From Matching to Reasoning: Query-aware Long Video Summarization

Mingu Kang
Seoul National University
gms5560@snu.ac.kr

Sumin Kim
Seoul National University
sumink@snu.ac.kr

Hyunjin Lee
Seoul National University
leel1257@snu.ac.kr

Sungmin Yang
Seoul National University
yangsp704@snu.ac.kr

Yoori Oh[†]
Seoul National University
yoori0203@snu.ac.kr

Joonseok Lee[†]
Seoul National University,
Hanwha Aerospace
joonseok@snu.ac.kr

Abstract

Query-aware video summarization (QVS) aims to select a concise set of moments relevant to a textual query in the context of the entire video. In long videos, addressing a query can require complementary evidence dispersed across distant moments. When such evidence must be combined under a limited summary budget, the usefulness of an additional segment depends not only on its relevance to the query, but also on which query-relevant information is already covered. We propose **QLVSumm**, which models this query-aware composition process through successive segment selections conditioned on the query, full video context, and partial summary. The model learns from unordered reference summaries while conditioning on its own selections, accommodating multiple valid selection orders. We introduce **LQVS**, a large-scale benchmark pairing long videos with multiple open-vocabulary queries and budget-constrained extractive summaries. By pairing each video with multiple queries, LQVS also enables evaluation of how summaries vary across different information needs. Query Disentanglement Consistency (QDC) complements per-query evaluation by measuring whether queries with more similar reference summaries also have more similar predicted summaries within the same video. Experiments on LQVS demonstrate state-of-the-art performance and stronger cross-query consistency, with gains on queries requiring evidence from multiple parts of a video.

1 Introduction

Video summarization aims to select a concise set of moments that preserves the essential content of a video within a limited time budget. Most prior work [23, 24, 40, 19, 13, 2, 9] has focused on generating a general-purpose summary, composed of broadly important and representative moments. This setting is well-suited for relatively short videos centered on a single event, where the summary objective is often less ambiguous. For longer videos with more complex temporal structures, however, the appropriate summary often varies substantially depending on the viewer's information need [22, 43, 37]. For example, when summarizing a 90-minute soccer game into 5 minutes, one viewer may want the moments most critical to the outcome of the game, while another may focus on the actions of a particular player. Such diverging user intents often lead to substantially different selections of moments even from the same video.

[†]Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

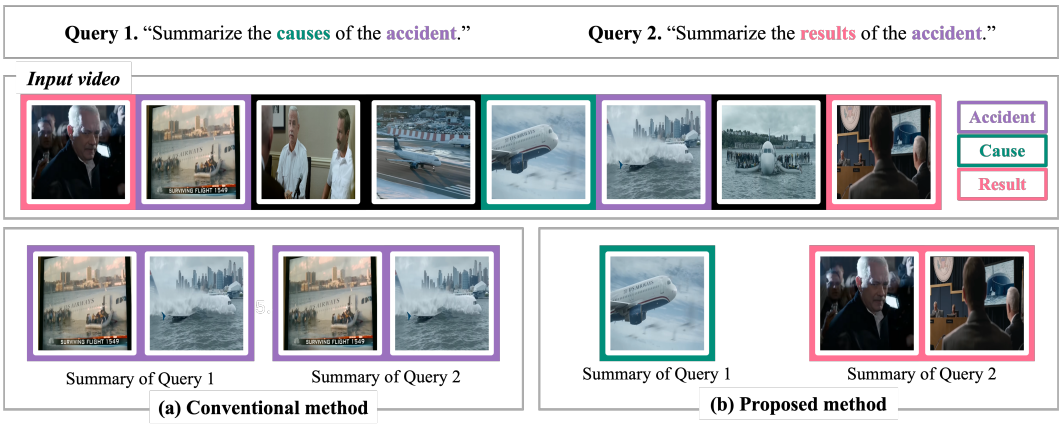


Figure 1: **Motivation for query-aware summary composition.** Given two queries about the same accident, a conventional method may select similar accident-related segments, whereas our proposed method selects intent-specific supporting segments for the cause and result queries.

This motivates Query-aware Video Summarization (QVS), which aims to summarize a video *conditioned on* a natural language query that specifies the user's intent. QVS requires selecting moments that collectively address the information requested by the query from the video [38, 39]. Unlike the conventional (unconditional) video summarization, the expected summary video can be dramatically different depending on the query. Fig. 1 illustrates an example; on the same video about a flight accident, the resulting summaries in (b) are completely different depending on the 'causes' and the 'results' requested in the queries. A cause-focused query calls for preceding events (bird-strike), while a result-focused query calls for subsequent outcomes (trial of the pilot). To sum up, the same event requires different supporting moments depending on the query.

Such supporting evidence can be even more subtle and temporally dispersed in long-form videos. Previous QVS methods [50, 47, 46, 18, 12, 34] estimate query-conditioned segment scores and subsequently construct a summary through segment selection. However, query-conditioned relevance alone does not determine how multiple relevant moments should be combined to address the query as a whole, since the value of a segment also depends on other content selected in the summary.

This motivates a query-aware composition process that iteratively updates the summary, conditioned on the query and the current partial summary, so that the subsequent choices can contribute information that is still needed to address the query. In this paper, we propose **Query-aware Long Video Summarization (QLVSumm)**, which models this query-aware composition process through successive segment selections under a duration budget. Each choice is conditioned on the query, the full video context, and the partial summary, allowing subsequent choices to account for the query-relevant information already included. The model learns this process from unordered reference summaries, allowing multiple valid selection orders while learning to continue from its own choices.

Although this might look like a straightforward extension of the conventional video summarization to a query-conditioned version, QVS on long-form videos suffers from the lack of suitable training data, which require a triplet of a long video, a suitable query, and its corresponding summary. There is only one existing benchmark for this task: QFVS [39]. However, containing only 4 videos and a limited set of queries constructed from predefined visual concepts only, this dataset is not suitable for training, and even not for delicate evaluation. To fill in this gap, we introduce the **Long-form Query-aware Video Summarization (LQVS)** benchmark, which pairs 1,383 long videos with multiple open-vocabulary queries and budget-constrained summaries, yielding 25,972 query-video pairs, constructed by a fully-automated annotation pipeline employing an LLM. For precise evaluation and analysis, the test split has been fully verified manually. By pairing each video with multiple queries, LQVS also enables evaluation of cross-query summary relationships. To this end, we introduce Query Disentanglement Consistency (QDC), which measures whether the inter-query similarity structure of predicted summaries is consistent with that of the reference summaries.

Our contributions are summarized as follows:

- We propose **QLVSumm**, which models query-aware summary composition by conditioning each segment selection on the previously-selected content as well as the query, for the first time.
- We present a large-scale benchmark for long-form QVS (**LQVS**), pairing long videos with multiple open-vocabulary queries and corresponding summaries.
- Extensive experiments demonstrate state-of-the-art summarization performance and stronger cross-query consistency. Analyses of multi-segment evidence and diverse query intents further support the value of query-aware summary composition.

2 Related Work

Video Summarization. Generic video summarization aims for query-agnostic content coverage, typically by predicting frame- or shot-level importance scores. Early approaches [49, 51, 31, 14, 53, 52] have focused on capturing local temporal patterns with recurrent neural networks (RNNs) [7, 15], struggling to capture long-range dependencies. To take such longer-range context into account, hierarchical designs and temporal attention mechanisms [44] have been explored [23, 22, 20, 3, 54, 30, 25, 8, 16, 27]. These methods are limited to producing only a general summary for each video, without adapting to specific user intents or open-ended queries.

QVS addresses this limitation by prioritizing content aligned with an additional natural language query. Early approaches formulate QVS as a budgeted subset selection problem, optimizing relevance and diversity with Determinantal Point Processes-style objectives [39, 38], while recent methods leverage pretrained vision-language models for improved semantic alignment [33, 36]. Recent QVS models tend to adopt query-conditioned frame or segment scores [17, 50, 47, 46, 18, 12, 45]. Building on contextual relevance estimation and structured selection, our method explicitly conditions each next-segment decision on the partial summary, enabling selections to adapt as query-relevant evidence accumulates across a long video.

Video Summarization Datasets. Standard benchmarks such as SumMe [11] and TVSum [41] are widely used for generic summarization, but their short durations and small sample sizes limit evaluation of long-form temporal structure. To mitigate data scarcity, LLM-assisted annotation [4] or weak supervision from "Most Replayed" statistics [42, 23] have been recently explored. However, the former often relies on textual metadata such as subtitles, while the latter collapses diverse user intents into a single scalar signal, making both less suitable for query-conditioned tasks.

Existing benchmarks designed specifically for QVS have addressed the query conditioning issue, but remain limited in scale and expressiveness. For instance, QFVS [39] contains only a few videos, relying on predefined shot-level concept annotations. This design restricts queries to combinations of a closed set of vocabularies, making it difficult to evaluate open-vocabulary natural language queries. Moreover, concept-level supervision provides limited signal for assessing whether a summary is globally coherent and aligned with the query intent.

3 Method

We present our proposed approach, **QLVSumm**, depicted in Fig. 2 It adopts an iterative approach, selecting segment one by one considering the query-relevant content already included in the partial summary available at each decision. It is equipped with the **Segment Adapter** (Sec. 3.2), which maps frame-level features to compact segment representations $\mathcal{H}$ providing the **Query-aware Summary Selector** (Sec. 3.3) access to the full video context. The selector iteratively predicts a segment at a time conditioned on the query and previous selections, while the duration budget is enforced by constrained decoding. We wrap up this section with our training strategy (Sec. 3.4).

3.1 Problem Formulation

The Query-aware Video Summarization (QVS) task aims to produce an extractive summary that serves as an answer to the textual query about the video. A video $\mathcal{X} = \{\boldsymbol{x}_t\}_{t=1}^{T}$ is a sequence of T frames, where $\boldsymbol{x}_t$ is the frame at timestep t. The task is to select an optimal subset $\mathcal{S} \subseteq \mathcal{X}$ that maximizes relevance to the query q, while satisfying the temporal budget constraint: $|\mathcal{S}| \leq \rho \cdot T$, where $\rho \in (0, 1)$ is the budget ratio. However, as this frame-level selection would result in an excessively fragmented summary beyond the minimal perception time of human, it is a common

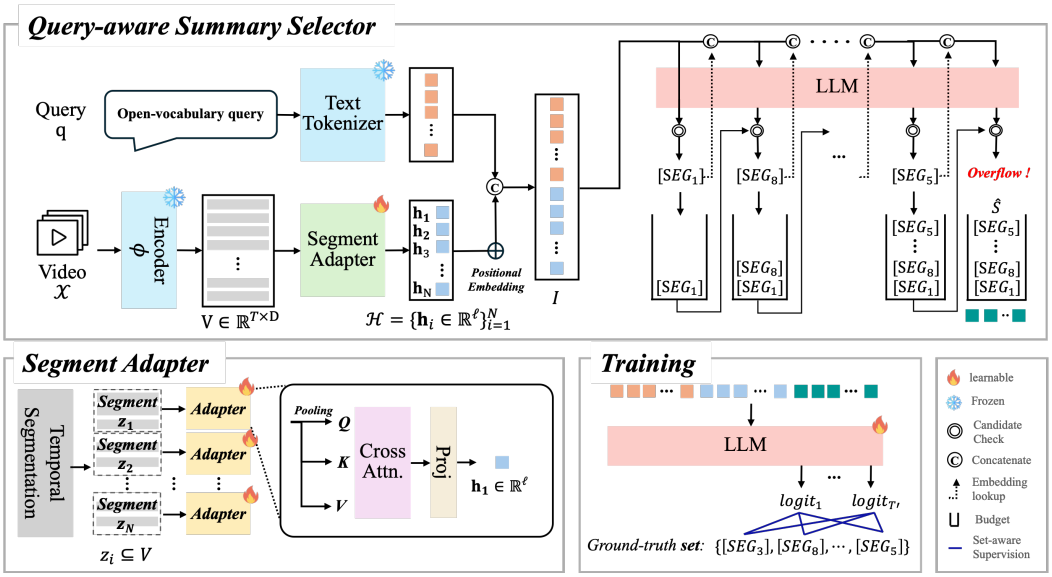


Figure 2: **Overview of the proposed QLVSumm pipeline.** QLVSumm converts a long-form video into compact segment-level tokens via the Segment Adapter, and feeds them with the user query into a Query-aware Summary Selector.

practice to partition the video into N semantically coherent, non-overlapping temporal segments $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^{N}$, where each segment $\boldsymbol{z}_i$ contains $n_i \in \mathbb{Z}^+$ frames, and $\sum_{i=1}^{N} n_i = T$. Then, the QVS task is to select $\mathcal{S} \subseteq \mathcal{Z}$ such that $\sum_{z_i \in \mathcal{S}} n_i \leq \rho \cdot T$. In this paper, we treat long-form videos spanning tens of minutes, and implicitly assume that the frame rate is around 1 FPS, unless noted otherwise.

3.2 Segment Adapter for Long-Form Video Encoding

Despite recent advances, it is infeasible for multimodal LLMs to directly consume video frames, due to the prohibitive memory and computational overhead. To provide a compact interface, we first extract a sequence of frame features $\mathcal{V} = \{\mathbf{v}_t \in \mathbb{R}^d\}_{t=1}^{T}$ using a frozen vision encoder ϕ, where $\mathbf{v}_t = \phi(\boldsymbol{x}_t)$. We follow the common practice to split the input video into chunks of semantic units, denoted by $\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$. Any third-party boundary detection method can be utilized for this, *e.g.*, Kernel Temporal Segmentation (KTS) [35], which identifies change points by minimizing intra-segment visual variance, grouping temporally consecutive and visually similar frames into a segment.

For each segment $\boldsymbol{z}_i$, we aggregate corresponding frame features into a single representative segment-level feature through a cross-attention:

$$\bar{\mathbf{f}}_i = \mathrm{MeanPool}(\mathbf{F}_i), \qquad \tilde{\mathbf{h}}_i = \mathrm{Attn}(\bar{\mathbf{f}}_i, \mathbf{F}_i, \mathbf{F}_i), \qquad \mathbf{h}_i = \mathrm{Proj}(\tilde{\mathbf{h}}_i),$$

where $\mathbf{F}_i \in \mathbb{R}^{n_i \times d}$ is a stack of frame-level features belonging to the i-th segment, $\bar{\mathbf{f}}_i, \tilde{\mathbf{h}}_i \in \mathbb{R}^d$ represent the pooled and attended features, respectively, and $\mathbf{h}_i \in \mathbb{R}^\ell$ is the segment token projected to the LLM hidden dimension ℓ. $\mathrm{MeanPool}(\cdot)$ averages the frame features within the segment to produce a global context vector $\bar{\mathbf{f}}_i$, which serves as the *query* for the attention. The $\mathrm{Attn}(\cdot)$ is a cross-attention function that uses $\bar{\mathbf{f}}_i$ to attend over the frame-level features $\mathbf{F}_i$ (acting as both *key* and *value*), distilling the most salient visual information into a single vector $\tilde{\mathbf{h}}_i$. $\mathrm{Proj}(\cdot)$ then maps this representation into the hidden dimension ℓ of the LLM. Applying this process to all segments yields a sequence of segment-level tokens $\mathcal{H} = \{\mathbf{h}_i \in \mathbb{R}^\ell\}_{i=1}^{N}$, providing the LLM with a concise yet semantically rich view of the entire video.

3.3 Query-aware Summary Selector

Autoregressive Segment Selection. QLVSumm generates a video summary in an autoregressive manner; that is, one additional segment is chosen and added to the summary at each step, given the

input video, the query, and the segments that have been selected so far. Formally, the input $\mathcal{I}$ consists of the token embeddings $\mathcal{Q} = \{\mathbf{q}_j \in \mathbb{R}^\ell\}_{j=1}^M$ of the text query q of length M and the sequence of segment representations $\mathcal{H}$. To preserve the temporal order of segments, we add a learnable segment-level positional embedding to each segment representation $\mathbf{h}_i$. Overall, the input sequence is denoted by

$$\mathcal{I} = \{\mathcal{Q}; \mathcal{H}\} = \{\mathbf{q}_1, \dots \mathbf{q}_M, \mathbf{h}_1, \dots \mathbf{h}_N\}.$$

From this, we train an LLM to select specific segment identifiers, denoted by $[\texttt{SEG}_i]$, which correspond to the temporal intervals from the input video, rather than generating natural language. Including the `EOS` signals the completion of the summary, the restricted output token space $\mathcal{O}$ for a video with N segments is defined as

$$\mathcal{O} = \{[\texttt{SEG}_1], \dots, [\texttt{SEG}_N], \texttt{EOS}\},$$

where each $[\texttt{SEG}_i]$ acts as a unique pointer to the i-th temporal segment.

At decoding, QLVSumm uses the LLM as a constrained next-token predictor. Given the current input context, it first produces next-token logits in the same way as standard language generation. Instead of the natural words, however, it outputs one of the segment identifier tokens in $\mathcal{O}$. The decoder then selects one feasible output token according to the budget-aware decoding rule below. Generating $[\texttt{SEG}_i]$ corresponds to selecting a segment $\boldsymbol{z}_i$, and generating `EOS` terminates the summary. The generated segment identifier is appended to the context, and the next iteration is repeated.

Budget-Aware Decoding. To ensure that the final summary strictly satisfies the temporal constraint, we apply a budget-aware, constrained decoding strategy at inference. At each decoding step t, the model chooses a segment which does not exceed the remaining budget. In order to exclude segments that have been already selected and those whose duration n_i would violate the budget constraint when added, we define the feasible candidate set $\mathcal{C}_t$ over segment identifiers at decoding step t by

$$\mathcal{C}_t = \left\{ [\texttt{SEG}_i] \;\middle|\; [\texttt{SEG}_i] \notin \mathcal{S}_{<t}, \quad \sum_{[\texttt{SEG}_j] \in \mathcal{S}_{<t}} n_j + n_i \leq \rho \cdot T \right\} \cup \{\texttt{EOS}\}. \tag{1}$$

where $\mathcal{S}_{<t}$ denotes the set of segment identifiers selected prior to the decoding step t.

The decoder estimates how valuable each candidate token in $\mathcal{C}_t$ is when added to the summary, given the video and the previously selected segments. Formally, the next segment identifier is greedily selected by

$$s_t = \underset{s \in \mathcal{C}_t}{\text{argmax}} \;\; \pi_t(s \mid \mathcal{I}, \mathcal{S}_{<t}).$$

The decoding process terminates when the model predicts the `EOS` token, or when no additional segment can be selected without violating the budget. This constrained decoding guarantees that the resulting summary always satisfies the predefined duration budget, despite autoregressive selection.

3.4 Training

As the reference summary specifies an unordered *set* of segments while the proposed selector generates an ordered *sequence*, imposing a single target order would unintentionally penalize other valid permutations. For instance, both selection sequences $\boldsymbol{z}_2 \to \boldsymbol{z}_5$ and $\boldsymbol{z}_5 \to \boldsymbol{z}_2$ correspond to the same reference set $\{\boldsymbol{z}_2, \boldsymbol{z}_5\}$, so their loss should be the same. We therefore use *set-aware supervision* that treats any remaining feasible reference segment as a valid next target. This set-valued supervision allows flexible selection orders, since any reference segment that remains feasible at each decoding step is treated as a valid next target, rather than enforcing a single predefined order. (Note that this ambiguity concerns the order of selection, not the temporal positions of the segments in the video.)

Formally, let $\mathcal{Y}$ denote the true set of segment identifiers for the given query. Then, the set $\mathcal{Y}_t$ of valid targets at decoding step t consists of the true and currently eligible candidate segments:

$$\mathcal{Y}_t = \begin{cases} \mathcal{Y} \cap \mathcal{C}_t & \text{if } \mathcal{Y} \cap \mathcal{C}_t \neq \emptyset, \\ \{\texttt{EOS}\} & \text{otherwise}, \end{cases}$$

where $\mathcal{C}_t$ is the feasible candidate set defined in Eq. (1). During training, $\mathcal{C}_t$ is instantiated using the partial summary $\hat{S}_{<t}$ generated up to the previous decoding step. To reduce this training-inference discrepancy [6], we condition the loss on the partial summary $\hat{S}_{<t}$ obtained by a self-rollout with

Table 1: **Comparison of video summarization datasets.** Instance Hrs (hours) indicates total video hours for generic datasets and query-video hours for query-aware datasets. Open indicates public availability of the dataset annotations, features, or construction resources.

Dataset	Video Statistics			Query Properties		Open	Annotation
	Videos	Avg. Duration	Instance Hrs	Count	Open-Vocab		
Generic Video Summarization							
SumMe [ECCV'14]	25	2.4m	1.0	—	✗	✓	Human
TVSum [CVPR'15]	50	3.9m	3.3	—	✗	✓	Human
TL;DW? [ECCV'22]	12.2K	3.1m	628.3	—	✗	✓	Automated
BLiSS [CVPR'23]	13.3K	5.0m	1.1K	—	✗	✓	Human
Mr.HiSum [NeurIPS'23]	31.9K	3.4m	1.8K	—	✗	✓	Most Replayed [42]
LfVS-P [CVPR'24]	250K	13.3m	55.4K	—	✗	✗	Automated
MMSum [CVPR'24]	5.1K	14.5m	1.2K	—	✗	✓	Human
MoSu [ICLR'26]	52.7K	4.5m	4.0K	—	✗	✓	Most Replayed [42]
Query-aware Video Summarization							
QFVS [CVPR'17]	4	254.4m	780.2	46.0	✗	✓	Human
LQVS (Ours)	**1.4K**	**44.1m**	**16.9K**	**25.9K**	✓	✓	**Automated**

greedy budget-aware decoding. The model then learns to select any remaining feasible reference segment from that state.

Our training objective is to maximize the likelihood of selecting a valid target segment, explicitly conditioned on the model's own generation trajectory. Formally, the loss is formulated as

$$\mathcal{L} = -\frac{1}{T'}\sum_{t=1}^{T'} \log\left(\sum_{s\in\mathcal{Y}_t} \pi_t(s \mid \mathcal{I}, \hat{S}_{<t})\right), \tag{2}$$

where T' is the total number of decoding steps.

4 Benchmark

As mentioned in Sec. 1, one of the biggest bottlenecks in long-form QVS is in the lack of training data and evaluation benchmark at a proper scale for long-form videos. To fill in this gap, we design our **Long-form Query-aware Video Summarization (LQVS)** benchmark with 1.4K carefully curated long-form videos. To support learning query-dependent selections over the same visual context, each video is paired with multiple open-vocabulary queries, and for each query, we provide a budget-constrained extractive reference summary. To the best of our knowledge, this is the first benchmark to allow not only training a QVS model with a proper size to treat long-form videos, but also studying the effect of queries on the summary, equipped with various plausible queries on real videos.

4.1 Benchmark Collection

To create this benchmark, we first collect long-form (30-60 mins) YouTube videos from diverse content domains and apply KTS [35] to obtain temporally coherent segments. For each segment, uniformly sampled frames at 1 FPS are fed to Qwen3-VL-32B-Instruct [5] to generate a segment-level visual description. For each video, we also generate open-vocabulary candidate queries that reflect diverse user intents, and let GPT-OSS-120B [1] select a subset of segments that provide an answer to the query under the duration budget. The resulting benchmark contains 25,972 query-video pairs[1], where we randomly split the set at the video level into 18,198 training, 3,891 validation, and 3,883 test pairs. To ensure reliable evaluation, five independent annotators have manually verified the test split, retaining query-reference pairs by majority vote (App. A.4). Further details on the data generation pipeline are provided in App. A.

4.2 Benchmark Statistic

Tab. 1 compares LQVS with existing video summarization benchmarks. LQVS is distinguished from generic summarization benchmarks, which do not support query-conditioned examples. To the best of our knowledge, QFVS [39] is the only existing benchmark that explicitly targets query-aware video summarization. Despite its pioneering role, QFVS contains only 4 videos with predefined concept-based queries. In contrast, our LQVS substantially expands the scale, containing 1,383 videos, 25,972 open-vocabulary queries, and 16.9K query-video instance hours. On average, each video is paired with 18.8 queries and has a duration of 44.1 minutes on average. This average duration in LQVS is significantly longer than most generic summarization benchmarks, well-suited for our study of temporally dispersed evidence and budgeted summary construction.

5 Experiments

5.1 Experimental Setting

Evaluation Protocol. We train all baselines on the LQVS training set and select the best performing one on the validation set. Following the standard [41, 11, 42], we set the budget ratio to $\rho = 0.15$. For baselines that output importance scores, we construct the final summaries using the standard KTS and knapsack pipeline. Generic baselines produce one query-agnostic summary per video, which is evaluated against all query-specific references and averaged across queries. Query-aware methods including ours produce one summary per query and are evaluated against the corresponding reference.

Baselines. We compare our method against both generic and query-conditioned video summarization methods. For generic methods, we compare with VASNet [8], PGL-SUM [3], CSTA [40], SummDiff [22], UMT [29], CFSum [10], and TripleSumm [23]. They produce a query-agnostic summary for each video, providing a reference for how conventional video summarizers perform under query-specific evaluation. For query-aware methods, we compare with QC-DPP [39], CHAN [47], CLIP-It [33], IntentVizor [45], and UniVTG [28].

Evaluation Metrics. We measure the standard QVS metrics, such as ranking correlation and segment-level F1, comparing the predicted summary with its corresponding ground truth. While these metrics effectively measure per-query summary quality, they do not directly reveal whether summaries for diverse queries on the same video effectively reflect distinct user intents. To be specific, a video may contain some core segment which is expected to appear in the summary for most queries. In QVS evaluation, we are less interested in these parts; rather, we care more if the model correctly includes the segments specific to a particular query.

To complement such an aspect, we propose a new metric **Query Disentanglement Consistency (QDC)**, which measures whether the inter-query similarity structure among predicted summaries is consistent with that of the ground truth. This metric is defined per video, collectively with multiple queries on each video. For a video with M queries, let $\mathcal{Y}_a$ and $\hat{\mathcal{Y}}_a$ denote the true and predicted sets of selected segments for a query a, respectively. For each query pair (a, b), we compute the Jaccard similarity between their selected segment sets, and QDC is then defined as the Spearman rank correlation ρ [55] between the upper-triangular entries of the true ($\mathbf{S}_{ab}$) and the predicted similarity matrices ($\hat{\mathbf{S}}_{ab}$):

$$\text{QDC} = \rho\Big(\{\mathbf{S}_{ab}\}_{a<b}, \{\hat{\mathbf{S}}_{ab}\}_{a<b}\Big), \quad \text{where } \mathbf{S}_{ab} = \frac{|\mathcal{Y}_a \cap \mathcal{Y}_b|}{|\mathcal{Y}_a \cup \mathcal{Y}_b|}, \ \hat{\mathbf{S}}_{ab} = \frac{|\hat{\mathcal{Y}}_a \cap \hat{\mathcal{Y}}_b|}{|\hat{\mathcal{Y}}_a \cup \hat{\mathcal{Y}}_b|}. \tag{3}$$

A higher QDC indicates that the model better preserves the relative inter-query relations among query-specific summaries, and vice versa. Intuitively, we expect the model to produce similar summaries for two queries with similar true summaries, and distinct summaries for those with distinct true summaries. Note that QDC is not intended to replace per-query metrics. Together with F1 and ranking-based metrics [55, 21], QDC provides a more comprehensive evaluation by capturing cross-query summary disentanglement. A more detailed discussion of its relation to per-query accuracy and its empirical reliability is provided in App. B.

[1] Each query-video pair is considered as an instance, since query-aware summarization aims to generate different summaries for the same video depending on each query, unlike generic summarization where one video corresponds to a single output.

Table 2: **Performance comparison on LQVS.** (a) Generic and (b) query-aware video summarization results. QDC is undefined for generic methods.

(a) Generic Summarization

Method	Ranking	F1	QDC
VASNet [ACCV'18]	-0.000	11.52	–
PGL-SUM [ISM'21]	0.098	19.83	–
CSTA [CVPR'24]	0.163	25.57	–
SummDiff [ICCV'25]	0.138	23.18	–
UMT [CVPR'22]	0.136	23.06	–
CFSum [ICASSP'25]	0.134	22.87	–
TripleSumm [ICLR'26]	0.153	24.49	–

(b) Query-Aware Summarization

Method	Ranking	F1	QDC
QC-DPP [CVPR'17]	-0.009	10.73	0.000
CHAN [AAAI'20]	0.077	17.99	0.180
CLIP-It [NeurIPS'21]	0.117	21.39	0.046
IntentVizor [CVPR'22]	0.077	17.91	0.198
UniVTG [ICCV'23]	0.166	25.61	0.137
QLVSumm-8B	0.226	30.19	**0.256**
QLVSumm-14B	**0.230**	**30.92**	0.237

Table 3: **Evaluation on S-VideoXum.** Baseline scores are quoted from Mylonas et al. [32].

Method	F-score (%)
CLIP-It	22.80
SD-VSum	24.80
QLVSumm	**25.16**

Table 4: **Ablation study.** Performed for segment adapter (SA), autoregressive selection (AR), set-aware target (ST), and rollout conditioning (RC), on LQVS.

Row	SA	AR	ST	RC	Rank.	F1	QDC
(a)	✗	✗	✗	✗	OOM	OOM	OOM
(b)	✓	✗	✗	✗	0.131	22.75	0.219
(c)	✓	✓	✓	✗	0.220	30.12	0.107
(d)	✓	✓	✗	✓	0.139	17.99	0.218
(e)	✓	✓	✓	✓	**0.226**	**30.19**	**0.256**

Implementation Details. We use Qwen3 [48] as our LLM backbone and CLIP ViT-Large [36] as visual features. The Segment Adapter is initialized from Q-Former weights from BLIP-2 [26], and the LLM is fine-tuned with LoRA. Further experimental details are provided in App. C.

5.2 Main Results

Comparison with Baselines. Tab. 2 indicates that QLVSumm consistently outperforms both generic and query-aware summarization baselines across Ranking, F1, and QDC, evaluated on the LQVS test set. Surprisingly, most existing query-aware summarization methods do not significantly outperform the generic methods, even if they have limited ability to reflect diverse user intents on the same video. On the other hand, our QLVSumm achieves significantly higher performance on all metrics by generating summaries clearly specific to each query.

External Benchmark Evaluation. To verify the generalization capability of our method, we further evaluate QLVSumm on S-VideoXum [32], another video summary benchmark providing human-labeled summaries. According to Tab. 3, QLVSumm achieves stronger performance than the two baselines, proving its effectiveness even outside of the training data distribution. See App. D.4 for more details about the dataset and analysis.

Qualitative Results. Fig. 3 illustrates an example how competing methods adaptively generate a summary depending on the query on a same video. For the bread-making (query 1), CLIP-It selects the final bread alongside title and speaker segments, while QLVSumm includes intermediate preparation steps and the final result. When the query shifts to the closing message (query 2), QLVSumm instead focuses on the closing remarks, while CLIP-It retains unrelated process and title segments.

App. D provides additional analyses on backbone scaling, query categories, and training data.

5.3 Ablation Study

We breakdown the performance gain over the proposed ideas: segment adapter (Sec. 3.2), autoregressive selection (Sec. 3.3), set-aware target and rollout conditioning (Sec. 3.4), reported in Tab. 4.

Long-form Video Representation. Directly operating on all frames (row (a) in Tab. 4) exceeds memory under our environment. Applying our segment adapter, producing compact segment-level representation of the video (row (b)), it runs without an issue.

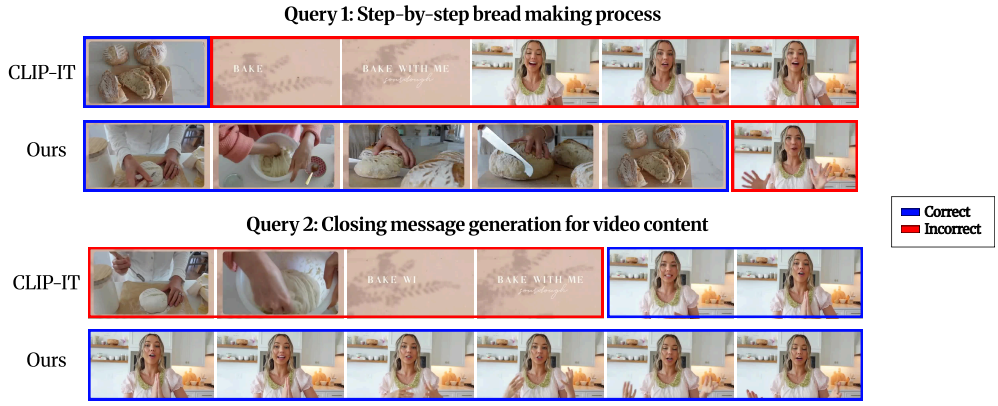


Figure 3: **Qualitative demonstration for diverse queries on a same video.** Blue and red boxes mark selections that match or do not match with the query intent, respectively.

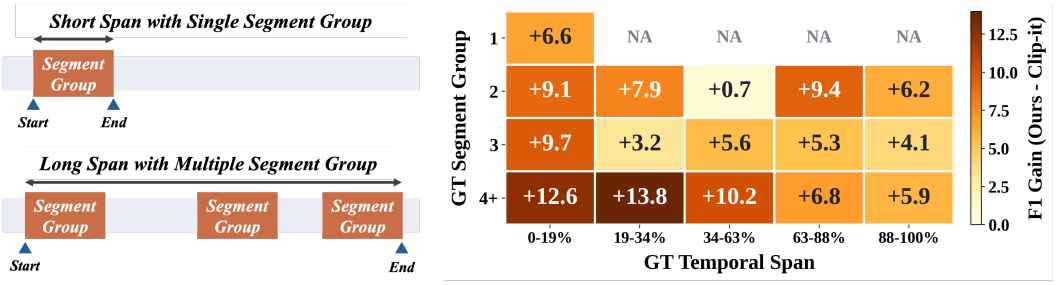


Figure 4: **Performance by Summary composition complexity.** Each cell reports the F1 gain of QLVSumm over CLIP-It. N/A indicates bins with no test examples.

Selection Formulation. The baseline in row (b) predicts an individual score for each segment by minimizing MSE loss, and the final summary is then obtained by knapsack. Comparing with this, our full model (row (e)) significantly outperforms in all metrics, demonstrating the effectiveness of our successive-selection framework (AR in Tab. 4) over the independent segment scoring.

Training the Summary Selector. Among the options using autoregressive selection, we examine the two training components: set-aware target and rollout conditioning. When we remove the set-aware supervision (row (d)), the performance substantially degrades from the full model (row (e)). This result strongly supports that treating the reference summary as an unordered set is more than essential to achieve the high performance. When we remove the rollout conditioning (row (c)) from the full model, on the other hand, the ranking and F1 remain similar (0.220 vs. 0.226 and 30.12 vs. 30.19), whereas QDC significantly drops from 0.256 to 0.107. This result concludes that the rollout conditioning has a stronger effect on the cross-query relations captured by QDC than on per-query overlap. Putting together, this study verifies complementary roles of the set-aware supervision and rollout conditioning in training the selector.

5.4 Analysis

Beyond the component-wise ablations, we further analyze QLVSumm from two complementary perspectives. We first examine how performance varies with summary composition complexity, and then investigate how well the predicted summaries preserve inter-query relationships within the same video.

Performance by Summary Composition Complexity. We next examine the performance gap among the models on the queries whose reference evidence is spread over multiple parts of a video. We characterize each true summary by two aspects. First, we consider the number of separated segments that comprise the summary, assuming that a summary requiring multiple segments to recall would be harder to predict. For example, the two summaries shown in Fig. 4(left) have one and three

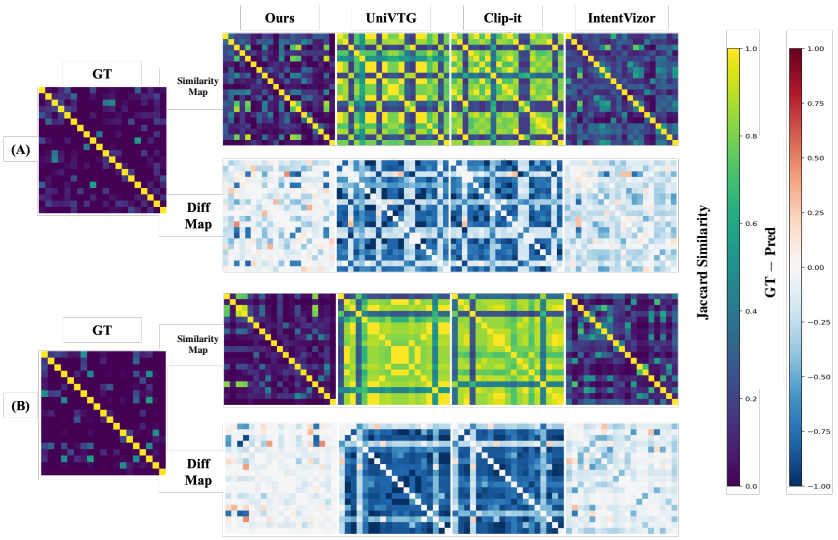


Figure 5: **Visualization of query-aware summary disentanglement.** The ground-truth and predicted pairwise Jaccard similarity matrices are shown for different methods, along with their signed differences ($S^{gt} - S^{pred}$). Blue and red entries in the difference maps indicate overestimated and underestimated overlap, respectively. QLVSumm better preserves the inter-query similarity structure of the reference summaries.

'GT Segment Groups', respectively. The other aspect is the normalized temporal span of the summary, assuming that a shorter one would be more tricky to precisely estimate than the one covering almost the entire video. In the two examples in Fig. 4(left), the summaries span roughly 20% and 50% of the input video.

Fig. 4(right) plots the relative gain of our method over CLIP-It, according to these two complexity factors. QLVSumm improves F1 across all populated bins, with a larger gain when the true summary contains more segments and the true summary spans only a short range of the video, which both are considered as 'more complex' scenarios above.

Visualization of Query Disentanglement. Fig. 5 visualizes the pairwise Jaccard similarity among query-specific summaries for the same video, along with their differences from the ground truth. UniVTG [28] and CLIP-It [33] exhibit pronounced vertical and horizontal bands, indicating that certain segments are repeatedly selected across different queries. In contrast, QLVSumm produces similarity patterns that more closely resemble the ground truth, preserving both distinct and overlapping relationships among query-specific summaries. The smaller deviations from the ground-truth similarity structure further support the effectiveness of QLVSumm in adapting its selections to diverse user intents within the same video.

6 Summary

Long-form query-aware video summarization (QVS) requires composing moments that address a user's information need within a duration budget. Our proposed **QLVSumm** conditions each selection on the query, full video context, and partial summary, learning from reference segment sets through set-aware supervision on its own selection trajectories. We propose a large-scale benchmark, **LQVS**, to fill in the gap in the data bottleneck on this task, enabling evaluation across multiple information needs per video at unprecedented scale. Our extensive experiments verify the benefit of the proposed ideas, including sequential selection, set-aware targets, and rollout conditioning. Analyses of temporally dispersed evidence and qualitative selections further illustrate how summaries cover different stages of a video and adapt to the query.

Acknowledgments

This work was supported by the Youlchon Foundation and Korean government-funded grants from the National Research Foundation of Korea (NRF; RS-2021-NR05515, RS-2024-00336576, and RS-2023-0022663), the Institute for Information & Communication Technology Planning & Evaluation (IITP; RS-2022-II220264, RS-2024-00353131, and RS-2026-25505492), and the National IT Industry Promotion Agency (NIPA; RS-2026-25621677). Additional support was provided by a Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (RS-2026-25525550).

References

[1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv:2508.10925*, 2025.

[2] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. Unsupervised video summarization via attention-driven adversarial learning. In *Proceedings of the International Conference on Multimedia Modeling (MMM)*, 2020.

[3] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, 2021.

[4] Dawit Mureja Argaw, Seunghyun Yoon, Fabian Caba Heilbron, Hanieh Deilamsalehy, Trung Bui, Zhaowen Wang, Franck Dernoncourt, and Joon Son Chung. Scaling up video summarization pretraining with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[5] 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:2511.21631*, 2025.

[6] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[7] Jeffrey L Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[8] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Proceedings of the Workshop on Attention/Intention Understanding (AIU) at the Asian Conference on Computer Vision (ACCV)*, 2019.

[9] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[10] Yaowei Guo, Jiazheng Xing, Xiaojun Hou, Shuo Xin, Juntao Jiang, Demetri Terzopoulos, Chenfanfu Jiang, and Yong Liu. CFSum: A transformer-based multi-modal video summarization framework with coarse-fine fusion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[11] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[12] Jiajun Han, Xuran Yang, and Hui Zhang. Query-focused multimodal summarization with gate-guided mixture-of-experts. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2025.

[13] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] Xufeng He, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Unsupervised video summarization with attentive conditional generative adversarial networks. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2019.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

[16] Tzu-Chun Hsu, Yi-Sheng Liao, and Chun-Rong Huang. Video summarization with spatiotemporal vision transformer. *IEEE Transactions on Image Processing (TIP)*, 32:3013–3026, 2023.

[17] Jia-Hong Huang, Luka Murn, Marta Mrak, and Marcel Worring. GPT2MVS: Generative pre-trained transformer-2 for multi-modal video summarization. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*, 2021.

[18] Pin Jiang and Yahong Han. Hierarchical variational network for user-diversified & query-focused video summarization. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*, 2019.

[19] Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. Discriminative feature learning for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[20] Yunjae Jung, Donghyeon Cho, Sanghyun Woo, and In So Kweon. Global-and-local relative position embedding for unsupervised video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[21] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

[22] Kwanseok Kim, Jaehoon Hahm, Sumin Kim, Jinhwan Sul, Byunghak Kim, and Joonseok Lee. SummDiff: Generative modeling of video summarization with diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[23] Sumin Kim, Hyemin Jeong, Mingu Kang, Yejin Kim, Yoori Oh, and Joonseok Lee. TripleSumm: Adaptive triple-modality fusion for video summarization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

[24] Min Jung Lee, Dayoung Gong, and Minsu Cho. Video summarization with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[25] Haopeng Li, Qiuhong Ke, Mingming Gong, and Rui Zhang. Video joint modelling based on hierarchical transformer for co-summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(3):3904–3917, 2022.

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

[27] Guoqiang Liang, Yanbing Lv, Shucheng Li, Xiahong Wang, and Yanning Zhang. Video summarization with a dual-path attentive network. *Neurocomputing*, 467:1–9, 2022.

[28] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[29] 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 (CVPR)*, 2022.

[30] Yen-Ting Liu, Yu-Jhe Li, Fu-En Yang, Shang-Fu Chen, and Yu-Chiang Frank Wang. Learning hierarchical self-attention for video summarization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2019.

[31] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial LSTM networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] Manolis Mylonas, Evlampios Apostolidis, and Vasileios Mezaris. SD-VSum: A method and dataset for script-driven video summarization. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2025.

[33] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. CLIP-It! language-guided video summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[34] Bryan A Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[35] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[37] Zirui Shang, Yubo Zhu, Hongxi Li, Shuo Yang, and Xinxiao Wu. Video summarization using denoising diffusion probabilistic model. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[38] Aidean Sharghi, Boqing Gong, and Mubarak Shah. Query-focused extractive video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[39] Aidean Sharghi, Jacob S Laurel, and Boqing Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] Jaewon Son, Jaehun Park, and Kwangsu Kim. CSTA: CNN-based spatiotemporal attention for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[41] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TVSum: Summarizing web videos using titles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[42] Jinhwan Sul, Jihoon Han, and Joonseok Lee. Mr. HiSum: A large-scale dataset for video highlight detection and summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[43] Hacene Terbouche, Maryan Morel, Mariano Rodriguez, and Alice Othmani. Multi-annotation attention model for video summarization. In *Proceedings of the Workshop on Large Scale Holistic Video Understanding (LSHVU) at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[45] Guande Wu, Jianzhe Lin, and Claudio T Silva. IntentVizor: Towards generic query guided interactive video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[46] Shuwen Xiao, Zhou Zhao, Zijian Zhang, Ziyu Guan, and Deng Cai. Query-biased self-attentive network for query-focused video summarization. *IEEE Transactions on Image Processing (TIP)*, 2020.

[47] Shuwen Xiao, Zhou Zhao, Zijian Zhang, Xiaohui Yan, and Min Yang. Convolutional hierarchical attention network for query-focused video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[48] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv:2505.09388*, 2025.

[49] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[50] Yujia Zhang, Michael Kampffmeyer, Xiaodan Liang, Min Tan, and Eric P Xing. Query-conditioned three-player adversarial network for video summarization. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[51] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2017.

[52] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. HSA-RNN: Hierarchical structure-adaptive RNN for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[53] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. TTH-RNN: Tensor-train hierarchical recurrent neural network for video summarization. *IEEE Transactions on Industrial Electronics (TIE)*, 68(4): 3629–3637, 2021.

[54] Bin Zhao, Maoguo Gong, and Xuelong Li. Hierarchical multimodal transformer to summarize videos. *Neurocomputing*, 468:360–369, 2022.

[55] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999.

Appendix

A Dataset Details

In this section, we provide the comprehensive details of the MLLM-assisted annotation pipeline, query taxonomy analysis, and the quality control protocols used for the LQVS benchmark.

A.1 Dataset Collection Pipeline

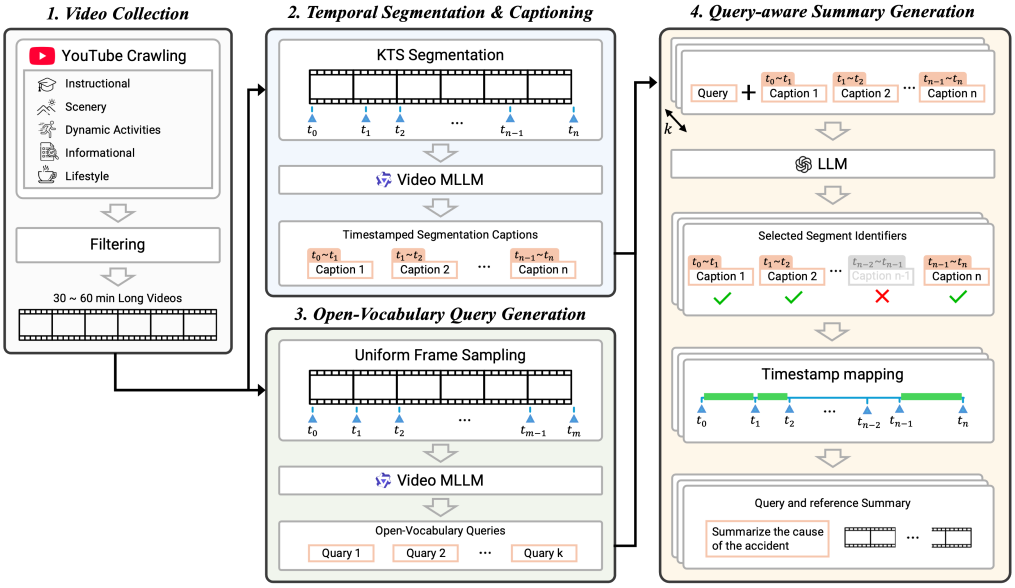


Figure I: **Overview of the LQVS dataset construction pipeline.** We collect long-form videos, partition them into temporally coherent segments, generate timestamped segment-level descriptions, produce open-vocabulary queries, and construct budget-constrained extractive reference summaries by selecting query-relevant segment identifiers.

Fig. I illustrates the full dataset construction pipeline for LQVS. The construction of LQVS consists of four main stages. First, we crawl long-form videos from YouTube across diverse content domains. Second, we convert each video into a timestamped sequence of segment-level visual descriptions. Third, we generate diverse open-vocabulary candidate queries that reflect different user intents over the full video. Finally, given the segment-level descriptions and each query, we construct query-aware extractive summaries by selecting the segments that jointly satisfy the query under a fixed duration budget. This pipeline enables scalable construction of query-specific, budget-constrained summaries for long-form videos.

We first crawl long-form videos from YouTube across diverse domains, including instructional, scenery, dynamic activities, informational, and lifestyle content. We retain videos between 30 and 60 minutes to match the long-form setting targeted in this work.

For each video, we construct a timestamped sequence of segment-level visual descriptions. We sample frames at 1 FPS, extract frame-level features using CLIP ViT-Large [36], and partition the video into visually coherent temporal segments using KTS [35]. For each segment, we uniformly sample at most 64 frames, depending on its duration, and provide them to Qwen3-VL-32B-Instruct [5] to generate a caption describing the local visual content. This yields an ordered sequence of timestamped segment captions used for query-aware summary construction.

We then generate open-vocabulary candidate queries from the full-video context using the same MLLM [5]. Specifically, we provide 64 uniformly sampled frames from the entire video and prompt the model to produce diverse user queries spanning events, causes, consequences, procedures, object-centric details, temporal changes, and comparisons across different parts of the video.

Given each candidate query and the timestamped segment captions, we construct a query-aware extractive reference summary using GPT-OSS-120B [1]. The LLM selects a subset of segment identifiers whose visual content jointly satisfies the query, and the selected identifiers are mapped back to their original timestamps. We constrain the total duration of the selected segments to be at most 15% of the video duration, following the standard budget setting in video summarization [41, 11]. The prompts for both query generation and query-aware summary construction are provided in App. A.2.

To ensure reliable evaluation, we manually verify the test split after automatic annotation. Five independent annotators inspect the query, selected segment identifiers, timestamps, representative frames, and captions to assess answerability, visual grounding, and evidence sufficiency. We retain pairs by majority vote and enforce the 15% duration budget with a deterministic check. An example of the verification interface is provided in App. A.4.

We split LQVS into training, validation, and test sets at the video level with a ratio of 7:1.5:1.5, resulting in 18,198, 3,891, and 3,883 query-video pairs, respectively. The validation set is used for model selection, and all reported results are computed on the test set.

A.2 Annotation Prompts

The construction of LQVS follows a three-stage pipeline designed to scale long-form video summarization data. We provide the specific prompts used for each stage below.

Segment-level Visual Description. To obtain a dense textual representation of the video, we divide the input into temporal segments. For each temporal segment, we uniformly sample frames and provide them to the MLLM as visual input. The prompt requests a detailed description of the segment in three to five sentences.

Segment-level Visual Description Prompt

```
Describe this video segment in detail within 3 to 5 sentences.
```

Open-vocabulary Query Generation. To generate query candidates, we uniformly sample frames across the full video and provide them to the MLLM. The prompt encourages diverse information needs that reflect the overall video context, including narrative progression, emotional changes, key objects or entities, instructional value, and visual setting. This design avoids restricting queries to a predefined concept vocabulary.

Open-vocabulary Query Generation Prompt

```
You are an expert video content analyst.
The provided images are sampled uniformly from the entire duration of
a video.
Analyze the visual content, key events, objects, people, actions,
scene changes, and temporal progression across the video.

Generate 10 to 20 diverse open-vocabulary user queries that could be
answered by summarizing this video.
The queries should reflect different information needs, such as:
- important events or activities
- causes or reasons behind an event
- consequences or outcomes of an event
- procedural steps or temporal progression
- specific objects, people, or entities
- scene changes or visual transitions
- comparisons between different parts of the video

Avoid using only generic queries such as "What is this video about?"
```

```
Each query should be a concise natural-language question or request
that can be answered from the visual content of the video.

Output the result as a numbered list.
```

Query-conditioned Extractive Summary Construction. Given a query and the full set of timestamped segment-level visual descriptions, the LLM selects a subset of segment identifiers that jointly satisfies the query while respecting the summary budget. The selected identifiers are then mapped back to their original timestamps to form the extractive reference summary.

Query-conditioned Extractive Summary Construction Prompt

```
I am providing you with a user query and a list of video segment
captions, each with a specific timestamp.
Please generate an extractive summary based on the query.

Here are your instructions:
1. The summary should consist of only the segments that are most
relevant and critical to the provided [Query].
2. Do not paraphrase or reword the captions. Maintain the exact
wording from the provided list.
3. Each caption you extract must include its original timestamp.
4. IMPORTANT: The total duration of selected segments must be <=
{max_duration:.1f} seconds (approximately up to {max_percent}% of the
total video duration of {total_duration:.1f} seconds).
5. Prioritize the most relevant segments and stop before exceeding
the time budget. If only a few segments are relevant, you may return
fewer.

Query:
{query}

Total Video Duration: {total_duration:.1f} seconds
Summary Time Budget (Max): {max_duration:.1f}s

Captions with Timestamps:
{captions_with_timestamps}
```

A.3 Query Category Distribution

To evaluate the diversity and semantic breadth of the open-vocabulary queries in LQVS, we categorize them into a hierarchical taxonomy. The classification follows a human-in-the-loop procedure: first, an LLM-assisted taxonomy assignment is performed to map each query to a high-level intent and a fine-grained subcategory. Subsequently, we manually inspect the assigned labels to resolve ambiguities and merge semantically overlapping subcategories. Manual inspection consolidates the automatically assigned categories and resolves overlapping labels. The resulting taxonomy, visualized in Fig. II, comprises six high-level query intents: Events, Temporal Changes, Consequences, Object-centric, Procedures, and Causes.

As detailed in Tab. I, LQVS covers a range of information needs. *Events* account for the largest share (25.5%), followed by *Temporal Changes* (19.6%) and *Consequences* (17.5%). Alongside object-centric queries, the benchmark includes requests about changes, outcomes, and procedures, enabling evaluation of summary selection across different query intents.

A.4 Manual Verification of the Test Split

Fig. III shows the interface used for manual verification of the test split. For each query-reference pair, annotators inspect the query, selected segment identifiers, timestamps, representative frames, and



Figure II: **Query category distribution in LQVS.** The sunburst chart visualizes the hierarchical taxonomy of open-vocabulary queries. The inner ring denotes high-level intent categories, and the outer ring denotes fine-grained subcategories assigned through LLM-assisted taxonomy assignment followed by manual inspection and label merging.

Table I: **Query category statistics in LQVS.** We report the frequency (Count) and relative distribution (Dist. %) of queries across high-level intent categories and their respective fine-grained subcategories. Categories are obtained through LLM-assisted taxonomy assignment followed by manual inspection and label merging.

High-level Category	**Count**	**Dist. (%)**	**Subcategories (Count)**
Events	6,624	25.5%	activity event (4,186), story overview (1,541), interaction event (575), milestone event (322)
Temporal Changes	5,088	19.6%	before-after change (2,512), state transition (1,311), trend evolution (1,124), comparison across time (141)
Consequences	4,547	17.5%	immediate outcome (2,320), emotional effect (1,437), long-term impact (614), status result (176)
Object-centric	3,848	14.8%	object focus (1,386), place focus (962), person focus (873), attribute focus (627)
Procedures	3,129	12.0%	chronological overview (2,156), instructional steps (344), journey sequence (341), workflow process (288)
Causes	2,736	10.5%	motivation cause (1,476), context cause (1,103), trigger cause (109), problem cause (48)
Total	25,972	100.0%	24 fine-grained subcategories

segment-level captions. Five annotators independently assess query answerability, visual grounding (including caption–frame consistency), and whether the selected evidence is sufficient to answer the query. We separately check the 15% duration constraint with a deterministic rule. Majority-vote filtering removes 91 of 3,974 candidate test pairs (2.29%), leaving 3,883 verified pairs; rejected annotations are discarded without revision.



Figure III: **Example of the manual verification interface.** Annotators inspect each query-reference pair using selected segment identifiers, timestamps, representative frames, and captions to assess answerability, visual grounding, and evidence sufficiency. Budget compliance is checked automatically.

Mean pairwise agreement is 99.2%, unanimous agreement is 98.4%, and Fleiss' κ is 0.824, indicating strong agreement among annotators. This verification protocol applies to the test split; training and validation use the automatically generated annotations.

Under majority voting, query-answerability issues are flagged in 0.1% of candidate pairs, and selected-segment grounding or evidence-sufficiency issues in 2.2%; their union is the 91 rejected pairs reported above. The rejection rate is 1.6% when all five annotators must reject a pair, compared with 2.29% under the majority rule used for curation and 3.2% when any single objection is sufficient. Annotators did not assess redundancy as a separate criterion. The audit checks the validity of the provided query-reference pairs, rather than exhaustive coverage, uniqueness of the reference, or preference among alternative valid summaries. It does not establish the same acceptance rate for the unverified training and validation splits.

A.5 Cost

LQVS construction required substantial computational resources due to the scale of long-form video annotation. We deployed GPT-OSS-120B locally through Ollama on a server with 16 NVIDIA RTX A6000 GPUs for the LLM-based summary construction stage. The overall pipeline, including

video preprocessing, CLIP feature extraction, KTS segmentation, MLLM-assisted caption and query generation, LLM-based reference construction, and post-processing, took approximately two months of wall-clock time. Because all LLM inference was performed locally, no commercial API cost was incurred for GPT-OSS-120B. The dominant computational cost came from repeated MLLM/LLM inference over long videos and multiple query-video pairs.

B Discussion and Reliability of QDC

Query Disentanglement Consistency (QDC) is introduced as a complementary metric for query-aware video summarization. While standard per-query metrics evaluate the quality of each predicted summary independently, QDC evaluates whether the collection of summaries for the same video preserves the relational structure among different user intents. In this section, we clarify how QDC should be interpreted together with per-query accuracy metrics and analyze its reliability under query-count variation, tied similarity values, and test-video resampling.

B.1 QDC and Per-query Accuracy

QDC is designed to complement standard per-query metrics rather than replace them. Metrics such as F1 and ranking correlation evaluate whether each predicted summary is accurate for its corresponding query. In contrast, QDC evaluates whether multiple query-specific summaries for the same video preserve the relative similarity structure of the ground-truth summaries. We use the Jaccard similarities and Spearman correlation defined in Eq. (3) of Sec. 5.1; the corresponding heatmaps are shown in Fig. 5.

We use Spearman rank correlation because QDC is intended to measure whether the relative ordering of inter-query similarities is preserved, rather than whether their absolute magnitudes exactly match. This rank-based formulation is also appropriate for the discrete and skewed nature of Jaccard similarity values. A high QDC indicates that query pairs with more overlapping ground-truth summaries also tend to have more overlapping predicted summaries, while query pairs with distinct ground-truth summaries remain distinct in the predictions.

However, QDC does not evaluate whether each predicted summary selects the correct segments for its own query. It only compares relationships among summaries. For example, a model could produce summaries whose pairwise overlap pattern resembles the ground-truth structure, but the selected segments themselves may still be incorrect for each query. Such a model may obtain a reasonable QDC while having low F1 or ranking performance. Conversely, a model may obtain high per-query F1 by selecting relevant segments for each query, but still produce summaries that are too similar across different intents, resulting in low QDC.

This distinction explains why QDC should be interpreted together with per-query metrics. F1 measures the segment-level overlap between each predicted summary and its corresponding ground-truth summary, while ranking correlation measures whether segment priorities are aligned for each query. These metrics provide the per-query correctness signal that QDC alone does not enforce. QDC instead measures whether the collection of query-specific summaries preserves the ground-truth inter-query similarity structure. Therefore, high performance on both per-query metrics and QDC provides stronger evidence that a model generates summaries that are individually accurate and properly differentiated across queries.

Accordingly, QDC is intended as a complementary metric for query-aware summarization, especially when multiple queries are associated with the same long-form video.

B.2 Reliability Analysis of QDC

We further examine the empirical reliability of QDC. Since QDC is defined as a rank correlation over pairwise inter-query summary similarities, its behavior can depend on the number of queries associated with each video and the presence of tied similarity values. This section analyzes these factors to clarify when QDC is most informative on LQVS.

Degenerate and synthetic cases. We first consider degenerate cases that reveal structural properties of QDC. For a video with only two queries, there is only one query pair, so the rank correlation

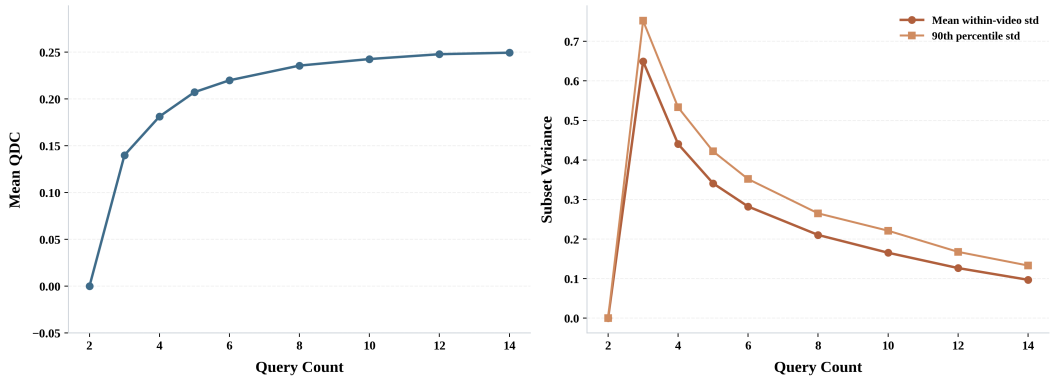


Figure IV: **Effect of query count on QDC stability.** We subsample different numbers of queries per video and compute QDC on the induced query subsets. The left panel shows mean QDC; the right panel shows the mean and 90th percentile of within-video standard deviations, as indicated in its legend. This variability is high when only a few queries are available, but decreases as the number of query pairs increases. Since LQVS test videos contain 14–20 queries, the main evaluation lies in the more stable regime.

between the ground-truth and predicted inter-query similarity vectors is undefined. Similarly, if all pairwise ground-truth similarities are identical, the ground-truth similarity vector is constant, making Spearman correlation undefined regardless of the predicted summaries. In our implementation, such cases are assigned a QDC value of zero. These cases reflect the limitation of rank-correlation-based evaluation when the query set does not induce a meaningful pairwise similarity structure, rather than a failure of the summarization model. In LQVS, all 208 test videos have non-constant ground-truth pairwise similarity vectors, with a median of 108 distinct values per video, providing the variation needed for rank-based evaluation.

We also examine tie-heavy synthetic settings, where many query pairs share the same Jaccard similarity. In one representative case, the default Spearman-based QDC is 0.847, whereas random tie breaking yields an average score of 0.673. This indicates that ties can affect the magnitude of QDC when the number of distinct similarity levels is small. Accordingly, QDC is most informative when a video contains enough queries to form a rich inter-query similarity structure.

Sensitivity to the number of queries. We next analyze how QDC stability changes with the number of queries per video. For each video, we subsample $Q' \in \{2, 3, 4, 5, 6, 8, 10, 12, 14\}$ queries, compute QDC on the induced query subset, and repeat this procedure multiple times. This measures how the variability of QDC changes as the number of available query pairs increases.

The results show that QDC is unstable when only a small number of queries is available. With 3 queries, the mean within-video standard deviation is 0.649, and with 5 queries it remains 0.341. The standard deviation decreases substantially as the number of queries increases, reaching 0.166 for 10 queries and 0.097 for 14 queries. This trend is expected because the number of query pairs grows as $Q'(Q'-1)/2$, yielding a more stable estimate of the inter-query similarity structure. In the LQVS test split, each video contains between 14 and 20 queries, placing the main evaluation in the stable regime of this analysis. Fig. IV visualizes this trend.

Effect of tied similarities. Because QDC is computed from Jaccard similarities between selected segment sets, ties naturally occur when multiple query pairs have the same degree of overlap. On the LQVS test split, the average tie ratio is 0.285 for predicted pairwise similarities and 0.347 for ground-truth pairwise similarities. To quantify the effect of ties, we perturb tied values with small random noise and recompute QDC over multiple trials.

The average absolute difference between the default QDC and the random tie-broken QDC is 0.009 across videos, indicating that tie handling has only a limited effect on the aggregate dataset-level estimate. However, the effect is larger for a small number of tie-heavy videos: five videos show an absolute shift larger than 0.05, with a maximum shift of 0.121. Thus, ties do not substantially

change the overall conclusions on LQVS, although they may affect the interpretation of individual video-level QDC scores.

Uncertainty under test-video resampling. Bootstrap resampling over test videos gives an aggregate QDC of 0.256 with a 95% confidence interval of $[0.233, 0.280]$. This interval describes uncertainty in the absolute score of one trained model under test-video resampling; it does not measure variation across training runs or the uncertainty in a difference between models. The paired comparison and training-seed replications in App. D.5 address these distinct sources of uncertainty separately.

Summary. These analyses clarify the conditions under which QDC should be interpreted. First, QDC is most meaningful when each video contains enough queries to induce a non-trivial pairwise similarity structure. Second, tied similarities can affect individual video-level scores, but their impact on the aggregate LQVS evaluation is modest. Together with per-query metrics such as F1 and ranking correlation, these results support the use of QDC as a complementary metric for evaluating whether query-specific summaries preserve the relational structure of user intents over the same long-form video.

C Implementation Details

C.1 Experimental Setup

The hyperparameters used for QLVSumm are summarized in Tab. II. We use CLIP ViT-Large features as visual inputs and Qwen3 [48] as the LLM backbone. Video frames are sampled at 1 FPS for feature extraction, and temporal segments are obtained using KTS. Unless otherwise specified, all models are trained for 5 epochs and the checkpoint with the best validation performance is used for test evaluation. At inference time, QLVSumm uses greedy budget-aware decoding with a summary budget ratio of $\rho = 0.15$. All experiments were conducted using six NVIDIA RTX A6000 GPUs, each with 48GB of memory. Training QLVSumm took approximately 2 hours per epoch under this setup.

Table II: **Hyperparameters for the proposed model.**

Category	**Hyperparameter**	**Value**
Model Architecture	Base LLM	Qwen3-8B
	Vision Input Dimension	768
	Segment Adapter Heads	12
	Segment Adapter Layers	1
LLM LoRA Configuration	LoRA Rank	8
	LoRA Alpha	16
	LoRA Dropout	0.05
Training Details	Epochs	5
	Batch Size per GPU	1
	Optimizer	AdamW
	Initial Learning Rate	1×10^{-4}
	Weight Decay	1×10^{-5}
	Scheduler	Cosine warmup
	Warmup Ratio	0.03
	Maximum Gradient Norm	1.0

C.2 Baseline Implementation

Baseline training, checkpoint selection, summary construction, and evaluation follow the protocol in Sec. 5.1.

C.3 Inference Cost

We separate costs incurred once per video from those incurred for each query. Unless otherwise specified, the stage-level measurements in Tab. III use NVIDIA RTX A6000 GPUs and approximately one-hour videos. The reported total for one video and one query is approximately 28.9 s.

Table III: **Stage-level inference costs of QLVSumm.** Costs are incurred per video or per query as indicated. KTS runs on the CPU.

Stage	Frequency	Time (s)
CLIP frame encoding (1 FPS)	Per video	15.3
KTS segmentation (CPU)	Per video	11.66
Segment Adapter + AR selection	Per query	1.95

Over 110 queries from approximately one-hour videos, QLVSumm has a median query latency of 1.95 s, a mean of 2.08 s, and a 90th percentile of 2.61 s, with peak memory of 16.77 GB. This query-time measurement includes the Segment Adapter and autoregressive selection. The median output contains 21 selected segments and requires 22 decoding steps.

Matched selection cost. Profiling the independent-scoring and full autoregressive configurations in Tab. 4 on an approximately one-hour video yields 0.222 and 1.295 s per query, respectively, with 2,692.6 and 3,042.9 GFLOPs. The autoregressive configuration therefore uses approximately 5.8× the latency and 1.13× the FLOPs, with per-query peak memory of approximately 17 GB, largely determined by the shared 8B backbone. This comparison characterizes the computational cost of history-conditioned selection alongside the accuracy comparison in App. D.5.

Video-length scaling. Tab. IV reports selection cost as video length and segment count increase. Latency increases from 0.925 to 1.326 s over the evaluated lengths, while reported peak memory remains 16.99 GB. The stage-level, matched-selection, and video-length measurements come from distinct profiling setups; their latency values should not be treated as interchangeable estimates.

Table IV: **Selection cost by video length.** Throughput is measured in queries per second.

Video length (min)	Segments	Latency (s)	Throughput	Peak memory (GB)
30.0	91	0.925	1.08	16.99
45.0	136	1.192	0.84	16.99
59.9	180	1.326	0.75	16.99

D Additional Analysis

D.1 Effect of LLM Size

Table V: **Effect of LLM backbone size on the LQVS benchmark.** All variants use the same Segment Adapter, training objective, decoding strategy, and evaluation protocol.

LLM Size	**Rank.**	**F1**	**QDC**
0.6B	0.172	22.647	0.201
1.7B	0.219	28.868	0.201
4B	0.219	29.072	0.250
8B	<u>0.226</u>	<u>30.192</u>	**0.256**
14B	**0.230**	**30.917**	<u>0.237</u>

We vary the LLM backbone size while keeping the Segment Adapter, training objective, decoding strategy, and evaluation protocol fixed. This experiment examines how backbone capacity affects

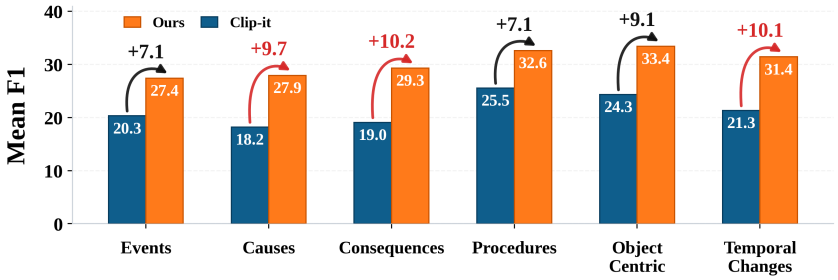


Figure V: **Category-wise F1 performance on LQVS.** QLVSumm consistently outperforms CLIP-It across major query categories, with clear gains on temporally demanding intents.

query-aware selection of temporally dispersed evidence, conditioned on the query and selection history under a duration constraint.

As shown in Tab. V, Rank and F1 generally improve as the backbone scales from 0.6B to 14B parameters. This suggests that larger LLMs provide stronger capacity for selecting query-relevant evidence across long video timelines. The 14B model achieves the highest Rank and F1, while the 8B model obtains the highest QDC. These are observations from individual training runs. Without matched replications for the 14B model or a paired uncertainty estimate for this comparison, the QDC difference does not establish a general relationship between backbone size and query disentanglement. Considering both performance and computational cost, we use the 8B backbone as the default configuration unless otherwise specified.

D.2 Query Category-wise Performance

Fig. V summarizes category-wise F1 performance on LQVS, and Fig. VI provides the full F1 and ranking distributions. We compare QLVSumm with CLIP-It, a representative query-aware baseline, across six major query categories: Events, Causes, Consequences, Procedures, Object-centric queries, and Temporal Changes. QLVSumm consistently improves both F1 and ranking correlation across all categories, showing that the proposed summary composition formulation generalizes beyond a single query type.

The improvement is particularly pronounced for queries about causes, consequences, and temporal changes. For *Causes*, *Consequences*, and *Temporal Changes*, QLVSumm improves F1 by 9.75, 10.24, and 10.12 points, respectively, over CLIP-It. Similarly, ranking correlation improves by 0.121, 0.126, and 0.124 in the same categories. These categories often require connecting evidence across distant parts of the video, such as identifying what led to an event, what changed afterward, or how a state evolved over time. The larger gains in these categories support the value of composing evidence from temporally distributed segments for query-aware summarization.

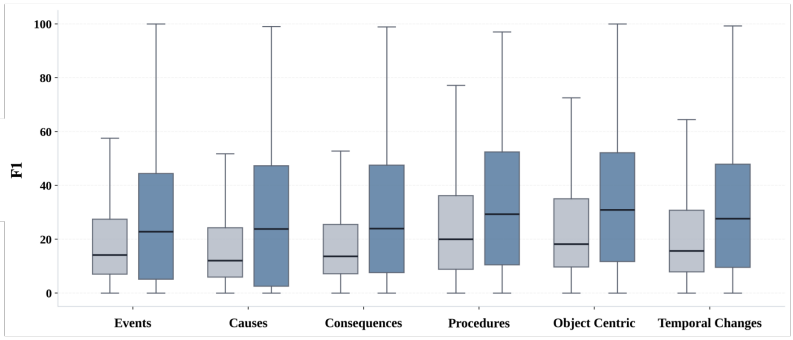
QLVSumm also improves on Events, Procedures, and Object-centric queries. Together, these results show gains across both entity-focused and temporally structured information needs, consistent with the broader aim of query-aware summary composition.

D.3 Effect of Data Scale and Query Diversity

Fig. VII examines how the scale and structure of training data affect QLVSumm. In Fig. VIIa, Rank and F1 peak at 75% of the training data, while QDC is highest with the full training set. The full-data setting thus yields the strongest cross-query consistency among the evaluated training-data fractions. Repeated runs of the full-data model are reported in App. D.5, but no corresponding replications are available for the 75% setting; the observed local decrease in Rank and F1 therefore cannot be attributed to training variation or a systematic data-scale effect.

Fig. VIIb varies the number of training queries per video while keeping the video set fixed. A single query per video provides query-conditioned supervision across videos, while additional queries expose the model to different intents over the same visual context. QDC improves more clearly than Rank and F1, consistent with the value of this supervision for learning intent-dependent selections. Increasing the number of queries expands both intent coverage and the amount of query–summary

(a) F1



(b) Ranking

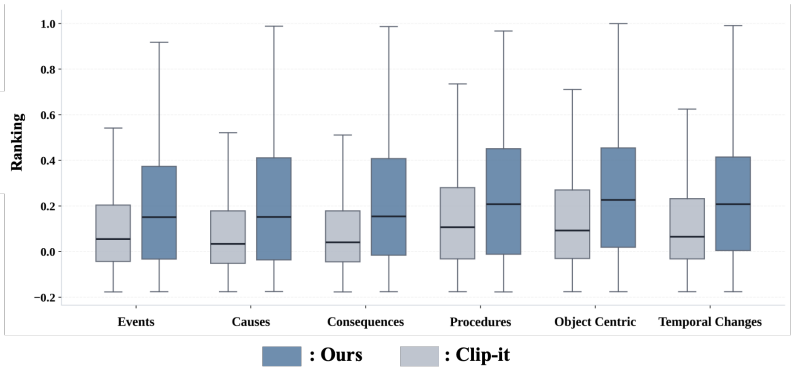


Figure VI: **Query category-wise performance distributions on LQVS.** Each pair of boxplots compares CLIP-It and QLVSumm across major query categories. The boxes indicate the interquartile range, the horizontal line denotes the median, and whiskers show the overall spread excluding outliers. QLVSumm consistently improves both (a) F1 and (b) ranking performance across diverse query intents.

supervision in this experiment. The model learns from individual query–summary pairs, without an explicit pairwise or QDC training objective.

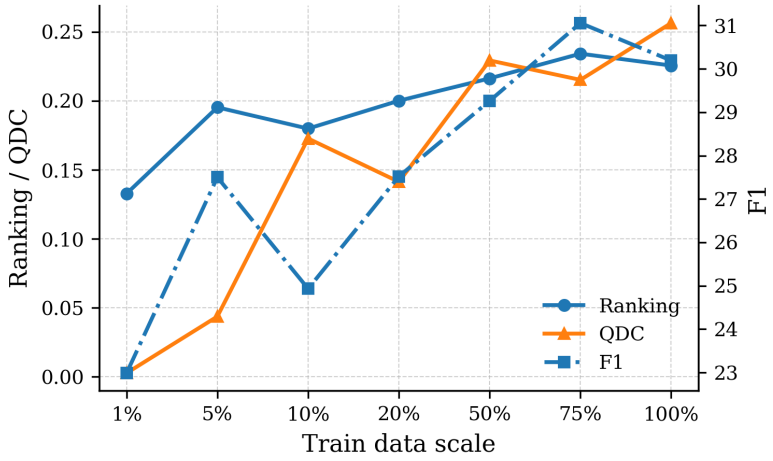
D.4 Evaluation on S-VideoXum

We further evaluate QLVSumm on S-VideoXum [32], a script-driven video summarization benchmark with 11,908 videos and ten independently collected human reference summaries per video. Conditioning scripts are generated by a separate multimodal model. Videos average approximately 2.5 minutes and reach approximately 12.5 minutes, providing a shorter-video setting alongside LQVS. We use frame-level features as input and produce frame-level summary outputs in the same format as the baselines, with KTS omitted and the Segment Adapter disabled. We evaluate using the native frame-level F-score protocol.

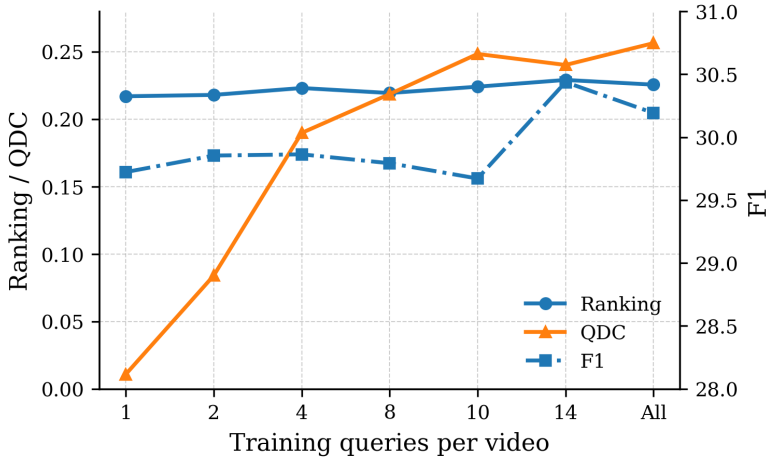
Results are reported in Tab. 3, with baseline scores quoted from Mylonas et al. [32]. We train and evaluate QLVSumm using S-VideoXum's human summary targets under this native configuration. This evaluates the summary selector on shorter videos with independently collected human targets; it does not directly evaluate the long-video compression component or replace independent human re-annotation of LQVS.

D.5 Statistical Reliability of the Capacity-matched Comparison

We evaluate the reliability of the difference between the independent-scoring configuration and full QLVSumm, corresponding to rows (b) and (e) of Tab. 4. Both configurations use the same CLIP features, Segment Adapter, Qwen3-8B backbone, LoRA parameter budget, training examples and reference summaries, summary budget, and evaluation protocol. The independent scorer predicts



(a) Training data scale.



(b) Training queries per video.

Figure VII: **Effect of training data scale and query diversity.** (a) Ranking, F1, and QDC at different training-data fractions. (b) Increasing training queries per video yields clearer gains in QDC than in per-query metrics.

segment scores and uses knapsack selection; the full model uses history-conditioned autoregressive selection with set-aware supervision and rollout conditioning. Their objectives follow their respective output formulations, so this comparison evaluates the complete selection formulation rather than autoregression alone.

Paired uncertainty over test videos. We use 10,000 paired bootstrap resamples of test videos to estimate 95% confidence intervals for the full-model minus independent-scoring differences, and separately perform two-sided video-level sign-flip tests. The intervals in Tab. VI exclude zero for all three metrics. These paired intervals quantify finite-test-set uncertainty in differences between models; the absolute-score QDC interval in App. B.2 estimates a different quantity.

Replications across training seeds. Both configurations were retrained with three matched random seeds. Tab. VII reports the mean and standard deviation across these runs. The seed-wise gains are $0.097/0.091/0.101$ for Rank, $7.70/7.24/8.09$ for F1, and $0.035/0.047/0.032$ for QDC. All matched differences are positive. These replications describe training-run variation separately from test-video resampling; they do not provide repeated-run evidence for the 14B backbone or the 75%-data setting.

Table VI: **Paired uncertainty for the capacity-matched comparison.** Differences are full QLV-Summ minus independent scoring, using the configurations in rows (e) and (b) of Tab. 4.

Metric	95% paired-bootstrap CI	Two-sided sign-flip p
Rank	$[0.085,\ 0.104]$	0.0001
F1	$[6.58,\ 8.30]$	0.0001
QDC	$[0.018,\ 0.058]$	0.0006

Table VII: **Three-seed replications of the capacity-matched models.** Values are mean $\pm$ standard deviation across three matched training seeds.

Model	Rank	F1	QDC
Independent scoring	0.1288 ± 0.0009	22.55 ± 0.07	0.2215 ± 0.0021
Full QLVSumm	0.2253 ± 0.0041	30.22 ± 0.37	0.2595 ± 0.0100

D.6 Shared-content Coverage of Generic Baselines

To examine how query-agnostic summaries obtain nontrivial scores on LQVS, we analyze the relationship between baseline outputs and content shared across query-specific references. Among positive query-segment annotations, 85.2% involve segments selected for at least two queries from the same video. The segment scores of CSTA [40] and TripleSumm [23] have Spearman correlations of 0.50 and 0.52, respectively, with ground-truth segment prevalence across queries. Their recall on segments selected by at least 25% of a video's queries is 0.62 and 0.60, respectively. These observations explain how fixed summaries can receive credit by covering shared content, without establishing an architectural cause or demonstrating query adaptation. As specified in Sec. 5.1, each fixed summary is evaluated against every corresponding query-specific reference and the scores are averaged, with no best-reference selection. Query-aware methods can obtain the same shared-content credit while additionally adapting to query-specific evidence.

E Limitations & Broader Impacts

E.1 Limitations

Although QLVSumm demonstrates strong performance on long-form QVS, several limitations remain. First, LQVS is constructed through an MLLM-assisted annotation pipeline. Fully human-authoring query-specific summaries for long-form videos is prohibitively expensive, as it requires annotators to inspect long videos, design diverse user intents, and select budget-constrained segments for each query. Our pipeline provides a scalable alternative by generating queries and references from timestamped segment-level visual descriptions, followed by manual verification of the test split. Nevertheless, the generated annotations may still inherit errors or biases from the underlying MLLM and LLM, and future work could further improve reliability through larger-scale human annotation or multi-annotator validation.

Second, QLVSumm produces extractive summaries over predefined temporal segments. This design preserves the visual grounding of the output and enables explicit budget control, but the summary quality can depend on the segmentation granularity. If an important event occurs within a short sub-interval of a segment, the model may be forced to select a longer segment than necessary. Future work could explore finer-grained temporal refinement or hierarchical segment selection to improve temporal precision.

Third, QLVSumm relies on an LLM-based autoregressive decoder, which is more computationally expensive than lightweight segment-scoring models. Although the Segment Adapter reduces the input length by compressing long videos into compact segment-level representations, inference still requires sequential decoding over video-specific segment identifiers. Improving decoding efficiency and reducing the computational cost of long-form query-aware summarization remain important directions for future work.

Finally, our benchmark focuses on visual query-aware summarization and does not explicitly incorporate audio, speech, or external contextual information. This design makes the reference summaries visually grounded, but some real-world user queries may require multimodal cues beyond visual content alone. Extending QLVSumm to jointly reason over visual, audio, and textual modalities could further broaden its applicability.

E.2 Safeguards for responsible release

If the paper is accepted, we plan to release LQVS with documentation describing its intended research use, annotation pipeline, limitations, and known risks. We will avoid redistributing raw video content directly and instead release annotations, metadata, timestamps, and derived resources only when permitted, while respecting the terms of the original video sources. We will also exclude videos or annotations that are found to contain sensitive, private, or unsafe content during release preparation, and provide usage guidelines discouraging privacy-invasive or misleading applications.

E.3 Broader Impacts

QLVSumm and LQVS are intended to support user-centered access to long-form video content. Query-aware summarization can help users efficiently navigate educational, instructional, sports, documentary, and other long-form videos by producing summaries tailored to different information needs. The benchmark may also facilitate research on long-form video understanding, controllable summarization, and intent-conditioned evaluation.

Potential negative impacts include privacy risks when summarizing sensitive videos, as well as biased or misleading summaries caused by errors in the underlying models or training data. Because summaries can shape user interpretation of video content, incorrect segment selection may omit important context or overemphasize particular events. We partially mitigate these concerns through extractive summarization, which keeps outputs grounded in original video segments, and through manual verification of the test split. Future deployment should incorporate privacy-aware data handling, transparent uncertainty reporting, and human oversight for sensitive applications.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope. The paper claims to formulate long-form QVS as query-aware summary composition, introduce the LQVS benchmark and QDC metric, and demonstrate improved performance; these claims are supported by the method description in Sec. 3, the benchmark and evaluation setup in Sec. 5, and the quantitative results in Tab. 2.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations of the proposed framework in App. E.1. In particular, it discusses limitations arising from the MLLM-assisted annotation pipeline, dependence on predefined temporal segmentation, the computational cost of LLM-based autoregressive decoding, and the visual-only scope of the benchmark.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The mathematical expressions in the paper define the task formulation, decoding procedure, training objective, and evaluation metric rather than theoretical claims requiring proof.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides the information needed to reproduce the main experimental results, including the model formulation in Sec. 3, the LQVS benchmark construction and evaluation protocol in Sec. 5, and implementation details and hyperparameters in App. C. It also specifies the dataset splits, evaluation metrics, baseline training and evaluation procedures, summary budget, and decoding strategy.

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [No]

   Justification: The code and LQVS benchmark are not publicly released at submission time. If the paper is accepted, we plan to release the dataset, code, preprocessing scripts, and instructions for reproducing the main experimental results upon publication.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the experimental settings needed to understand the results, including the LQVS train/validation/test splits, model selection protocol, evaluation metrics, baseline evaluation setup, summary budget, and decoding strategy in Sec. 5. Additional implementation details, including hyperparameters and optimizer settings, are provided in App. C.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

Justification: The paper reports quantitative results on the LQVS test set, but does not currently report error bars, confidence intervals, or statistical significance tests. Since all experiments are conducted with a single random seed, we ensure fair comparison by using the same dataset split, metrics, summary budget, and evaluation protocol for all methods; quantifying variance across random seeds is left for future work.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports the compute resources used for the main experiments in App. C, including the GPU type, number of GPUs, GPU memory, and approximate training time per epoch. It also reports the additional compute used for constructing LQVS in App. A.5.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and conducted the research in accordance with it. The paper uses publicly available videos and model-assisted annotations for benchmark construction, does not involve human-subject experiments or collection of private sensitive information, and preserves submission anonymity.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses broader impacts in App. E.3. It describes positive impacts for user-centered access to long-form videos and research on long-form video understanding, as well as negative impacts related to privacy, biased or misleading summaries, omitted context, and sensitive applications.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [Yes]

    Justification: The paper describes safeguards for the responsible release of LQVS in App. E.2. The planned release will include documentation of intended use, limitations, and risks, avoid direct redistribution of raw video content when not permitted, exclude sensitive or unsafe content identified during release preparation, and provide usage guidelines discouraging privacy-invasive or misleading applications.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: The paper credits the original works for the existing models, tools, datasets, and baselines used in the study, but does not currently specify the corresponding licenses and terms of use. We will document the licenses and usage terms of these assets before releasing the code and LQVS benchmark.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces LQVS as a new benchmark and provides detailed documentation of its construction, annotation pipeline, prompts, splits, verification protocol, and limitations in Sec. 5, App. E.1 and App. A. The released version, upon acceptance, will include accompanying documentation and usage guidelines.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects. The manual verification of the LQVS test split was conducted as an internal data quality-control step rather than a human-subject study.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects. The manual verification of the LQVS test split was conducted as an internal data quality-control step rather than a human-subject study, so IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: The paper describes the use of LLMs as part of the core methodology and benchmark construction. Specifically, QLVSumm uses an LLM-based autoregressive decoder for query-aware segment selection, and LQVS is constructed using MLLM/LLM-assisted captioning, query generation, and reference summary construction, as described in Sec. 3, Sec. 5, and App. A.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.