comparison

$$G_{ij} = \sigma_\tau(\hat{s}_j - \hat{s}_i) \quad (i \neq j), \qquad G_{ii} = 0, \tag{14}$$

where $\sigma_\tau(x) = (1 + \exp(-x/\tau))^{-1}$. Thus, $G_{ij}$ softly indicates whether chunk $j$ is ranked above chunk $i$.

For graded labels, we evaluate AP at multiple relevance thresholds. For a threshold $m$, define the positive set

$$\mathcal{I}_m = \{i : \mathrm{RL}_i \geq m\}. \tag{15}$$

We use the valid thresholds

$$\mathcal{M} = \{m \in \{2, 3, 4\} : 1 \leq |\mathcal{I}_m| < N\}, \tag{16}$$

so each threshold contains at least one relevant and one irrelevant chunk. The soft rank of chunk $i$ among all chunks is $r_i = 1 + \sum_{j \neq i} G_{ij}$, while its soft rank among chunks relevant at threshold $m$ is $r^+_{i,m} = 1 + \sum_{j \in \mathcal{I}_m,\, j \neq i} G_{ij}$. The per-threshold Smooth-AP is

$$\widehat{\mathrm{AP}}_m = \frac{1}{|\mathcal{I}_m|} \sum_{i \in \mathcal{I}_m} \frac{r^+_{i,m}}{r_i}, \tag{17}$$

and the Smooth-AP loss averages over valid thresholds:

$$\mathcal{L}_{\mathrm{SAP}} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \left(1 - \widehat{\mathrm{AP}}_m\right). \tag{18}$$

This term encourages chunks above each relevance threshold to rank ahead of chunks below that threshold.

Smooth-AP can still be satisfied by placing only a few relevant chunks at the top, especially for long relevant intervals. We therefore add a coverage term that encourages all relevant chunks to score above the irrelevant baseline. For threshold $m$, let

$$\mu_m^- = \frac{1}{N - |\mathcal{I}_m|} \sum_{i \notin \mathcal{I}_m} \hat{s}_i \tag{19}$$

be the mean standardized score of chunks that are irrelevant at that threshold. The soft coverage score and loss are

$$\rho_m = \frac{1}{|\mathcal{I}_m|} \sum_{i \in \mathcal{I}_m} \sigma_\tau(\hat{s}_i - \mu_m^-), \qquad \mathcal{L}_{\text{cov}} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} (1 - \rho_m). \tag{20}$$

The full objective averages the per-example loss over the training set and is optimized only over the LoRA parameters $\{A, B\}$:

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{(V,\mathcal{T},\text{RL}) \in \mathcal{D}} \left(\mathcal{L}_{\text{SAP}} + \lambda_{\text{cov}} \mathcal{L}_{\text{cov}}\right), \qquad \tau = 0.5, \quad \lambda_{\text{cov}} = 0.4, \quad \varepsilon = 10^{-6}. \tag{21}$$

**Hyperparameters.** Table 6 lists the settings. Most are shared across the two stages; only the learning rate, number of epochs, and the Stage 2 warm-start differ.

**Ablation at different fractions of training data.** Refer to Table 5 for details

Table 5: Effect of the amount of HD/TG training data on highlight detection (QVHighlights-val, 1550 pairs; ≥Very-Good mAP / Hit@1). We train the LoRA adapter on stratified 25/50/75% subsets of the QVHighlights+Charades mixture (sampling preserves the per-dataset ratio) using the identical recipe as the full model, and evaluate all checkpoints with the same pipeline.

| Train data | Queries (qvh + charades) | mAP | Hit@1 |
|---|---|---|---|
| 25% | 993 (793 + 200) | 45.5 | 75.4 |
| 50% | 1986 (1586 + 400) | 46.6 | 77.7 |
| 75% | 2980 (2380 + 600) | 47.1 | **78.8** |
| 100% | 3973 (3173 + 800) | **47.3** | 77.9 |

# C Baseline Details

Unless noted otherwise, every content-aware sampler scores a common candidate pool of frames extracted at 1 fps (subsampled to at most 512 frames for very long videos) and then selects the target budget of $k$ frames. Within each comparison block all samplers hand their selected frames to the *same* downstream model with the *same* prompt, so only the frame selection differs. Table 7 summarizes which prompt each component receives, and the prompts themselves are listed in Appendix D.

## C.1 Highlight Detection

For all highlight-detection baselines except VideoITG and BLIP-ITM, namely Moment-DETR, UniVTG, R$^2$-Tuning, LLMEPET, CG-DETR, and FlashVTG, we report the numbers as given in the FlashVTG paper (Cao et al., 2025). We run VideoITG and BLIP-ITM ourselves under the same evaluator as our method:

- **VideoITG.** We use the official `nvidia/VideoITG-8B` selector (Eagle: SigLIP + Qwen2-7B) at target fps 1 with up to 512 frames. VideoITG is instruction-conditioned by design; for highlight detection we pass the raw query as that instruction, with no additional template. A single forward pass yields a per-frame relevance logit, whose sigmoid gives the frame-level saliency curve; we mean-pool it into 2 s clip scores and evaluate with the official QVHighlights evaluator.
- **BLIP-ITM.** A training-free scorer using LAVIS `blip_image_text_matching` (large, 446M). Each 1 fps frame is scored by the softmax image–text matching probability $P(\text{match})$ against the raw query, and frame scores are mean-pooled into 2 s clip saliency.

Table 6: Training hyperparameters. The adapter is first trained on the HD/TG mixture (Stage 1) and then fine-tuned on the VQA data (Stage 2), warm-started from the Stage 1 checkpoint. Settings shared by both stages are listed once; only the stage-specific rows differ.

| **Hyperparameter** | **Stage 1 (HD/TG)** | **Stage 2 (VQA)** |
|---|---|---|
| *Adapter (shared)* | | |
| Base VLM | Qwen3-VL-8B-Instruct (frozen) | |
| LoRA rank $r$ / $\alpha$ / dropout | 16 / 32 / 0.05 | |
| Target modules | $W_q, W_k, W_v, W_o$, all 36 layers | |
| Precision | bf16 | |
| *Selector & loss (shared)* | | |
| Heads kept $K$ (detached) | 5 | |
| Smooth-AP temperature $\tau$ | 0.5 | |
| Grade thresholds | $\{2, 3, 4\}$ | |
| Loss weights (SAP / coverage) | 1.0 / 0.4 | |
| Max frames per video | 150 (at $\leq 1$ fps) | |
| *Optimization (shared)* | | |
| Optimizer | AdamW $(\beta_1, \beta_2) = (0.9, 0.95)$ | |
| Weight decay / grad clip | 0.0 / 1.0 | |
| LR schedule / warmup | cosine / 0.03 | |
| Effective batch size | 8 (grad-accum $4 \times 2$ GPUs) | |
| Seed | 0 | |
| *Stage-specific* | | |
| Learning rate | $5 \times 10^{-5}$ | $5 \times 10^{-6}$ |
| Epochs | 6 | 3 |
| Initialization | random LoRA | Stage-1 checkpoint |

### C.2 Temporal Grounding

Each sampler selects $k \in \{8, 16\}$ frames, which are passed to the shared span-predicting grounder (TimeLens-8B or Qwen3-VL-8B) with $\mathcal{P}_{\text{gnd}}^{\text{TimeLens}}$ or $\mathcal{P}_{\text{gnd}}^{\text{Qwen}}$ respectively. Every sampler scores against the same text, the event description with no answer options, so the four selectors receive identical query information. We reproduce all numbers in this block ourselves.

- **VideoITG.** The `nvidia/VideoITG-8B` selector (target fps 1, up to 512 frames) produces per-frame sigmoid relevance from one forward pass; we pass the raw event description as its instruction, with no additional template, then keep the top-$k$ ranked frames and sort them chronologically.
- **BOLT.** Frames are scored by CLIP-L/14 cosine similarity to the query (fp16, scaled by the CLIP logit scale). We select frames by inverse-transform sampling over the similarity curve with the BOLT sharpness exponent $\alpha = 2.5$, within the range recommended by Liu et al. (2025).
- **AKS.** Each 1 fps frame is scored by BLIP-ITM (large) softmax matching against the query; frames are then chosen by the recursive judge-and-split procedure of Tang et al. (2025), using the default split threshold $t_1$=0.8 from their released implementation.

### C.3 Video Question Answering

Each sampler selects $k \in \{16, 32\}$ frames, which are passed to the shared answering VLM (Qwen3-VL-8B or InternVL3.5-8B) with $\mathcal{P}_{\text{ans}}$. The content-aware selectors are conditioned on the options-conditioned prompt $\mathcal{P}_{\text{sel}}$, except AKS as noted below. We reproduce all numbers in this block ourselves.

- **VideoITG.** The same `nvidia/VideoITG-8B` selector, conditioned on $\mathcal{P}_{\text{sel}}$; we keep the top-$k$ ranked frames in chronological order.
- **BOLT.** CLIP-L/14 cosine similarity to $\mathcal{P}_{\text{sel}}$, with inverse-transform sampling using $\alpha = 3.0$ for VideoMME and $\alpha = 2.5$ for LongVideoBench. Liu et al. (2025) do not specify the

query text for multiple-choice benchmarks; we condition on the full option list so that BOLT receives the same information as the other content-aware selectors.

- **AKS.** BLIP-ITM softmax matching followed by the same judge-and-split sampling. Because BLIP's text encoder is short-context and would truncate the long option prompt, AKS scores against the raw question rather than $\mathcal{P}_{\text{sel}}$.

## D PROMPTS

Table 7: Which prompt is used where. Within each task, all samplers feed their selected frames to the same downstream model with the same prompt, so only the selection differs.

| Task | Stage | Model | Prompt |
|---|---|---|---|
| HD | scoring | THMAP (ours)*, VideoITG, BLIP-ITM | raw query, no template |
| TG | selection | THMAP (ours)*, VideoITG, BOLT, AKS | event description, no template |
| | grounding | TimeLens-8B | $\mathcal{P}_{\text{gnd}}^{\text{TimeLens}}$ |
| | grounding | Qwen3-VL-8B | $\mathcal{P}_{\text{gnd}}^{\text{Qwen}}$ |
| VQA | selection | THMAP (ours)*, VideoITG, BOLT | $\mathcal{P}_{\text{sel}}$ |
| | selection | AKS | raw question† |
| | answering | Qwen3-VL-8B, InternVL3.5-8B | $\mathcal{P}_{\text{ans}}$ |

*On all three tasks THMAP additionally prepends a fixed system prompt ($\mathcal{P}_{\text{sys}}$) before the frame tokens; the query text listed above is identical across all samplers.
†AKS scores against the raw question rather than $\mathcal{P}_{\text{sel}}$, because BLIP's text encoder is short-context and would truncate the option list.

**$\mathcal{P}_{\text{sys}}$: THMAP system prompt (all tasks)**

```
You are given a video (as a sequence of frames).  Your role is
to temporally localize where in the video is the given query
taking place.
```
*Prepended once before the frame tokens. The query then follows them with no further template: the raw query for highlight detection, the event description for temporal grounding, and $\mathcal{P}_{sel}$ for video question answering. The text-to-vision value-projection contributions $c_{ij}$ are read from this single forward pass and no text is generated.*

**$\mathcal{P}_{\text{sel}}$: Frame-selection prompt (VQA; THMAP, VideoITG, BOLT)**

```
Select the best answer to the following multiple-choice
question based on the video and the subtitles.  Respond
with only the letter (A, B, C, or D) of the correct option.
{question}
A. {option A}
B. {option B}
C. {option C}
D. {option D}
The best answer is:
```
*Identical in text to $\mathcal{P}_{ans}$ (VideoMME). For VideoITG, the per-frame relevance logits are read from this same forward pass.*

**$\mathcal{P}_{\text{ans}}$: VQA answering prompt (Qwen3-VL-8B / InternVL3.5-8B)**

**VideoMME (short / medium / long):**

```
Select the best answer to the following multiple-choice
question based on the video and the subtitles.  Respond
with only the letter (A, B, C, or D) of the correct option.
{question}
A. {option A}
B. {option B}
```

```
C. {option C}
D. {option D}
The best answer is:

LongVideoBench:
{question}
A. {option A}
...
Answer with the option's letter from the given choices
directly.
```

**$\mathcal{P}_{\text{gnd}}^{\text{Qwen}}$: Temporal-grounding prompt (Qwen3-VL-8B grounder)**

```
You are given {N} frames sampled from a {duration}-second
video, each labeled with its timestamp in seconds.  Identify
the start and end time (in seconds) of the moment described
by the query.
Query:  {query}
Answer with ONLY two numbers:  start_seconds end_seconds
(e.g. '4.0 6.5').  The window must lie within 0 and
{duration} seconds.
```

**$\mathcal{P}_{\text{gnd}}^{\text{TimeLens}}$: Temporal-grounding prompt (TimeLens-8B grounder)**

```
Please find the visual event described by the sentence
'{query}', determining its starting and ending times.  The
format should be:  'The event happens in <start time> -
<end time> seconds'.
```

## E TRAINING-FREE RELEVANCE: READING THE HEATMAP WITHOUT ADAPTATION

This section isolates the readout from the adapter. We remove LoRA and compute the same value-projection heatmap directly from the frozen Qwen3-VL-8B model, keeping the head-selection rule, chunk scoring, sampler, prompts, downstream models, and evaluation settings unchanged. Thus, **Ours w/o LoRA** differs from the full method only in that no adapter is trained or inserted.

Table 8: Effect of LoRA training on highlight detection (QVHighlights val, $\geq$VeryGood mAP / Hit@1). The full baseline comparison is in Table 1.

| Method | Backbone / feat. | mAP | Hit@1 |
|---|---|---|---|
| VideoITG | VideoITG-8B | 35.58 | 49.29 |
| BLIP-ITM | BLIP (446M) | 41.6 | 69.7 |
| FlashVTG | InternVideo2 | 44.15 | 72.90 |
| Ours w/o LoRA | Qwen3-VL-8B (frozen) | 26.9 | 39.6 |
| **Ours** | Qwen3-VL-8B | **47.33** | **77.94** |

Without adaptation, the readout performs poorly on highlight detection. This task evaluates the full ranking of all 2 s clips, including many visually similar or irrelevant clips, so small misorderings throughout the relevance curve strongly affect mAP. The frozen readout often places mass near the correct region, but its clip-level ordering is noisy; LoRA sharpens this signal into a usable saliency ranking.

Temporal grounding is more forgiving because the sampler only needs to place frame density near the queried span. The frozen readout improves over uniform sampling in most settings and occasionally matches stronger selectors, but it remains below the trained adapter overall. This indicates that

Table 9: Effect of LoRA training on temporal grounding (mIoU / R@0.5, %).

| | | Charades-TimeLens | | | | QVHighlights-TimeLens | | | | ActivityNet-TimeLens | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $k=8$ | | $k=16$ | | $k=8$ | | $k=16$ | | $k=8$ | | $k=16$ | |
| **Grounder** | **Method** | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 | mIoU | R@0.5 |
| TimeLens | Uniform | 26.1 | 18.1 | 36.5 | 31.5 | 25.2 | 20.8 | 33.7 | 31.6 | 20.0 | 17.8 | 24.9 | 22.0 |
| | VideoITG | 32.7 | 26.8 | 42.2 | 39.3 | 32.3 | 27.0 | 42.7 | 41.5 | 26.9 | 24.4 | 34.6 | 32.8 |
| | Ours w/o LoRA | 27.2 | 19.7 | 36.1 | 31.9 | 29.0 | 24.7 | 38.1 | 38.1 | 22.3 | 20.7 | 29.1 | 26.9 |
| | **Ours** | **39.0** | **38.3** | **45.5** | **46.4** | **41.8** | **45.6** | **50.8** | **55.9** | **28.1** | **27.2** | **37.5** | **39.8** |
| Qwen3-VL | Uniform | 31.6 | 28.4 | 37.5 | 35.4 | 29.1 | 27.4 | 39.4 | 41.5 | 21.2 | 18.0 | 27.3 | 24.7 |
| | VideoITG | 35.2 | **35.7** | **42.1** | **44.7** | 34.8 | 32.2 | 47.0 | 48.9 | 29.9 | 30.0 | 38.3 | **41.0** |
| | Ours w/o LoRA | 29.1 | 25.3 | 37.6 | 36.7 | 36.8 | 38.0 | 45.1 | 48.3 | 26.5 | 24.1 | 34.6 | 33.5 |
| | **Ours** | **36.0** | 35.4 | 40.2 | 40.9 | **47.8** | **53.3** | **54.7** | **60.4** | **32.9** | **33.8** | **39.0** | 39.0 |

the frozen VLM already contains a usable query-conditioned temporal signal, while LoRA makes it sharper and more reliable.

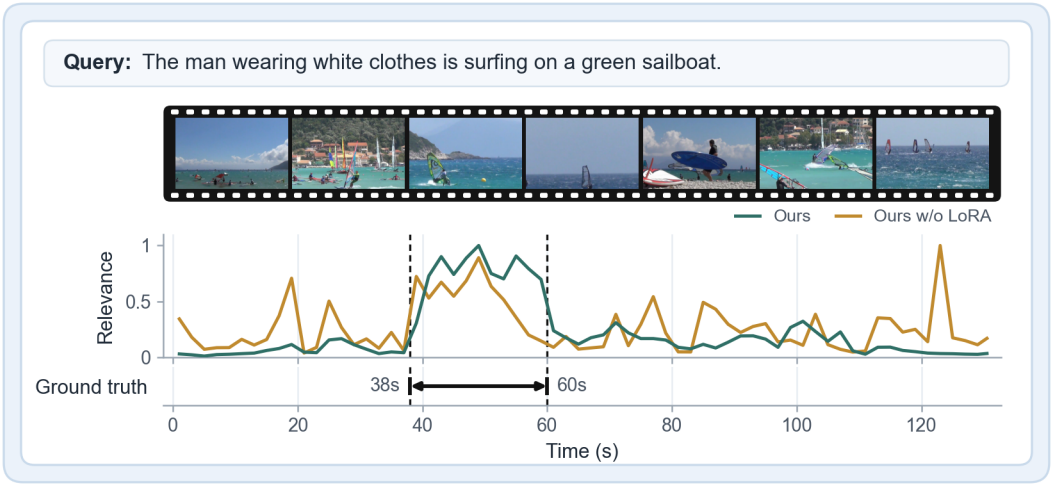



Figure 4: Relevance curves with and without the LoRA adapter on the same video-query pair.

Figure 4 illustrates the effect qualitatively. Both curves place relevance near the annotated span, but the unadapted curve spreads mass over distractor regions, whereas the adapted curve suppresses them and concentrates on the target interval. Overall, the readout is not sufficient as a training-free method for dense ranking, but it provides the latent signal that the small LoRA adapter calibrates into the final THMAP heatmap.

# F Additional Experiments

## F.1 Head-selection analyses

We analyze the query-conditioned head-selection procedure from Sec. 3.3. For each example $i$, all layer–head pairs are ranked by their RCScore and the top $K$ heads are kept:

$$\mathcal{H}^i = \underset{(\ell,h)}{\text{top-K}}\, \text{RCScore}^i(\ell, h), \qquad |\mathcal{H}^i| = K. \tag{22}$$

For Qwen3-VL-8B, this selection is made from $36 \times 32 = 1152$ candidate heads.

### F.1.1 Head stability during training

We first test whether head selection stabilizes as the LoRA adapter trains. We save checkpoints before training and every 60 optimizer steps, giving 25 checkpoints in total. For each checkpoint $t$

and each example $i$ in a held-out set of 75 videos, we recompute the selected head set $\mathcal{H}_t^i$. We then compare it to the head set selected by the final checkpoint $\mathcal{H}_T^i$ using the Jaccard index,

$$J_t^i = \frac{|\mathcal{H}_t^i \cap \mathcal{H}_T^i|}{|\mathcal{H}_t^i \cup \mathcal{H}_T^i|}. \tag{23}$$

At each step, we report the mean Jaccard score over the evaluation videos and show a $\pm 1$ standard deviation band, which measures variation across examples rather than uncertainty of the mean.

Figure 3a shows that the selected heads converge steadily during training. For both $K = 5$ and $K = 10$, the mean Jaccard score increases from roughly $0.3$ at initialization to about $0.97$ at the end of training, indicating that the adapter settles on a stable grounding-head set rather than selecting heads erratically.

#### F.1.2 Number of selected heads $K$

We next vary the number of selected heads using the trained checkpoint. We evaluate $K \in \{1, 5, 10, 15\}$ on 500 examples each for highlight detection on QVHighlights-val (VeryGood mAP) and temporal grounding on QVHighlights-TimeLens (mIoU, using an 8-frame budget). All other settings are fixed.

As shown in Fig. 3b, using a single head is insufficient, while performance saturates at $K = 5$. Selecting more heads gives no further benefit and slightly degrades temporal grounding, likely because lower-ranked heads add example-specific noise. We therefore use $K = 5$ in all experiments.

### F.2 Different Stage-1 Backbone

**Backbone transferability.** To test whether the method depends on a particular VLM backbone, we replace the frozen model used to read value-projection relevance signals and retrain only the lightweight LoRA adapter, keeping the readout, head selection, scoring, and training setup unchanged. Table 10 reports highlight detection results on the QVHighlights validation split for two different backbones: Qwen3-VL-8B and the larger Nemotron-Nano-12B (Deshmukh et al., 2025). The two models obtain nearly identical performance (47.33 vs. 47.25 mAP, and 77.94 vs. 77.68 Hit@1), indicating that THMAP is not tied to a single backbone. The relevance signal appears to be available across model families, and in this setting the larger backbone does not provide additional gains.

Table 10: Highlight detection on the QVHighlights validation split ($\geq$VeryGood mAP / Hit@1). Our method transfers across Stage-1 backbones, matching performance with either Qwen3-VL-8B or Nemotron-Nano-12B.

| Stage-1 backbone | mAP | Hit@1 |
|---|---|---|
| Qwen3-VL-8B | 47.33 | 77.94 |
| Nemotron-Nano-12B | 47.25 | 77.68 |

**Heatmap-generation latency.** Figure 5 reports the time to generate the relevance signal for a single video as a function of its duration. Each point is one real (video, query) pair, drawn from Charades-STA, QVHighlights, and ActivityNet Captions. All measurements use Qwen3-VL-8B with SDPA attention on a single NVIDIA A100 (80 GB), sampling frames at 1 fps. At our operating resolution each frame is resized to $192 \times 96$ pixels and encoded as 18 tokens, so a 150-frame video forms a prefill of $\sim$3,000 tokens; latency grows near-linearly with duration, staying under $0.65$ s up to 150 s and reaching $\sim$1.2 s at $\sim$240 s. VideoITG's selector is comparable in scale (8B) but encodes every frame with SigLIP at $384 \times 384$, and incurs a $\sim$0.7 s fixed cost even on the shortest clips. The gap is therefore driven by input resolution rather than model size: the value-projection relevance cue is coarse and does not require high spatial detail. All of our reported HD, temporal-grounding, and VQA numbers are produced at this same $192 \times 96$ setting. For reference we also plot our own model forced to $384 \times 384$ (144 tokens/frame under Qwen3-VL's patch merging); it remains competitive on short videos but scales less favorably on long ones.

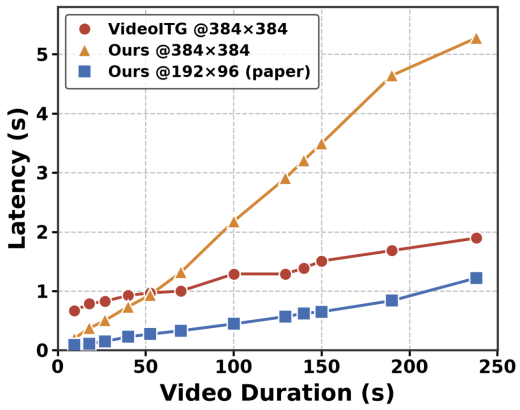



Figure 5: Heatmap-generation latency vs. video duration, on a single NVIDIA A100.

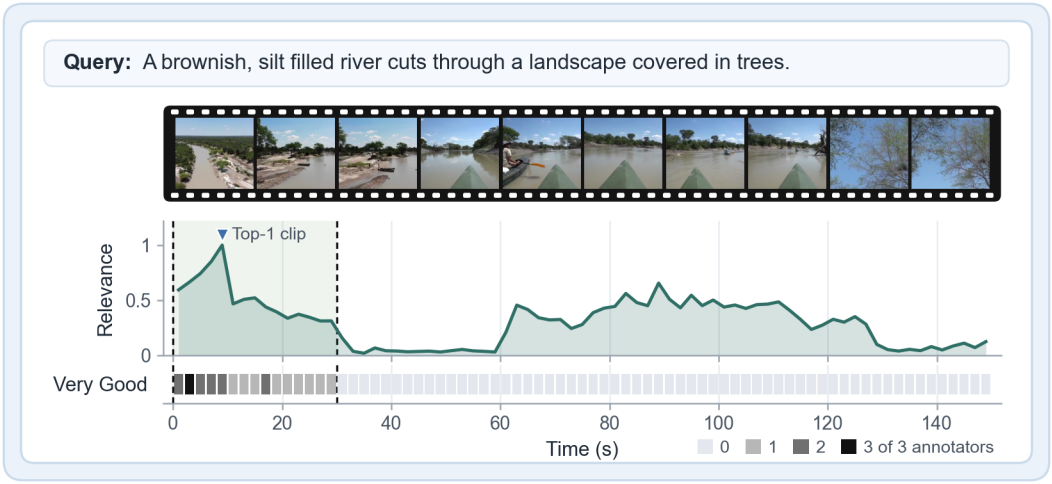



Figure 6: **Highlight detection** on QVHighlights (val, qid 5958). Teal: our per-clip relevance. Shaded: clips at least one annotator rated Very Good. The strip below counts how many of the three annotators rated each clip Very Good, and the triangle marks our top-scoring clip.

## G Qualitative Examples

We show the three examples of Figure 1 at full size. In each, the teal curve is THMAP's relevance (2 s chunks, normalized, unsmoothed), the shaded band marks the query-relevant region, and vertical lines mark the $k = 8$ frames chosen by uniform sampling (orange) and our CDF sampler (blue, Eq. 3).

**Highlight detection.** Figure 6 shows QVHighlights val query 5958. Our top-scoring clip (8–10 s) lies in the aerial shot that annotators rate Very Good most consistently (HIT@1 100, mAP 58.3). Most of the lost mAP comes from 60–130 s, which shows the same brown river but lies outside the annotated window.

**Temporal grounding.** Figure 7 shows ActivityNet-TimeLens video `v_LIJBolW8k5o`. Uniform sampling puts one of its eight frames in the ground-truth moment (38–60 s); ours puts four there, and the predicted span improves from IoU 0.59 to 0.77.

**Video question answering.** Figure 8 shows Video-MME question 051-2, which asks about the Indus and Yangtze scenes (shaded; boundaries taken from the video's on-screen titles). Uniform

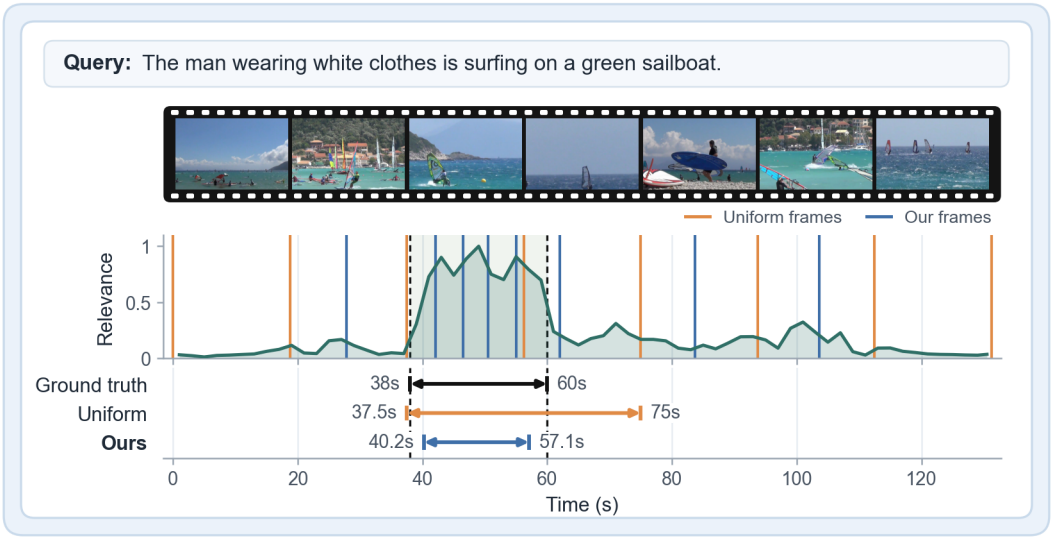



Figure 7: **Temporal grounding** on ActivityNet-TimeLens (video v_LIJBolW8k5o). Shaded: ground-truth moment (38–60 s). Vertical lines: the $k = 8$ frames from uniform sampling (orange) and from our sampler (blue). Bottom: the ground truth and the span predicted from each set of frames.

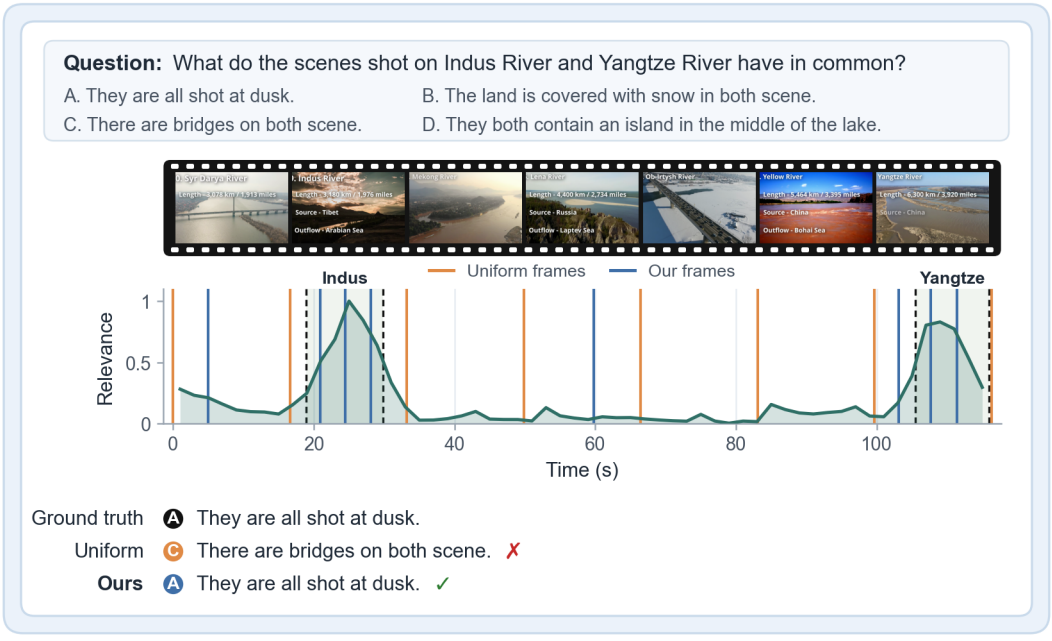



Figure 8: **Video question answering** on Video-MME (question 051-2). Shaded: the Indus and Yangtze segments the question refers to. Vertical lines: the $k = 8$ frames given to the model by uniform sampling (orange) and by our sampler (blue). Bottom: the ground-truth answer and each method's prediction.

sampling misses both scenes and answers incorrectly; ours places five of its eight frames in them and answers correctly.