TCVP: A Practical Pipeline for Video Moment Retrieval Datasets Leveraging Timestamped Video Comments

Anonymous ACL submission

Abstract

Video Moment Retrieval (VMR) aims to identify a temporal moment in a video that corresponds to a user query. Most existing VMR datasets are constructed by randomly selecting temporal moments and generating queries from the corresponding visual and auditory content. We find that this process often produces moments with limited importance and queries that resemble captions rather than user-driven searches. To address these limitations, we propose a practical pipeline for VMR dataset generation, named TCVP, where we leverage timestamped YouTube comments to identify interesting moments and reflect actual search intent. A naive use of YouTube comments introduces several challenges, as many comments are uninformative (*e.g.*, "07:22 lol"), and comments may correspond to different modalities, requiring modality-aware handling. Our pipeline alleviates them by introducing comment filtering and modality gating as key methodological components. Our qualitative analysis shows that users prefer our dataset by a substantial margin (*i.e.*, 70%) over existing baselines. Moreover, benchmarking models on our dataset highlights limitations of current VMR methods and offers insights for future work.

1 Introduction

Given a natural language query, the goal of Video Moment Retrieval (VMR) is to locate the corresponding temporal moments within a video (Lei et al., 2021a; Moon et al., 2023; Ren et al., 2024; Pan et al., 2025). This capability is useful in diverse scenarios, such as assisting video editors in identifying salient moments from lengthy recordings (Huh et al., 2025; Croitoru et al., 2023), or enabling Netflix users to search for specific scenes they wish to revisit in previously watched videos (Chen et al., 2023). By reducing the need for manual exploration, VMR can serve as a practical tool in user-facing applications.

To enable the development and evaluation of VMR models, existing studies (Gao et al., 2017; Lei et al., 2020, 2021a; Lin et al., 2023; Soldan et al., 2022) have introduced VMR datasets. Figure 1 (red boxes) illustrates a representative data generation pipeline used in these works. As shown, this pipeline often yields moments that are weakly aligned with real user interest and queries that are verbose and descriptive rather than search-oriented. We *identify* this mismatch as a fundamental limitation of existing VMR datasets, motivating the need for datasets that better reflect real user intent.

To address this gap, we present a new perspective by leveraging timestamped YouTube comments. These comments naturally indicate which moments users consider important and how they refer to them in practice. Based on this observation, we propose a simple-yet-effective dataset construction pipeline, termed the **T**imestamped **C**omment-guided **V**MR **P**ipeline (**TCVP**). Directly using the comments poses challenges, as many are uninformative and user reactions may focus on different modalities, which necessitates modality-specific data construction procedures. For example, some comments express general reactions or emotions, while others target specific either visual events or audio content. Hence, TCVP incorporates comment filtering and modality gating. Comment filtering removes irrelevant or weakly grounded comments, while modality gating assigns the remaining ones to either visual or auditory cues to ensure proper grounding of each moment–query pair.

To validate the efficacy of the proposed pipeline, we present extensive qualitative comparisons that demonstrate the naturalness of the generated moments. We further perform quantitative human evaluations, which show a strong user preference for our dataset compared to those constructed using standard methodologies. Finally, we benchmark

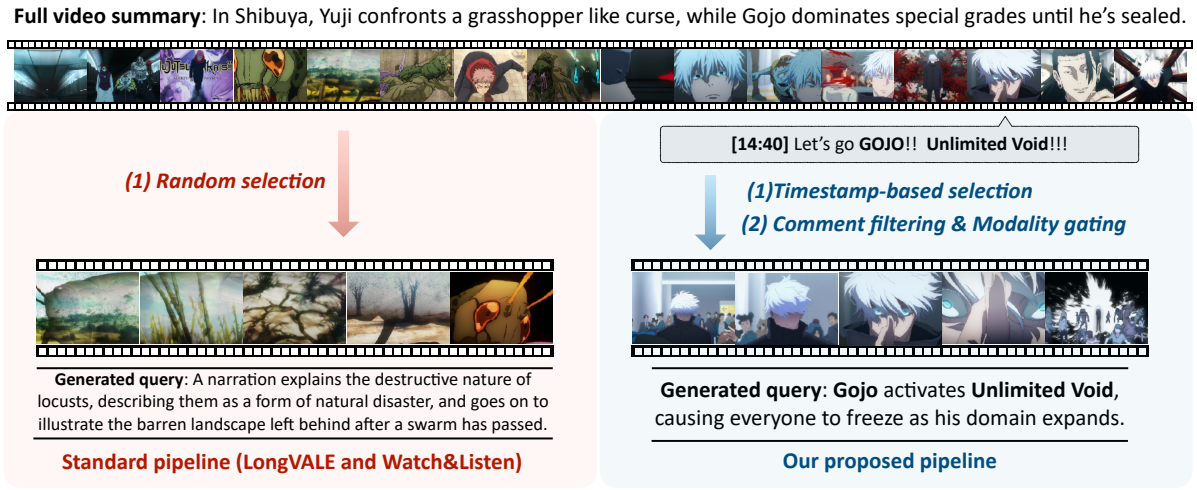

Figure 1: Comparison between our comment based pipeline and a standard pipeline. Our method selects moments using timestamped user comments with modality-aware filtering to generate natural, whereas standard pipelines selects effectively random moments.

a range of existing VMR models on our dataset, revealing limitations of current approaches and offering insights that may guide future research.

Our contributions are summarized as follows:

- We identify limitations of existing VMR datasets, including less meaningful moments and caption-like queries.
- We introduce a novel perspective on VMR dataset construction by leveraging timestamped YouTube comments.
- We propose a simple and effective dataset construction pipeline, TCVP, that incorporates comment filtering and modality gating.
- We conduct extensive qualitative and quantitative evaluations, as well as model benchmarking.

2 VMR: Where do we stand?

Existing work on VMR datasets can be divided into human-annotation–based and machine-generated approaches. Human-annotated datasets (Gao et al., 2017; Lei et al., 2020; Yuan et al., 2025) are constructed by asking annotators to *briefly* watch videos, identify *seemingly* interesting moments, and compose search queries describing those moments. Specifically, QVHighlights (Lei et al., 2021b) is constructed by asking annotators to write a query for each video and mark the matching time spans and rate how important each clip is for that query. Machine-generated datasets typically segment videos into multiple clips according to abrupt changes in visual or audio content, treating all resulting segments as sampled (which we regard as effectively *random*). Captions are then generated for each clip to serve as user queries, with recent work such as Watch&Listen leveraging large language models (LLMs) (Geng et al., 2025; Li et al., 2025).

Current Limitations Despite such advances in VMR datasets, we identify several limitations in existing pipelines. In human-annotated settings (Gao et al., 2017; Lei et al., 2021b; Yuan et al., 2025), annotators are typically constrained by limited time budgets, which hinders thorough viewing of a video and the selection of moments corresponding to globally important or truly salient events. In machine-generated approaches (Geng et al., 2025; Li et al., 2025), most segments from a long video are adopted regardless of their importance. As a result, the moments in the dataset may differ from those that real viewers would want to retrieve. Furthermore, the search queries are produced without access to prior knowledge about real user intent. This straightforward generation process often results in queries that are overly descriptive, as shown in Figure 1. Finally, several studies (Yuan et al., 2025) rely exclusively on visual information. However, an analysis of YouTube comments in YT-CommentQA (Yang et al., 2024) reveals that about half of user comments refer to audio content. VMR datasets built solely on visual cues therefore fail to capture this substantial portion of user behavior.

Such datasets may not be beneficial for improving the VMR ability during model training and may not be optimal for reliable model evaluation. This highlights the need for new ideas and pipelines.

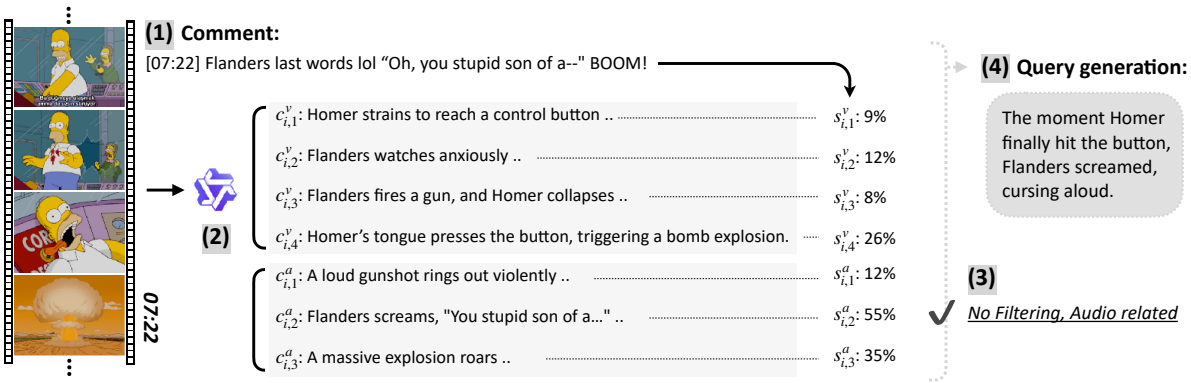

Figure 2: TCVP overview of comment filtering & modality gating and query generation. TCVP computes comment–caption similarity scores and applies modality gating and then generates a query for the target moment.

3 TCVP: Timestamped Comment-guided VMR Pipeline

We present a novel idea to leverage timestamped YouTube comments, which contain both an important moment and real user intent. A naive use of such comments, however, introduces several challenges. First, many comments are uninformative—for example, generic reactions such as “lol" or “this is amazing,". Second, user comments may rely on different modalities, such as visual or audio information, and these comments require distinct data construction procedures. To address these issues, we propose a realistic VMR dataset generation pipeline, named **TCVP**. Figure 2 illustrates the pipeline, which proceeds through (1) videos and comments collection, (2) modality specific captioning, (3) comment filtering & modality gating, and (4) query generation.

3.1 Collecting videos and comments

We collect videos from Youtube channels listed in Table 1, focusing on channels with more than one million subscribers. Such channels host contents that are widely viewed and contain sufficient user comments. For each video, we crawl all comments and retain only those that include timestamps. We then sort these timestamped comments by the number of likes, as likes indicate that many viewers regard the moment as important. For each video, we keep the top 20 timestamped comments and use them in the next steps.

3.2 Modality specific captioning

For each timestamped comment u_i, we perform modality-specific captioning. Specifically, we prompt Qwen2.5-Omni (Xu et al., 2025) to generate short sentences of at most 20 words, producing both visual and audio captions for a 9-second window before and after the timestamp. We denote the resulting caption sets as $c^v_{i,k}$ and $c^a_{i,k}$, respectively, as illustrated in Figure 2. The motivation for short sentences is discussed in the following section.

Table 1: Categories and channels used in our dataset.

Category	Channels
Knowledge / Education	TED, BigThink, Kurzgesagt, Veritasium, Vsauce
Science / Analysis	RealLifeLore, WendoverProductions, Vox
Documentary	NatGeo
Podcast (Long-form)	HubermanLab, LexFridman, joerogan, TimFerriss
Debate	OxfordUnion
Political Commentary	LastWeekTonight, TheDailyShow
News	PhilipDeFranco, TheYoungTurks, DWNews
Talk Show	LateNightSeth, JimmyKimmelLive, OfficialGrahamNorton, fallontonight
Variety / Entertainment	MrBeast, TopGear
Travel / Lifestyle	KaraandNate, AbroadinJapan, YesTheory
Culture / Society	Jolly, AsianBoss, GeographyNow
Cooking / Food	bingingwithbabish, bonappetit, aragusea
Making / Engineering	MarkRober, SmarterEveryDay, primitivedechnology9550, Corridor
Technology / Tech Review	TechLead, mkbhd, LinusTechTips
Art / Design	ProkoTV, theartassignment
Fashion	bestdressed, HauteLeMode, Vogue
Sports	NBA, fifa, Olympics
Gaming	PewDiePie, markiplier, pokimane, LoLEsports
Comedy / Sketch	SaturdayNightLive, KeyandPeele

3.3 Comment filtering & Modality gating

As noted above, directly using the collected timestamped comments can introduce several issues. To mitigate these issues, we first introduce a comment filtering mechanism to reduce vague or uninformative comments. To determine whether a comment contains information relevant to the video content, for each timestamped comment, we compute similarity scores with both visual and audio caption sentences generated in the surrounding temporal window. The similarity between the comment u_i

Table 2: Comparison of existing video moment retrieval datasets and our dataset. Real-world indicates that queries are tied to user behavior rather than controlled annotation protocols.

Dataset	Real-world	Visual Query	Audio Query	Dur. (s)	#Videos	#Queries	Domain
Moment Retrieval							
CharadesSTA (Gao et al., 2017)	✗	✓	✗	30.6	1334	3.7k	Activity
QVHighlights (Lei et al., 2021b)	✗	✓	✗	150	476	1.5k	Vlog / News
TVR (Lei et al., 2020)	✗	✓	✗	76.2	1090	5.5k	TV show
Ego4D-NLQ (Grauman et al., 2022)	✗	✓	✗	493.7	333	4k	Egocentric
MomentSeeker (Yuan et al., 2025)	✗	✓	✗	1201.9	268	1.8k	Open
LongVALE (Geng et al., 2025)	✗	✓	✓	235	8411	105k	Open
WavCaps (Mei et al., 2024)	✗	✗	✓	67.6	400k	400k	Open
UnAV-100 (Geng et al., 2023)	✗	✗	✓	42.1	10k	30k	Open
Ours	✓	✓	✓	1517.5	300	4.5k	YouTube

and the k-th visual or audio caption sentence is defined as:

$$s^v_{i,k} = \text{Sim!}\left(u_i, c^v_{i,k}\right), \quad s^a_{i,k} = \text{Sim!}\left(u_i, c^a_{i,k}\right), \quad (1)$$

where $\text{Sim}(\cdot,\cdot)$ denotes the cosine similarity computed in the embedding space from the Qwen-3 embedding model. The maximum similarity for each modality can refer to:

$$s^v_i = \max_k s^v_{i,k}, \quad s^a_i = \max_k s^a_{i,k}. \quad (2)$$

If $\max(s^v_i, s^a_i)$ is below a predefined threshold τ, we label the comment as unrelated and discard it. In our experiments, we set $\tau = 0.3$. We observe that overly long caption sentences $c^v_{i,k}$ or $c^a_{i,k}$ tend to yield low similarity scores, as the collected comments are typically short (around 20 words), and we mitigate this issue by encouraging concise captions in the captioning stage.

After comment filtering, we assign each comment to either the visual or audio modality. Using the maximum similarity scores computed for visual and audio captions, we determine the dominant modality for each comment. Formally, the modality assignment is defined as:

$$m_i = \begin{cases} \text{unrelated} & \text{if } \max(s^v_i, s^a_i) < \tau, \\ \text{vision-related} & \text{if } s^v_i \geq s^a_i, \\ \text{audio-related} & \text{if } s^a_i > s^v_i. \end{cases} \quad (3)$$

This modality gating step enables subsequent stages to apply modality-specific processing for moment grounding and query generation.

Figure 2 shows an example of modality gating. The comment at 07:22 contains a quoted line and yields a higher similarity score with audio captions (55%) than with visual captions (26%), and is therefore assigned to the audio modality and passed to the next stage.

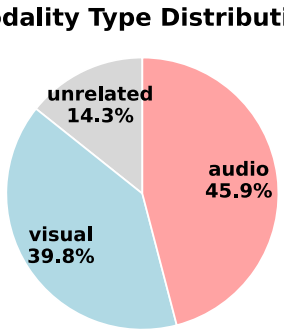

Figure 3: Dataset statistics for modality. The pie chart shows the overall split of visual-related, audio-related, and unrelated comments.

3.4 User query generation

In this step, we generate user queries q_i by leveraging the information collected in the previous stages. We prompt GPT-4.1 (OpenAI, 2025) with the timestamped comment and the modality-specific captions to produce a q_i. Unlike prior approaches, we explicitly instruct the GPT to referring the underlying user intent in user comments and extracting keywords from the comment. The q_i is then formulated based on these keywords to resemble a natural search query rather than a descriptive caption. Moreover, query generation is conditioned on the assigned modality, using visual captions for vision-related comments and audio or speech captions for audio-related comments. This prevents the model from relying on mismatched evidence and helps generate queries grounded in the correct modality.

We will release our dataset, which includes our code, video links, collected comments, modality labels, and user-like queries. The prompts used in our pipeline are provided in the appendix A. A comparison between our dataset and existing VMR datasets is provided in Table 2.

Sound: A male voice jokingly says, "Guys, John Stamos has a gross belly button," followed by laughter from the audience, then it suddenly goes quiet.

Comment: **1:43** Love how the crowd laughed at the ***John Stamos*** gross ***belly button*** comment but then went deathly ***silent*** when they saw how gross it actually is 😂

Generated Queries

LongVALE (2025)	A Twitter overlay pops up with a belly-button tweet about John Stamos. Right after the punchline, the man in the blue blazer raises his shirt, revealing his belly button as the crowd erupts.
Watch &Listen (2025)	You hear crude punchlines, including "Ted Danson sucks donkey balls he's right," followed by another voice saying, "guys John Stamos has a gross belly button," triggering louder laughter and applause. A man in a blue blazer then lifts his shirt and shows his belly button, visually punctuating the joke.
Ours w/o com.	comedian reads a tweet about John Stamos' gross belly button and the audience explodes laughing
Ours	the part where the crowd laughs at the ***John Stamos*** gross ***belly button*** joke and then **goes silent** when he shows it.

Sound: A single male voice speaks continuously in a clear, explanatory tone, with no background music or sound effects.

Comment: "1:28" Sorry sir, why this ***dollar is torn into 5 pieces***? Its 2 dollars, btch.

Generated Queries

LongVALE (2025)	A speaker in an orange shirt explains Banach Tarski while holding a dollar bill in both hands. His gestures track the explanation, and after a moment with no speech, he suddenly tears the bill into five pieces the sound of each rip aligns with the visible tearing.
Watch &Listen (2025)	A man with glasses and a beard stands in front of a black backdrop wearing an orange shirt. While speaking, he explains the Banach Tarski paradox and the idea of splitting an object into parts, using a dollar bill as a prop. He gestures with his hands, pauses briefly in silence, and then tears the bill into five pieces ...
Ours w/o com.	the scene where the man with glasses raises a dollar bill and gestures while speaking
Ours	the part where the man tears the ***dollar bill into five pieces*** while explaining.

Figure 4: Qualitative comparison of queries generated for the same moments. Each example shows sampled frames, an audio summary, and the anchor timestamped comment, followed by queries from caption-based baselines (LongVALE, Watch&Listen) and our variants (ours w/o comments and ours w/ comments).

4 Analysis of TCVP

In this section, we analyze our dataset generated by TCVP. We examine its modality statistics and conduct a qualitative comparison of both moments and user queries against existing VMR datasets, followed by human evaluation results. Detailed evaluation protocols are provided in Appendix B.

4.1 Results of our filtering & gating

Figure 3 reports the results after applying the third stage of our pipeline. Overall, 45.9% of comments are labeled as audio-related and 39.8% as vision-related, while 14.3% are filtered out as unrelated. The results show that many comments point to audio or speech cues, aligning with findings from YT-CommentQA (Yang et al., 2024) discussed in Section 2. This underscores the importance of audio modeling for realistic VMR datasets.

4.2 Qualitative comparison of user queries

An example in Figure 4 shows a timestamped comment querying the reason a dollar bill is torn into five pieces, with the corresponding clip depicting the speaker explaining the action. In LongVALE and Watch&Listen, the query is driven by broad clip descriptions, remaining tied to the lecture context and incorporating peripheral details (e.g., the Banach–Tarski explanation). Without the comment (*i.e.*, ours w/o com.), the query stays at a generic description such as holding the bill and speaking, so the key action that viewers want to retrieve is not singled out. In contrast, conditioning on the timestamped comment (*i.e.*, ours) yields a query directly targets the bill being torn into five pieces, closely reflecting the underlying user intent.

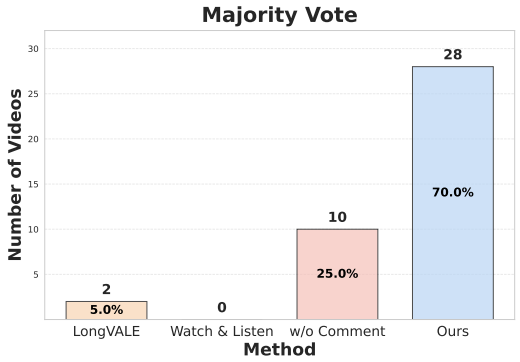

Figure 5: Human evaluation of generated queries. We compare caption-based baselines (LongVALE, Watch&Listen) with our dataset.

4.3 Human evaluation: are our queries realistic?

To support the qualitative comparison, we conduct a human evaluation study that assesses the naturalness and realism of the queries. The instruction can found in Figure 8. Annotators see four candidates from LongVALE, Watch&Listen, Ours w/o comments, and Ours w/ comments, and they select the best one, with three independent judgments aggregated by majority vote. Figure 5 shows that **Ours w/ comments** is preferred in 70% of cases and **Ours w/o comments** in 25%, while LongVALE ac-

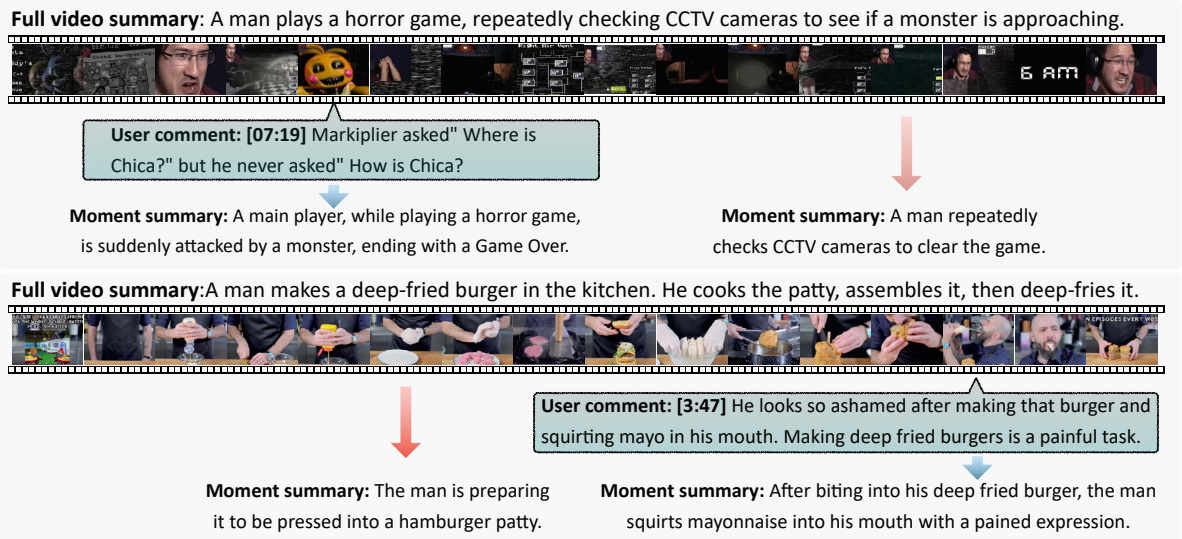

Figure 6: Qualitative comparison of moment selection within the same videos. Comment-based selection (timestamped user comments) tends to capture eventful segments that attract user reactions, whereas random sampling often yields ordinary segments with weak retrieval cues.

counts for 5% and Watch&Listen is never selected. This result indicates that leveraging timestamped comments improves both the naturalness and realism of the generated queries.

4.4 Qualitative comparison of moment selection

Figure 6 compares moments selected at comment timestamps with random selection from the same videos. In the horror gameplay video shown in Figure 6, the comment-aligned moment captures a sudden monster attack that results in a game over, whereas the randomly selected moment (*e.g.*, red arrow) depicts the player checking CCTV cameras, which is less relevant to viewer interest. In the cooking video, the random moment captures a routine preparation action (*e.g.*, pressing a patty) with limited distinctive retrieval cues. We further present a human evaluation on moment selection in the following subsection.

4.5 Human evaluation: Timestamped moment vs. randomly selected moment

For each video, annotators compare two candidate moments, one from timestamped comments and one from random sampling, and choose the moment that is more likely to be searched to retrieve. Judgments from three independent annotators are aggregated by majority vote. Figure 7 shows that annotators prefer comment-based selection in 95% of videos, while random sampling is preferred in 5%. These results demonstrate that timestamped

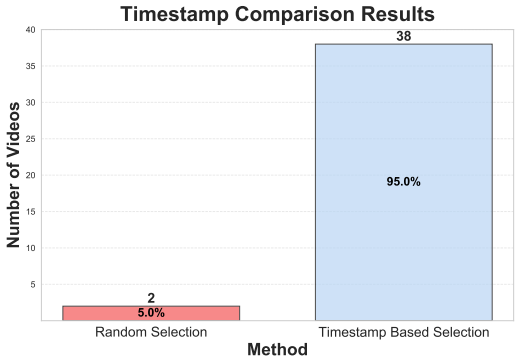

Figure 7: Human evaluation of moment selection. Annotators choose the moment that is more likely to be searched for retrieval between comment-based selection and random sampling.

comments are effective signals for identifying moments that align with user search intent.

5 Evaluation of Models on Our Dataset

In the previous sections, we presented a pipeline for constructing VMR datasets. This section evaluates VMR models on the resulting dataset, where models retrieve specific moments in a video given a user search query, reflecting real-world applications such as Netflix and Adobe Premiere Pro.

5.1 Experimental Setting

Task and metrics. VMR models retrieve a target moment given various input modalities, including visual frames (**V**), audio signals (**A**), and **ASR** transcripts (*i.e.*, subtitles). In our dataset construction, moment–query pairs are explicitly categorized into vision-related and audio-related subsets. Leverag-

Table 3: Comparison of VMR performance across models and input modality configurations, evaluated using Recall@1, Recall@5, and Recall@10. Avg(R@10) reports Recall@10 on the *ALL* split.

Model	Size	Input modality			Avg	Vision			Audio		
		V	A	ASR	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Random	–	–	–	–	17.6	1.8	9.3	20.1	1.3	8.4	16.4
Closed-source MLLM											
Gemini 2.5 Flash (Comanici et al., 2025)	–	✓	✗	✗	–	6.0	–	–	4.0	–	–
+ audio	–	✓	✓	✗	–	6.0	–	–	6.0	–	–
+ Sub.	–	✓	✓	✓	–	26.0	–	–	26.0	–	–
Open-source MLLMs											
Qwen2.5VL (Bai et al., 2025)	3B	✓	✗	✗	–	6.1	–	–	4.6	–	–
InternVL3 (Zhu et al., 2025)	2B	✓	✗	✗	–	3.7	–	–	2.5	–	–
Qwen2.5-Omni (Xu et al., 2025)	3B	✓	✓	✗	–	5.2	–	–	3.7	–	–
Open-source MLLMs with Segment Captions											
Qwen2.5VL	3B	✓	✗	✗	26.2	5.8	18.7	32.0	2.6	11.8	21.9
+ Sub.	3B	✓	✗	✓	39.8	8.7	23.7	37.2	20.0	37.3	47.3
InternVL3	2B	✓	✗	✗	26.9	7.4	20.5	33.0	3.2	12.2	23.1
+ Sub.	2B	✓	✗	✓	39.9	10.1	25.7	37.3	19.9	37.6	47.6
Qwen2.5-Omni	3B	✓	✓	✗	30.8	9.2	25.2	37.6	5.9	17.2	27.5
+ Sub.	3B	✓	✓	✓	43.8	11.1	28.5	41.8	19.2	38.2	48.2
Embedding Models											
CLIP (Radford et al., 2021)	151M	✓	✗	✗	43.9	14.9	40.8	54.3	6.4	22.1	34.4
LanguageBind (Zhu et al., 2024)	428M	✓	✗	✗	49.0	30.5	50.0	62.7	14.2	26.3	37.0
InternVideo2 (Wang et al., 2024)	1B	✓	✗	✗	43.6	16.5	43.0	57.9	5.5	19.2	31.5
CLAP (Elizalde et al., 2023)	154M	✗	✓	✗	25.7	4.1	15.7	27.3	2.3	12.7	24.2
LanguageBind-Audio (Zhu et al., 2024)	428M	✗	✓	✗	25.8	3.3	15.3	27.7	3.1	13.8	24.1

ing this distinction, we report model performance separately on each subset (**Visual** or **Audio** in the left columns of the table), and also provide the average performance over the full dataset (**Avg**).

For MLLM evaluation, we adapt two settings: MLLM only or MLLM with segment captions. In the 'MLLM only', for MLLMs that directly predict timestamps, we uniformly sample 100 frames from the full video and prompt the model to output a single predicted timestamp $\hat{t}_i$. A prediction is considered correct if $|\hat{t}_i - t_i| \leq 10$s, and performance is reported using Recall@1. 'MLLM with segment captions' denotes that each video is divided into non-overlapping 10-second segments. MLLMs generates captions for each segment, and both queries and segment captions are embedded using the Qwen-3 embedding model (Yang et al., 2025). Segments are ranked by cosine similarity between them, and Recall@K is computed by checking whether the top-K ranked segments include the one containing the ground-truth timestamp t_i.

Baselines. For MLLMs, we adopt Gemini 2.5 Flash, Qwen2.5-VL-3B (Bai et al., 2025), InternVL3-2B (Zhu et al., 2025), and Qwen2.5-Omni-3B (Xu et al., 2025), which directly predict a timestamp given a query and video input (*i.e.*, 'MLLM only') or instructed to generate visual and audio descriptions (*i.e.*, MLLM with segment captions) We also evaluate embedding-based retrieval baselines. For visual retrieval, we use CLIP (Radford et al., 2021), LanguageBind (Zhu et al., 2024), and InternVideo2 (Wang et al., 2024). For audio retrieval, we use CLAP (Elizalde et al., 2023) and LanguageBind-Audio (Zhu et al., 2024). Segments are ranked by cosine similarity in the embedding space.

5.2 Main Results

Table 3 reports results on vision-related queries and audio-related queries.

MLLMs in timestamp generation setting. Gemini 2.5 Flash is a competitive reference point for single timestamp prediction. With visual input only it reaches Visual R@1 of 6.0 and audio R@1 of 4.0. Adding raw audio keeps Visual R@1 at 6.0 and raises audio R@1 to 6.0. Adding subtitle based ASR text raises Visual R@1 to 26.0 and audio R@1 to 26.0, so subtitles drive most of the gain and they also provide a useful auxiliary cue for Visual-related retrieval. Open-source MLLMs remain weak on audio-related timestamp generation without subtitles, and Qwen2.5-Omni also shows only limited benefit from adding raw audio input in this setting. This suggests that waveform audio alone is not a reliable signal for single-timestamp prediction compared to subtitle based speech text.

MLLMs in segment caption setting. In the caption setting without subtitles, Qwen2.5-Omni performs best with Visual R@10 of 37.6 and audio R@10 of 27.5, and vision-related queries show higher retrieval than audio-related queries. With segment aligned subtitles, audio-related queries improve substantially, with audio R@10 rising from 27.5 to 48.2 for Qwen2.5-Omni. The same trend holds for Qwen2.5VL and InternVL3 and across both settings Qwen2.5-Omni remains the strongest open-source MLLM.

Embedding models. On vision-related queries, visual embedding models are the strongest, with LanguageBind reaching Visual R@1 of 30.5 and Visual R@10 of 62.7 and CLIP reaching Visual R@1 of 14.9 and Visual R@10 of 54.3, which stays above MLLMs setting. A similar trend also appears on audio-related queries. Visual embedding models remain competitive with audio R@10 of 37.0 for LanguageBind and 34.4 for CLIP, while audio embedding models are lower with 24.2 for CLAP and 24.1 for LanguageBind-audio. This suggests that many audio-related queries align with speech linked cues or visually correlated context, so visual representations can still localize the target moment better than acoustic embeddings alone.

6 Related Work

Video Moment Retrieval VMR localizes the time span that matches a natural language query in an untrimmed video. A common direction models localization as DETR style span detection and predicts a small set of candidate moments end to end and later variants strengthen query conditioning and improve boundary alignment (Lei et al., 2021b; Moon et al., 2023; Lee and Byun, 2024). In parallel, multimodal large language models are being adapted for video temporal grounding by combining multimodal understanding with structured timestamp prediction and stepwise reasoning (Ren et al., 2024; Guo et al., 2025; Liu et al., 2025). As videos get longer, exhaustively scoring every segment becomes expensive, so many systems adopt coarse to fine pipelines that first retrieve query related regions and then refine the boundaries with a stronger local model (Hou et al., 2023; Pan et al., 2025).

Video Moment Retrieval datasets. Prior work evaluates VMR using datasets that align natural language queries with annotated temporal windows. Short and domain specific datasets cover activities and lifestyle videos and TV shows and egocentric recordings and they pair each query with a temporal window (Gao et al., 2017; Lei et al., 2021b, 2020; Grauman et al., 2022). Long video datasets expand duration and domain diversity and they move closer to open world content, and some also annotate event boundaries with multimodal descriptions (Yuan et al., 2025; Geng et al., 2025). Audio resources scale audio text supervision or dense audio visual event boundaries, but they do not define user query driven moment retrieval in long videos (Mei et al., 2024; Geng et al., 2023). We use timestamped YouTube comments and viewer reactions to select moments and we generate modality aware queries so we can separate vision-related and audio-related queries.

7 Conclusion

We propose TCVP, a practical pipeline for video moment retrieval dataset generation using timestamped YouTube comments. By introducing comment filtering and modality gating, our pipeline removes uninformative comments and ambiguous modality cues, and generates concise queries that remain faithful to user intent. We validate the effectiveness of TCVP through both qualitative analyses and quantitative human evaluations. Annotators prefer queries generated with comments in 70% and favor comment-based moment selection in 95% of videos. We further evaluate existing VMR models on the dataset generated by TCVP. The results show that current models remain limited when queries depend on spoken or auditory content. We hope TCVP supports future work that better reflects real user intent and that handles both visual and audio evidence for moment retrieval.

Limitations

Our approach constructs a VMR dataset guided by user reactions, but it has several limitations stemming from automated processing. First, modality assignment is restricted to a binary choice between visual and audio cues. This design cannot capture cases where user intent jointly depends on both modalities. Extending the label space can be important direction. Second, comment filtering and modality gating depend on a fixed similarity threshold τ. Future work will calibrate τ with light human validation and it will explore adaptive rules that reduce sensitivity to a single threshold.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Boris Chen, Ben Klein, Jason Ge, Avneesh Saluja, Guru Tahasildar, Abhishek Soni, Juan Vimberg, Elliot Chow, Amir Ziai, Varun Sekhri, Santiago Castro, Keila Fong, Kelli Griggs, Mallia Sherzai, Robert Mayer, Andy Yao, Vi Iyengar, Jonathan Solorzano-Hamilton, Hossein Taghavi, and Ritwik Kumar. 2023. Building in-video search.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Ioana Croitoru, Simion-Vlad Bogolin, Samuel Albanie, Yang Liu, Zhaowen Wang, Seunghyun Yoon, Franck Dernoncourt, Hailin Jin, and Trung Bui. 2023. Moment detection in long tutorial videos. In *ICCV*.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2023. Clap learning audio concepts from natural language supervision. In *ICASSP*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*.

Tiantian Geng, Teng Wang, Jinming Duan, Runmin Cong, and Feng Zheng. 2023. Dense-localizing audio-visual events in untrimmed videos: A large-scale benchmark and baseline. In *CVPR*.

Tiantian Geng, Jinrui Zhang, Qingni Wang, Teng Wang, Jinming Duan, and Feng Zheng. 2025. Longvale: Vision-audio-language-event benchmark towards time-aware omni-modal perception of long videos. In *CVPR*.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, and 1 others. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. 2025. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. In *AAAI*.

Zhijian Hou, Wanjun Zhong, Lei Ji, Difei Gao, Kun Yan, Wk Chan, Chong-Wah Ngo, Mike Zheng Shou, and Nan Duan. 2023. Cone: An efficient coarse-to-fine alignment framework for long video temporal grounding. In *ACL*.

Mina Huh, Ding Li, Kim Pimmel, Hijung Valentina Shin, Amy Pavel, and Mira Dontcheva. 2025. Videodiff: Human-ai video co-creation with alternatives. In *CHI*.

Pilhyeon Lee and Hyeran Byun. 2024. Bam-detr: Boundary-aligned moment detection transformer for temporal sentence grounding in videos. In *ECCV*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021a. Detecting moments and highlights in videos via natural language queries. In *NeurIPS*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021b. Detecting moments and highlights in videos via natural language queries. *NeurIPS*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*.

Zinuo Li, Xian Zhang, Yongxin Guo, Mohammed Bennamoun, Farid Boussaid, Girish Dwivedi, Luqi Gong, and Qiuhong Ke. 2025. Watch and listen: Understanding audio-visual-speech moments with multimodal llm. In *NeurIPS*.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. 2023. Univtg: Towards unified video-language temporal grounding. In *ICCV*.

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

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. 2024. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *TASLP*.

WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. 2023. Query-dependent video representation for moment retrieval and highlight detection. In *CVPR*.

OpenAI. 2025. Gpt-4.1 model. OpenAI API Documentation.

Junwen Pan, Rui Zhang, Xin Wan, Yuan Zhang, Ming Lu, and Qi She. 2025. Timesearch: Hierarchical video search with spotlight and reflection for human-like long video understanding. *arXiv preprint arXiv:2504.01407*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*.

Mattia Soldan, Alejandro Pardo, Juan Leon Alcazar, Fabian Caba Heilbron, Chen Zhao, Silvio Giancola, and Bernard Ghanem. 2022. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *CVPR*.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, and 1 others. 2024. Internvideo2: Scaling foundation models for multimodal video understanding. In *ECCV*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Saelyne Yang, Sunghyun Park, Yunseok Jang, and Moontae Lee. 2024. Ytcommentqa: video question answerability in instructional videos. In *AAAI*.

Huaying Yuan, Jian Ni, Zheng Liu, Yueze Wang, Junjie Zhou, Zhengyang Liang, Bo Zhao, Zhao Cao, Zhicheng Dou, and Ji-Rong Wen. 2025. Momentseeker: A task-oriented benchmark for long-video moment retrieval. In *CVPR*.

Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, and 1 others. 2024. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *ICLR*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Generation Prompts

Modality specific captioning. Given a short segment, the model outputs exactly two lines. The VISUAL line describes what is visible in the sampled frames and the AUDIO line describes what is heard. This separation matters because the same comment can be supported by visual evidence or by speech and sound cues and we want to preserve both signals explicitly.

Modality Specific Captioning

```
Analyze this {segment_duration}-second video segment and
      write exactly two parts as two lines.

VISUAL: Describe only what is visible, including people,
      objects, actions, scene context, colors, and motion.
AUDIO: Describe only what is audible, including speech
      content, speaker tone, music, sound effects,
      ambient sounds, and silence.

Keep both lines concise but informative and do not add
      any other text.
```

Segment caption integration. We also merge consecutive segment captions into a single clip level description. We use this stage when the target moment spans multiple segments or when later steps require a consolidated view. The prompt enforces JSON output with separate visual caption and audio caption fields and it requires many short atomic sentences. This format keeps fine details and it preserves temporal order which improves matching and filtering.

Segment Caption Integrator

```
You consolidate consecutive caption segments into one
      caption with no information loss.

Input has {segment_count} captions labeled S1 through
      S{segment_count}.
Return only valid JSON and do not output any other text.

Write in present tense with a neutral descriptive tone
      and keep the event order consistent with the input.

Use many short sentences and keep each sentence under 15
      words.
Make each sentence atomic and keep one concrete
      observation per sentence.
Do not merge different events into one sentence.

Do not summarize and do not invent details.
Do not include timestamps or durations and avoid
      transition words such as then or afterward.

visual_caption describes only what is visible, including
      people, actions, objects, motion, background,
      lighting, camera viewpoint, and on screen text.
audio_caption describes only what is audible, including
      speech content, speaker tone, music, ambient noise,
      sound effects, silence, laughter, and applause.

{
  "visual_caption": "<short sentences describing only
       visible content>",
  "audio_caption": "<short sentences describing only
       audible content>"
}
```

Moment search query generation. Finally, we generate a moment query from a timestamped user comment. The model infers whether the intent is visual or audio and it extracts keywords directly from the comment to stay grounded. It then selects a minimal set of focus keywords and it writes one short natural query that a viewer would plausibly type. The JSON schema records the modality decision and the extracted keywords and the final query which makes the process auditable and supports error analysis.

Moment Search Query Generation

```
Generate one short and realistic query that a viewer
      would type to revisit the exact moment behind the
      comment.

First infer whether the intent is visual or audio.
Extract comment_keywords as words or short phrases taken
      from the comment.
Select focus_keywords as the smallest subset that carries
      the retrieval intent.

Write one query that centers on focus_keywords and add
      one or two small cues that help localize the moment.
Use a conversational search style and write one sentence
      in 12 to 25 words.

For visual queries emphasize visible entities, actions,
      and scene details.
For audio queries include a short quote or a natural
      paraphrase and add sound cues when helpful.

Avoid timestamps and durations and avoid narration that
      describes watching the video.
Return valid JSON only and follow this schema.

{
  "modality": "visual" or "audio",
  "comment_keywords": ["phrases taken from the comment"],
  "focus_keywords": ["subset used to build the query"],
  "intent": "one sentence describing what the user wants
       to revisit",
  "query": "one natural query in 12 to 25 words",
  "rationale": "why the focus keywords and cues match the
       comment intent"
}

Example
Comment: "Love this quote, Jillian isn't sick she's a
      dancer!"
{
  "modality": "audio",
  "comment_keywords": ["Love this quote", "Jillian isn't
       sick", "dancer"],
  "focus_keywords": ["Jillian isn't sick", "dancer"],
  "intent": "Find the moment when the memorable Jillian
       line is spoken",
  "query": "the moment someone says Jillian isn't sick
       she's a dancer and the audience reacts",
  "rationale": "The key phrase comes from the comment and
       the reaction cue helps localization"
}
```

We use three prompts that generate modality aware captions and that rewrite timestamped user comments into natural moment queries. We design them to limit variation across runs and to produce outputs that are straightforward to parse and compare across models. Each prompt fixes the output structure and length so the model stays tied to the given segment and avoids summaries.

B Human Evaluation

We run a human study with undergraduate annotators to evaluate query realism and moment selection quality. Each item shows a YouTube video with the target moment highlighted and annotators can watch the target and nearby context before making one choice. Annotators receive written instructions that define the decision criteria and they can skip any item and they can stop at any time. The instructions state that the study uses public YouTube content and that some items may be uncomfortable. We recruit annotators through university channels and we provide compensation at 10,000 KRW per hour. All annotators provide informed consent and we do not collect sensitive personal information beyond what is needed for compensation. We use public YouTube videos and timestamped comments and we do not redistribute raw videos. We remove user identifiers from any released data and we document permitted use and restrictions.

B.1 Realistic query selection

This task tests whether a generated query sounds like a natural query that a viewer would type after watching the full video. Annotators are shown a target moment and a list of candidate queries and they select the one they would most likely use to retrieve the same moment. We treat the vote as a preference signal for query naturalness and clarity.

Annotator Instructions: Realistic Query Selection

1. You will be given a YouTube video, a specific moment from that video, and a list of possible text queries.
2. Imagine the following situation.
   a. You previously watched the entire YouTube video.
   b. Later, you want to find that specific moment again using a text query.
   c. You have an AI system that can locate the exact timestamp if you provide a text description of the moment.
3. From the list below, choose the query you would most likely give to the AI system.
4. Select a natural and realistic search query.
5. `Natural` means it sounds like something a real person would type.
6. `Realistic` means it is clear and not overly long.

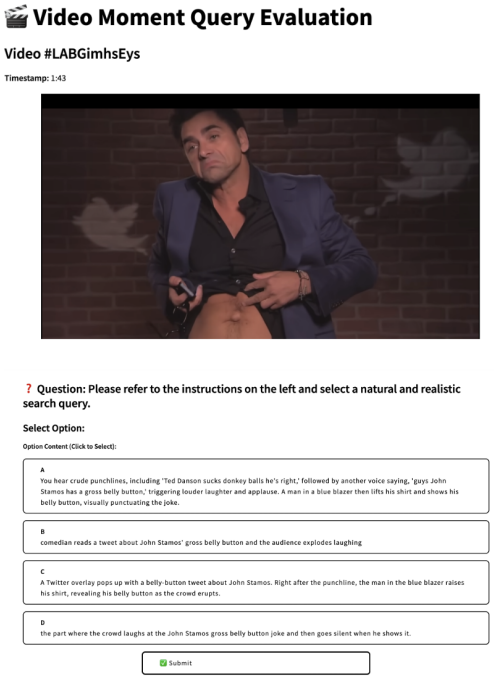

Figure 8: The survey environment for realistic query selection.

Timestamp selection. This task evaluates whether the candidate moments align with the target moment. Annotators watch two candidate moments from the same video and they choose the one that better matches the target moment based on the core content rather than length or excitement. We treat the vote as a measure of moment selection quality.

Annotator Instructions: Timestamp Selection

1. You will see two candidate moments from the same YouTube video shown side by side.
2. Watch both moments and choose the one that better captures the core content of the video.
3. Select the option that is more representative and more informative, rather than the option that is simply longer or more eventful.

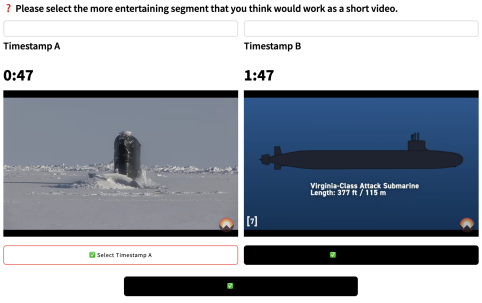

Figure 9: The survey environment for timestamp selection in moment comparison.

C Evaluation Setting & Prompts

We run all pipeline stages with inference only on a single machine that has one NVIDIA A100 80GB GPU. We also use API based models for some stages and the total API cost is under 30 USD.

We use two prompts during evaluation. The first prompt is used for segment captioning with Qwen2.5-VL and InternVL3 and Qwen2.5-Omni. We provide a time bounded segment and a set of uniformly sampled frames and we ask the model to write a short caption grounded in the segment. For Qwen2.5-Omni, we additionally provide audio input and we explicitly instruct the model to use both visual and audio cues. The second prompt is used for timestamp prediction with Qwen2.5-VL in the generation based setting. We provide uniformly sampled frames from the full video and we ask the model to output a single timestamp that best matches the query under a strict output format. For Gemini 2.5 Flash, we run this setting on a balanced subset with 50 Visual samples and 50 Audio samples.

Segment Captioning Prompt

```
You are given a video segment #{segment_idx} from
      {start:.2f}s to {end:.2f}s.
{len(frame_timestamps)} frames are uniformly sampled at
      these timestamps (sec): [{ts_str}].

Provide a concise caption summarizing what happens in
      this segment.
Use both visual frames and the accompanying audio.
Be specific to the segment content and keep it short in
      one or two sentences.
```

Timestamp Prediction Prompt

```
Video 1 lasts for {duration_text}, and {num_frames}
      frames are uniformly sampled from it.
{timestamp_block}

Identify the single most relevant moment as one timestamp
      in seconds that matches the query.

Strict output format:
Return only one timestamp in seconds as a float.
Preferred: a bare JSON number such as 12.345.
Also acceptable: {"timestamp":12.345} or [12.345] with
      only one value.
Do not return ranges or natural language.
Keep the timestamp within [0, video_duration].

Textual query: {query}
```