When One Moment Isn't Enough: Multi-Moment Retrieval with Cross-Moment Interactions Supplemental Material

Zhuo Cao[1]*, **Heming Du**[1]*, **Bingqing Zhang**[1], **Xin Yu**[1], **Xue Li**[1]†, **Sen Wang**[1]

[1] The University of Queensland, Australia
{william.cao, heming.du, bingqing.zhang, xin.yu}@uq.edu.au
xueli@eesc.uq.edu.au, sen.wang@uq.edu.au

In this supplementary material we:

- Provide the dataset license terms;
- Provide a detailed description of our dataset annotation process;
- Discuss key differences between QV-M^2 and QVHighlights;
- Discuss broader impact of our work;
- Present implementation details of our framework.

1 License and Data Usage

QV-M^2 dataset is built upon the QVHighlights dataset [1] (CC BY-NC-SA 4.0) and includes additional annotations. The original dataset is publicly available at GitHub Repository, and its license can be found at Creative Commons Attribution-NonCommercial-ShareAlike 4.0. The additional annotations introduced in our dataset follow the same CC BY-NC-SA 4.0 license, ensuring compatibility with the original dataset. Our dataset is strictly for research purposes and should not be used for applications that may violate human rights, consistent with the guidelines set forth in the original dataset.

2 Dataset Annotation Process

To facilitate the annotation of multi-moment temporal segments, we developed a dedicated annotation pipeline and an accompanying annotation tool based on Gradio [2]. This tool streamlines the process of selecting relevant video segments, defining multiple moment annotations, and refining query descriptions to ensure high-quality dataset annotations. Our annotation interface consists of three primary sections, as shown in Figure 1:

Video Selection Panel – Allows annotators to input the path of the target video and select the video to be annotated.

Temporal Annotation Panel – Displays the selected video and enables annotators to mark multiple relevant moments within the video.

Query Editing and Finalization Panel – Supports query generation and manual refinement before saving the final annotation.

The step-by-step annotation procedure is as follows:

*Equal Contribution
†Corresponding Authors

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

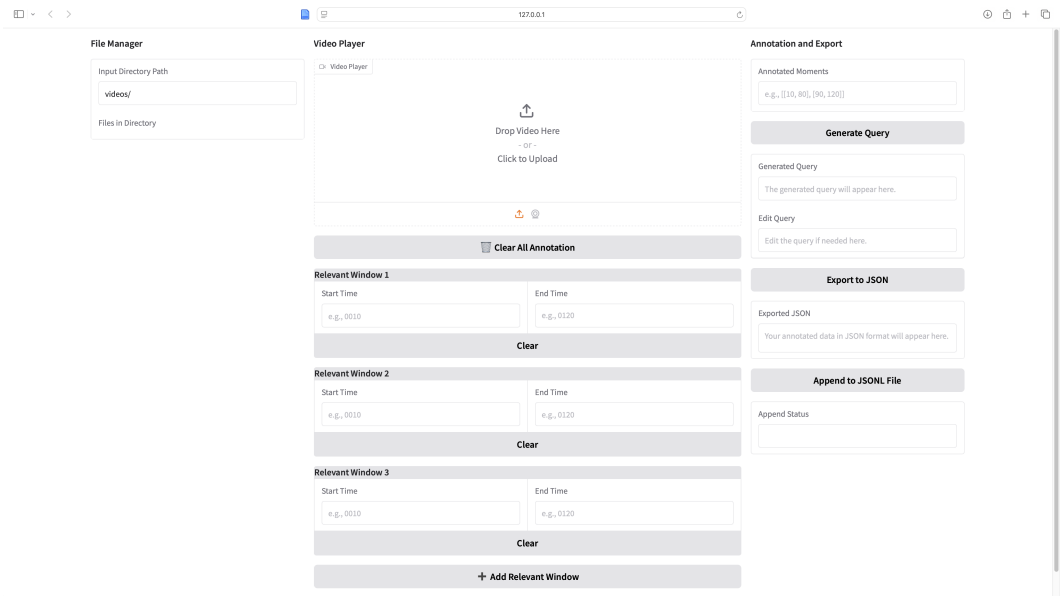


Figure 1: Annotation Interface

1. **Video Selection:** As shown in Figure 2a, annotators input the video path and select the video for annotation. The chosen video is displayed in the second panel for further processing.

2. **Multi-Moment Annotation:** After watching the video, annotators identify semantically relevant segments and mark them in the "Relevant Window" section. The tool allows multiple temporal segments to be labeled, automatically converting them into the standardized annotation format, as shown in Figure 2b.

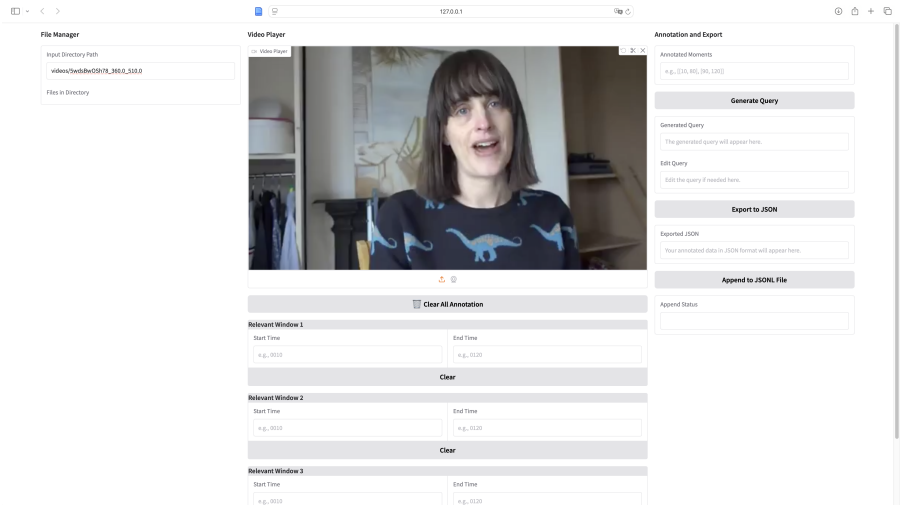
3. **Query Generation and Refinement:** As shown in Figure 2c, the tool leverages Qwen2.5-VL [3] to generate an initial query description based on the selected video segments. Annotators then review and refine the generated query based on their interpretation and annotation guidelines.

4. **Final Annotation Saving:** Once the query is finalized, the tool automatically formats the annotations and stores them in the designated output files.

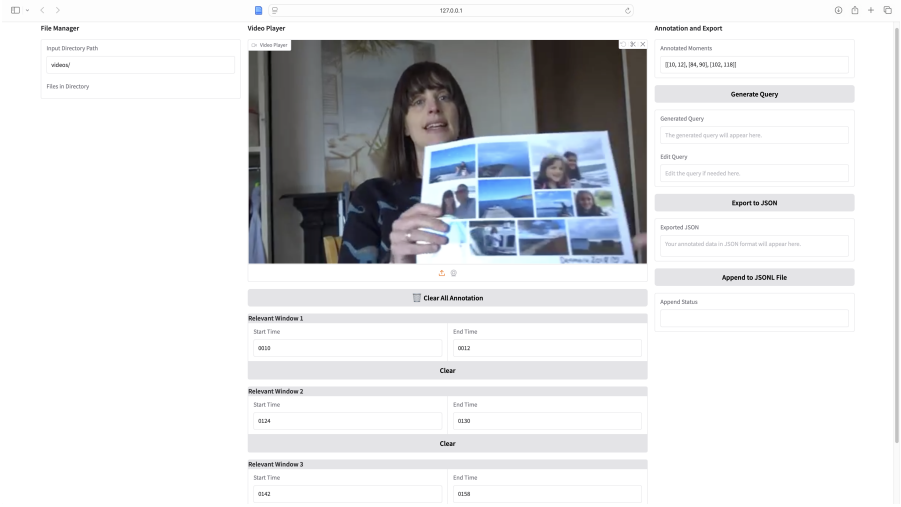
Our annotation pipeline enhances efficiency while maintaining human-in-the-loop quality control. This semi-automated approach ensures that annotations align with the intended dataset requirements, improving consistency and scalability in multi-moment retrieval datasets.

3 Case Study

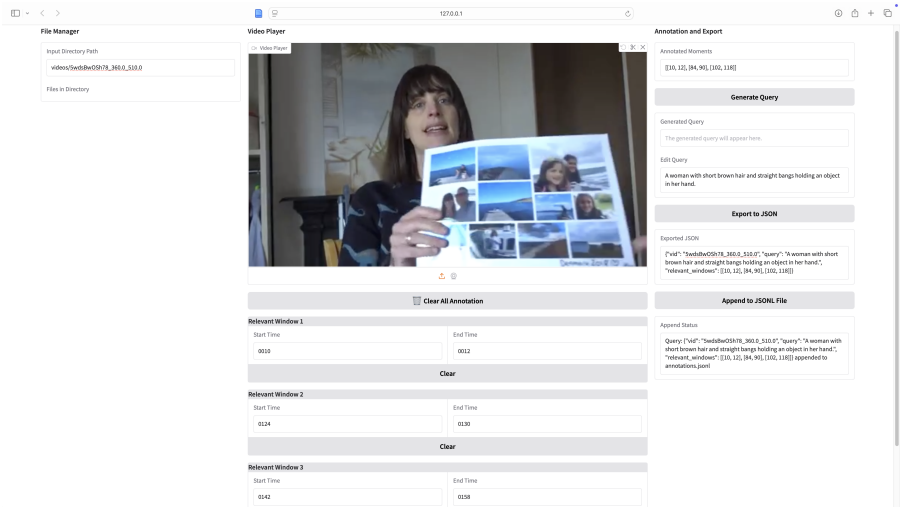
We present a comparative visualization of the prediction results from SMR and MMR in Figures 3 and 4. These examples showcase the performance of FlashVTG [4] and FlashMMR on the QV-M^2 test set. Notably, under multi-moment scenarios, the prior SMR method tends to concentrate its predictions around a single timestamp, failing to capture the temporal diversity of semantically similar moments. In contrast, our proposed FlashMMR method accurately predicts multiple temporally distinct but semantically related moments, demonstrating improved capability in multi-target settings.



(a) Video Selection



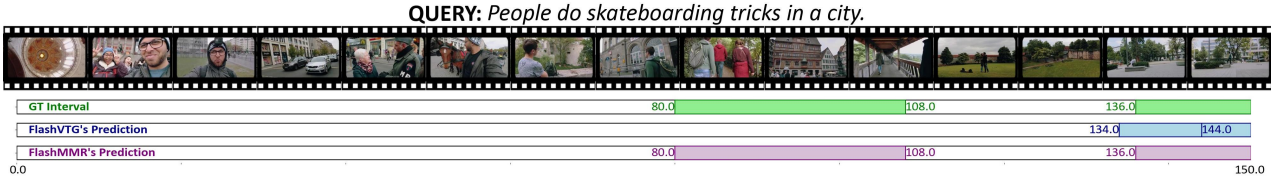
(b) Multi-Moment Timestamp Annotation



(c) Query Generation and Saving

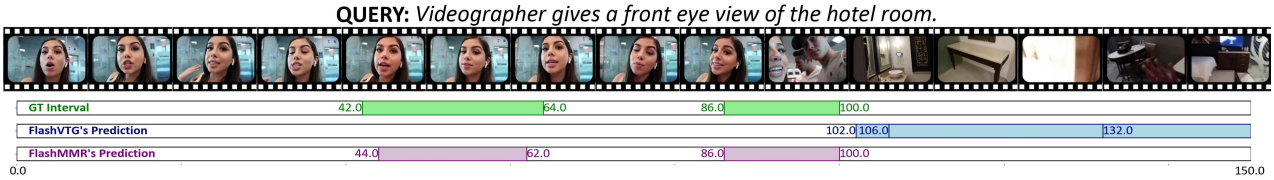
Figure 2: Annotation Interface Overview. (a) Video selection; (b) Multi-moment timestamp annotation; (c) Query generation and saving.

QUERY: *People do skateboarding tricks in a city.*



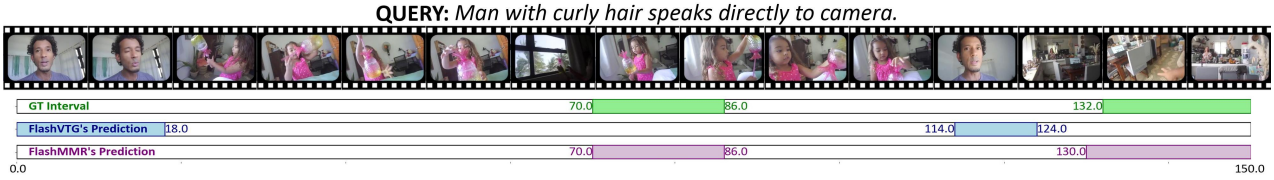
(a)

QUERY: *Videographer gives a front eye view of the hotel room.*



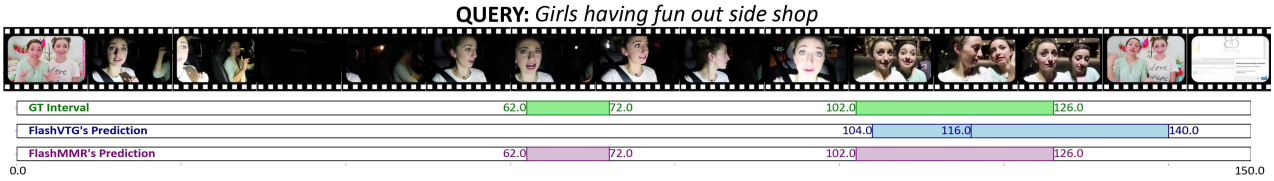
(b)

QUERY: *Man with curly hair speaks directly to camera.*



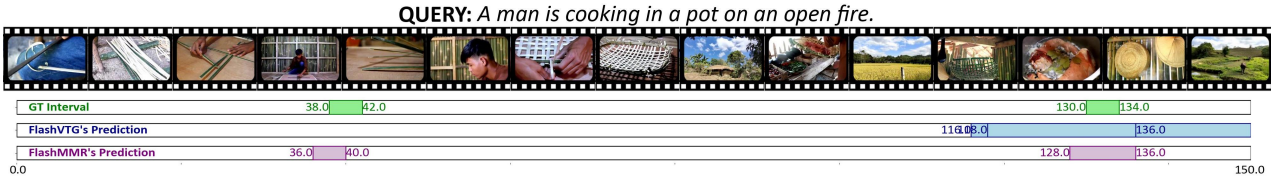
(c)

QUERY: *Girls having fun out side shop*



(d)

QUERY: *A man is cooking in a pot on an open fire.*



(e)

Figure 3: Case Study 1

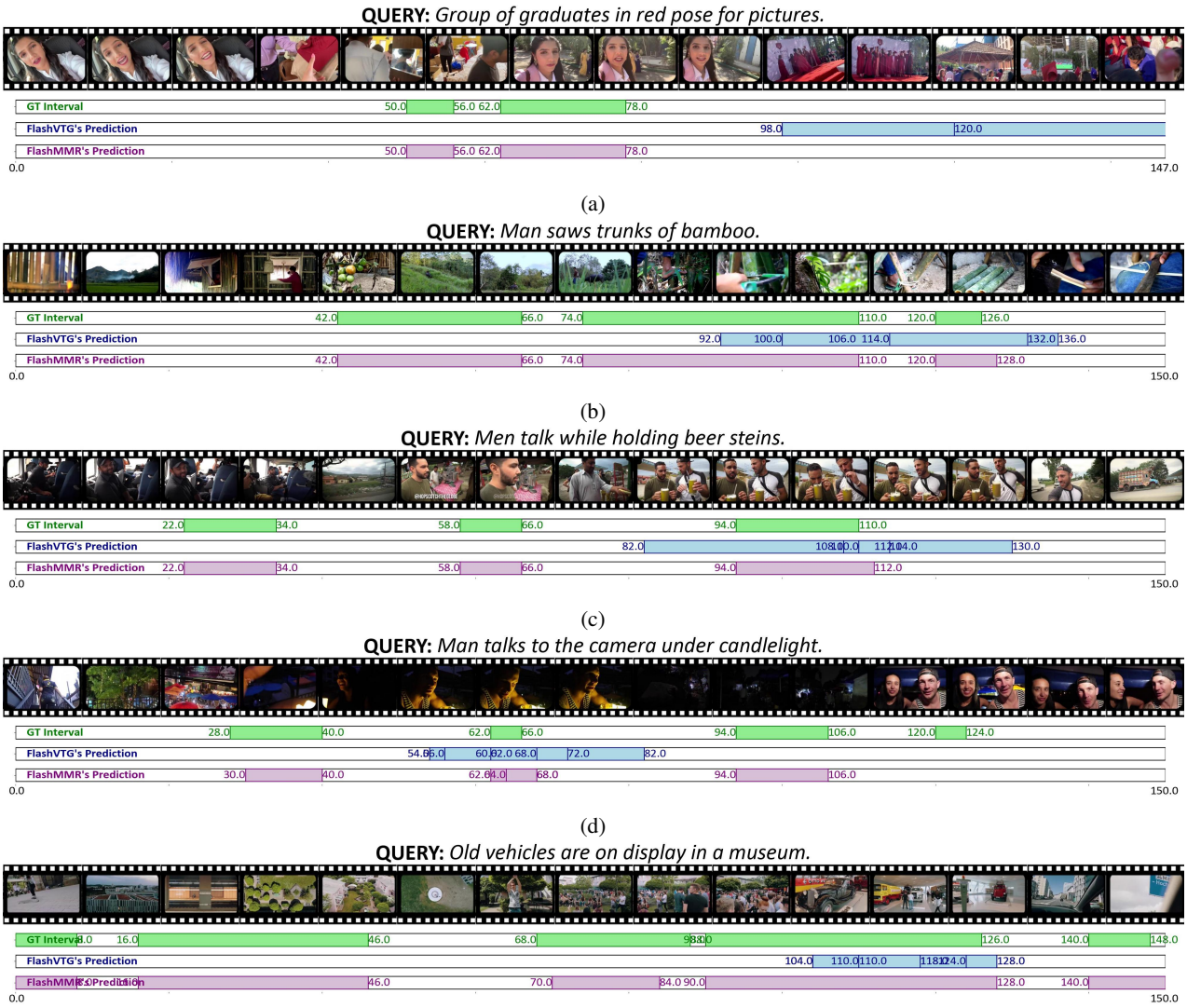


Figure 4: Case Study 2

4 Key differences between QV-M² and QVHighlights

We summarize the key differences between QVHighlights and QV-M^2 along five axes.

First, QV-M^2 exhibits a higher average number of relevant moments per query (2.9 vs. 1.8), explicitly reflecting a one-to-many retrieval regime rather than the predominantly one-to-one setting in QVHighlights. Second, moment density is substantially larger in QV-M^2 (25.5% of video duration contains annotated moments) than in QVHighlights (16.4%), which increases the need for precise temporal localization during training and evaluation.

Third, the annotation strategy differs: QVHighlights uses a single positive sentence per moment, whereas QV-M^2 contains multiple positive and negative sentences that emphasize contextual cues and temporal dependencies, which encourages cross-moment reasoning. Fourth, annotation scope in QV-M^2 is cross-video and dense (the same query can be grounded to moments in multiple videos), while QVHighlights restricts annotations to intra-video matches; this makes QV-M^2 better suited for video-corpus moment retrieval. Finally, evaluation protocols expand accordingly: QVHighlights relies on conventional mAP/IoU metrics, whereas QV-M^2 introduces G-mAP, mAP@k_tgt and

Table 1: Scalability with video length.

Video Length ($\bar{V}$)	Inference Runtime (s)	Inference FLOPs (M)	Training Runtime (s)	Training FLOPs (M)
50	0.0133	19051.2	0.0225	19175.0
100	0.0132	19051.2	0.0229	19546.0
150	0.0131	19051.2	0.0259	20164.4
200	0.0130	19051.2	0.0340	21030.2
250	0.0129	19051.2	0.0459	22143.3
300	0.0130	19051.2	0.0598	23503.8
400	0.0130	19051.2	0.0927	26966.8

mR@k to measure performance in multi-moment retrieval scenarios while remaining compatible with single-moment retrieval measures.

These differences make QV-M^2 a more challenging and realistic benchmark for developing models that must detect, disambiguate, and rank multiple temporally distributed moments across a video corpus.

5 Computational Complexity Analysis of PV module

We analyse the computational cost of the Post-Verification module both theoretically and empirically. Below, we state the asymptotic cost and clarify the notation, and validate these predictions with controlled experiments that vary video length and number of candidate predictions.

Let K denote the number of candidate predictions per video, d the feature dimensionality, L the average candidate length, and V the average video length.

5.1 Theoretical complexity

The theoretical complexity of the Post-Verification module is as follows:

- Inference: Time Complexity: $O(K\bar{L}d^2)$, Memory Complexity: $O(K\bar{L}d)$
- Training: Time Complexity: $O(K\bar{L}d^2 + \bar{V}^2 d)$, Memory Complexity: $O(K\bar{L}d + \bar{V}^2)$

Where K is the number of candidate predictions per video, d is the feature dimension, $\bar{L}$ is the average length of candidates and $\bar{V}$ is the average length of videos.

5.2 Experimental Validation

We empirically validated the above theoretical analysis by varying video lengths and candidate counts. All experiments utilized a batch size of 32, a feature dimension of 256, and were performed on an NVIDIA RTX 4090 GPU with an Intel(R) Core(TM) i9-14900KF CPU.

Scalability with Video Length. Table 1 confirms our theoretical expectation: during inference, runtime and FLOPs remain stable irrespective of video length. In contrast, during training, runtime and FLOPs increase linearly with video length due to the $\bar{V}^2 d$ term. Since $\bar{V}^2 < K\bar{L}d$ in our setting, the $\bar{V}^2 d$ term remains a lower-order contribution, which explains why the additional FLOPs during training are moderate.

Scalability with Candidate Set Size. Experiments show in Table 2 demonstrate a clear linear relationship between candidate count and computational overhead in both inference and training phases, again aligning well with our complexity analysis.

In summary, these experimental results demonstrate that, under the current setting, the computational complexity of our Post-Verification module is reasonable and acceptable.

6 Broader Impact

Our FlashMMR model and the accompanying QV-M^2 dataset advance "one-query, multiple-segment" moment retrieval, enabling precise localization of all relevant clips within single video. This capability

Table 2: Scalability with Candidate Set Size.

Candidate Count (K)	Inference Runtime (s)	Inference FLOPS (M)	Training Runtime (s)	Training FLOPS (M)
50	0.0134	19051.2	0.0261	20164.4
60	0.0156	22861.5	0.0284	23974.7
70	0.0181	26671.7	0.0310	27784.9
80	0.0203	30482.0	0.0337	31595.2
90	0.0230	34292.2	0.0364	35405.4

can substantially improve efficiency in content indexing, highlight generation, and automated editing, reducing manual effort in media production and analysis.

6.1 Positive Societal Benefits

Enhanced Accessibility and Learning. In educational and assistive settings, users can jump directly to demonstration steps or lecture segments, benefiting learners with diverse needs and those with disabilities.

Research Catalyst. By providing a standardized benchmark with clear metrics, QV-M^2 fosters fair comparisons and accelerates innovation in video understanding.

6.2 Risks and Ethical Considerations

Privacy and Surveillance. The same high-precision localization may be repurposed for large-scale monitoring or tracking of individuals without consent.

Dataset Bias. QV-M^2 focuses on vlogs and news footage may underrepresent certain languages, regions, or cultural contexts, leading to uneven model performance across communities.

6.3 Mitigation Strategies

Controlled Release. We distribute models and data under a CC BY-NC-SA 4.0 license, prohibiting commercial or surveillance applications.

Transparent Annotation. We publish detailed labeling guidelines and sampling protocols to enable community auditing and bias assessment.

Incremental Governance. We recommend mandatory human-in-the-loop verification for sensitive domains (e.g., face recognition, public safety), and future expansion of QV-M^2 to include more diverse cultural and linguistic samples for fairness evaluation.

7 Implementation Details of FalshMMR

7.1 Key/Value construction in ACA

The Adaptive Cross-Attention (ACA) module's internal attention structure is different from the classic cross-attention structure, especially in the key/value construction. Our ACA module adopts the exact same design and code implementation from CG-DETR [5]/FlashVTG [4]. Here we clarify the exact behaviour of this module.

Notation. Let $\mathbf{V}$ denote the input video clip features and $\mathbf{Q}$ the (textual) query token features. We use learnable linear projections $p_Q(\cdot), p_K(\cdot), p_V(\cdot)$ to produce query, key and value for cross-attention.

Table 3: Role, symbol and shape used in the cross-attention with dummy tokens.

Role	Symbol	Shape
Query = video clips	$p_Q(\mathbf{V})$	$L_v \times d$
Key = text + dummy	$p_K([\mathbf{Q}, \tilde{\mathbf{D}}])$	$(L_q + L_d) \times d$
Value = text only	$p_V(\mathbf{Q})$	$L_q \times d$

Attention weights and fused features. The attention weights and fused features are

$$W_{i,j} = \frac{\exp\big(Q_i K_j^\top / \sqrt{d}\big)}{\sum_{k=1}^{L_q+L_d} \exp\big(Q_i K_k^\top / \sqrt{d}\big)}, \tag{1}$$

$$F_i = \sum_{j=1}^{L_q} W_{i,j}\, V_j, \quad F \in \mathbb{R}^{L_v \times d}. \tag{2}$$

We denote the input video clips V as a query matrix $p_Q(\mathbf{V}) \in \mathbb{R}^{L_v \times d}$, the key matrix as $p_K([\mathbf{Q}, \tilde{\mathbf{D}}]) \in \mathbb{R}^{(L_q+L_d)\times d}$ formed by concatenating the text features $\mathbf{Q}$ with dummy tokens $\tilde{\mathbf{D}}$, and the value matrix as $p_V(\mathbf{Q}) \in \mathbb{R}^{L_q \times d}$ using only the text features. The functions $p_Q(\cdot)$, $p_K(\cdot)$, and $p_V(\cdot)$ are learnable linear projections that map input features into the query, key, and value spaces, respectively.

Code-level detail. In our implementation we compute attention over the full key bank but then mask out the dummy slice when forming the final attended output. The following snippet illustrates the essential operation:

Listing 1: Key masking when forming attended output.

```
# crossattention.py
attn_output = torch.bmm(
    attn_w[:, :, num_dummies:],    # weights over text positions
    v[:, num_dummies:, :]          # values = text tokens only
)
```

Hence, dummy tokens steer the attention via the key but never inject their own content into the output, which mitigating the mismatch between limited query scope and untrimmed video semantics while keeping the fused feature purely textual.

Comprehensive implementation details, including the original dataset annotations, training and evaluation scripts, as well as usage instructions, are provided in the supplementary materials. Specifically, dataset-related resources can be found in the `FlashMMR/data/QV-M`^{2} folder, while the complete codebase is available in the `FlashMMR` directory. Table 4 shows the exact dataset split statistics in experiment, QV-M^2 here include the original annotations from QVHighlights [1].

Table 4: Dataset statistics for training, validation, and test splits.

Dataset	Train	Val	Test
TACoS [6]	9790	4436	4001
Charades-STA [7]	12404	-	3720
QVHighlights [1]	7218	1550	1542
QV-M²	8878	1779	1865

References

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

[2] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] Zhuo Cao, Bingqing Zhang, Heming Du, Xin Yu, Xue Li, and Sen Wang. Flashvtg: Feature layering and adaptive score handling network for video temporal grounding. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 9208–9218, February 2025.

[5] WonJun Moon, Sangeek Hyun, SuBeen Lee, and Jae-Pil Heo. Correlation-guided query-dependency calibration in video representation learning for temporal grounding. *arXiv preprint arXiv:2311.08835*, 2023.

[6] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1: 25–36, 2013.

[7] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017.