# FlashVTG: Feature Layering and Adaptive Score Handling Network for Video Temporal Grounding

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

[1] The University of Queensland, Australia

{william.cao, bingqing.zhang, heming.du, xin.yu}@uq.edu.au

xueli@eesc.uq.edu.au, sen.wang@uq.edu.au

## Abstract

*Text-guided Video Temporal Grounding (VTG) aims to localize relevant segments in untrimmed videos based on textual descriptions, encompassing two subtasks: Moment Retrieval (MR) and Highlight Detection (HD). Although previous typical methods have achieved commendable results, it is still challenging to retrieve short video moments. This is primarily due to the reliance on sparse and limited decoder queries, which significantly constrain the accuracy of predictions. Furthermore, suboptimal outcomes often arise because previous methods rank predictions based on isolated predictions, neglecting the broader video context. To tackle these issues, we introduce FlashVTG, a framework featuring a Temporal Feature Layering (TFL) module and an Adaptive Score Refinement (ASR) module. The TFL module replaces the traditional decoder structure to capture nuanced video content variations across multiple temporal scales, while the ASR module improves prediction ranking by integrating context from adjacent moments and multi-temporal-scale features. Extensive experiments demonstrate that FlashVTG achieves state-of-the-art performance on four widely adopted datasets in both MR and HD. Specifically, on the QVHighlights dataset, it boosts mAP by 5.8% for MR and 3.3% for HD. For short-moment retrieval, FlashVTG increases mAP to 125% of previous SOTA performance. All these improvements are made without adding training burdens, underscoring its effectiveness. Our code is available at https://github.com/Zhuo-Cao/FlashVTG.*

## 1. Introduction

The increasing prevalence of video content across various platforms has amplified the need for advanced video analysis techniques, particularly in the context of Video Temporal Grounding (VTG). The task of VTG involves accurately identifying specific video segments that correspond to given natural language descriptions, a capability that is crucial for applications such as Moment Retrieval (MR), event detection, and Highlight Detection (HD). Addressing these challenges is critical as it directly impacts the performance and usability of systems that rely on video understanding. For instance, the ability to precisely localize and retrieve short yet significant moments within videos can enhance user experiences in applications ranging from video editing to automated video surveillance. Moreover, improving the ranking of the predictions ensures that the first retrieved moment is more accurate and contextually relevant, thereby reducing errors in downstream tasks.

Despite advancements in video temporal grounding, current methods remain limited, particularly in short moment retrieval in complex, densely packed video sequences. DETR [3]-based models [16, 33, 42], though effective, underperform in short moment retrieval due to their reliance on sparse and limited decoder queries, which tend to overlook short moments. However, simply increasing the number of queries significantly increases the computational complexity of the methods. Moreover, relying solely on isolated predicted moments for comparison and ranking can lead to suboptimal results, especially when fine-grained distinctions are required.

In this paper, we introduce a Temporal Feature layering architecture (FlashVTG) to solve VTG tasks, including MR and HD. We first develop a Temporal Feature Layering module to extract and integrate video features across multiple temporal scales, enabling a more nuanced and comprehensive representation of video content. Subsequently, we introduce an Adaptive Score Refinement Module to rank predicted moments, enhancing the confidence scores of moments by integrating context from adjacent moments and multi-temporal-scale features. Together, these components

*Equal Contribution
†Corresponding Authors

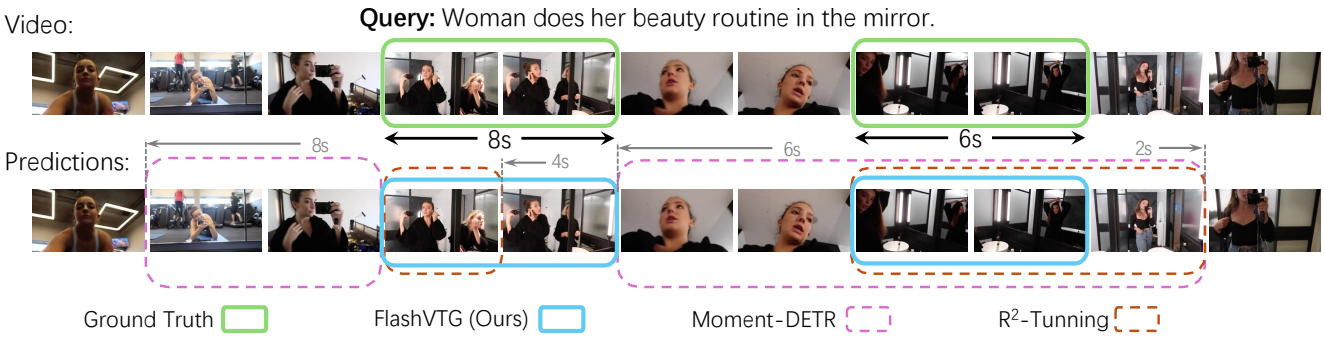


Figure 1. **Comparison of Model Performance on the Moment Retrieval Task** using video query pair from QVHighlights [16]. Ground Truth consists of two short moments, both of which are precisely retrieved by our model. In contrast, Moment-DETR [16], the established benchmark, and R2-Tuning [29], the previously leading method, failed to accurately retrieve the designated moments.

enhance the ability to accurately predict moments of varying durations, particularly short moments (see Fig 1), which are often problematic for previous methods.

Significant performance discrepancies are observed in DETR-based methods [15, 32], with mAP scores around 50% for moments longer than 10 seconds and a drastic drop to approximately 10% for shorter moments. These findings motivate us to develop a Temporal Feature Layering module for moment localization across various moment durations. We discard the conventional decoder structure and instead utilizes the Temporal Feature Layering module. By doing so, it addresses the inherent issue of sparse decoder queries in DETR-based methods, significantly improving the accuracy of moment retrieval without introducing additional training complexity.

For each predicted moment, a corresponding confidence score helps the model select the best prediction. However, we found that previous methods typically generate the confidence score only based on the current predicted moment, making it difficult to distinguish between closely adjacent moments when the video content is very similar. Therefore, we design the Adaptive Score Refinement Module. It enhances the model's ability to generate accurate confidence scores by leveraging both intra-scale and inter-scale features. This adaptive mechanism evaluates predicted moments not only across different feature scales but also within adjacent predicted moments on the same scale. Additionally, to further enhance accuracy for short moments, we introduced a novel Clip-Aware Score Loss, which applies labels from the Highlight Detection task to the Moment Retrieval task, offering fine-grained supervision that was previously unexplored.

To validate the effectiveness of FlashVTG, we conducted extensive experiments on widely adopted VTG benchmarks. Integrating FlashVTG with InternVideo2 [48], SlowFast [6], and CLIP [36] yield remarkable results for moment retrieval and highlight detection.

These results not only underscore the robustness of FlashVTG but also demonstrate its superiority over state-of-the-art methods, with performance improvements of 2.7%, 2.2%, and 11.9% respectively in MR, positioning FlashVTG as a leading approach in VTG tasks. Our contributions are as follows:

- We propose FlashVTG, a novel architecture that significantly enhances Video Temporal Grounding by employing a strategic integration of Temporal Feature Layering and Adaptive Score Refinement.
- We design a Temporal Feature Layering (TFL) Module that replaces the traditional decoder. TFL is designed to overcome sparse query limitations and improve retrieval without additional training.
- We introduce an Adaptive Score Refinement (ASR) Module that selects predicted moments using both intra-scale and inter-scale features, enhancing first-moment accuracy and highlight detection.

## 2. Related Work

**Video Temporal Grounding (VTG).** In the field of text-guided Video Temporal Grounding [15, 16, 20], the objective is to identify specific temporal segments within a video based on given natural language descriptions. Essentially, this task requires models to discern and quantify the associations between the content of the video and natural language descriptions across different time stamps. Subsequently, we will provide a detailed exposition of two specific VTG tasks: moment retrieval and highlight detection.

**Moment Retrieval (MR).** The goal of this task is to identify the start and end points of a video segment based on a natural language query. Given the limitations of current datasets, it is typically assumed that a single ground truth (GT) segment exists. If multiple GT segments are available, the one with the highest Intersection Over Union (IOU) with the predicted moment is selected as the GT. Existing MR methods primarily fall into two categories:

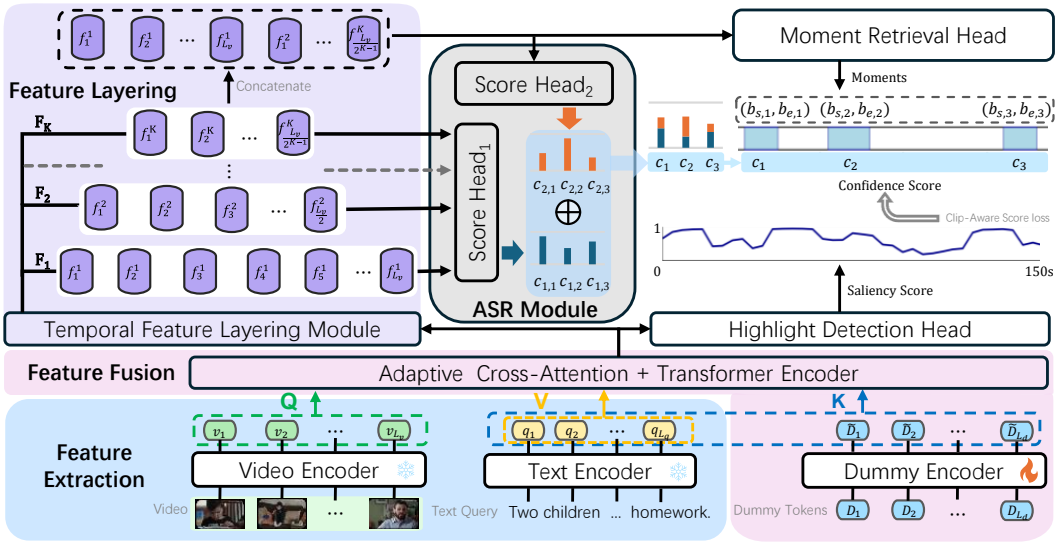


Figure 2. **Overview of the FlashVTG Framework.** As depicted in the blue section below, input videos and queries are first processed through frozen encoders to extract corresponding video and text features. These text features, concatenated with encoded Dummy tokens to form the Key, are merged with video text features in the Feature Fusion module to create Fused Features. These are then directed into the Temporal Feature Module and the HD Head, producing $K$ temporal scale features ($f_j^i$ refers to the token at the $j^{th}$ position of the feature $F_i$ at the $i^{th}$ scale.) and saliency scores, respectively. These features and their concatenated forms are input into the Adaptive Score Refinement Module, generating intra- and inter-scale confidence scores $c$ (shown here with three examples). Lastly, the Moment Retrieval Head uses all fused features for boundary prediction, outputting timestamps $(b_s, b_e)$ for the start and end points.

proposal-based [24, 26, 39, 50, 53, 62, 63] and proposal-free methods [4, 16, 23, 25, 31, 34, 59].

Proposal-based methods operate in two stages: generating candidate moments and treating retrieval as a matching problem. These methods can be further classified into sliding window [1,7,8,27,58], anchor-based [45,58,60,63], and proposal-generated [24, 39, 50, 53, 62] approaches. In contrast, proposal-free methods [23] use regression to directly predict the start and end points of relevant moments.

**Highlight Detection (HD).** Similar to MR, HD aims to assess the importance of video segments, but here it involves determining the relevance of each segment to a given query. Each clip is assigned a score reflecting its correlation with the query. From a methodological perspective, early research on HD predominantly utilized ranking-based approaches [10, 22, 28, 38, 43]. In recent years, methods [14, 16, 32, 33] based on the DETR have exhibited notable performance improvements.

Although both MR and HD are closely related, they were not studied concurrently until the publication of the QVHighlights dataset [16], which prompted numerous works integrating these tasks. The first study to merge them, Moment-DETR [16], was introduced alongside the QVHighlights dataset. This model serves as a baseline capable of addressing both MR and HD challenges. Subsequent research has built on this framework with various enhancements: UMT [30] incorporates audio information into query generation; QD-DETR [33] enhances performance through negative video-text pair relationships; UVCOM [51] improves outcomes by considering the varying importance of different tasks at different granularities; CG-DETR [32] enhances alignment between queries and moments by considering the clipwise cross-modal interactions; and LLMEPET [15] integrates LLM encoders into existing Video Moment Retrieval (VMR) architectures. Beyond DETR-based methods, the recent $R^2$-Tunning [29] framework, which leverages transfer learning from images to videos, has shown potential capabilities for the VTG task. However, these methods struggle to handle multi-scale information, particularly for short moment retrieval, leading to performance bottlenecks. in FlashVTG, we process multi-scale information separately, achieving improvements in both MR and HD tasks.

## 3. Methodology

### 3.1. Problem Formulation

In Video Temporal Grounding (VTG) tasks, we represent the input video and the extracted features as $\mathcal{V}$ and $\mathbb{V} = \{v_i\}_{i=1}^{L_v}$, respectively, where each $v_i \in \mathbb{R}^{D_v}$. Here, $L_v$ denotes the number of video clips and $D_v$ represents the feature dimension of each clip. For the natural language query $\mathcal{Q}$, we represent it as a set $\mathbb{Q} = \{q_i\}_{i=1}^{L_q}$, where each $q_i \in \mathbb{R}^{D_q}$ is the $i$-th word token in the query. Here, $L_q$ de-

notes the word number of query and $D_q$ indicates the feature dimension of each word token.

For Moment Retrieval (MR), the objective of the model is to predict the start point $b_s$ and the end point $b_e$ of the moment based on the input $V$ and the query. In the common practices, the output is a set of tuples $(b_s, b_e)$ representing the temporal boundaries of these moments. Additionally, the model generates confidence scores $c$ to rank these predictions, although these scores are not required by the task itself. Thus, the model produces: $\{(b_{s,i}, b_{e,i}, c_i) \mid i = 1, 2, \ldots, n\}$, with confidence scores $\{c_1, c_2, \ldots, c_n\}$ used for ranking.

In the Highlight Detection task, the model outputs a sequence of clip-wise saliency scores $S = \{s_i\}_{i=1}^{L_v}$, where each score $s_i \in [0, 1]$ quantifies the relevance of the respective clip to the query.

### 3.2. Overall Framework

As illustrated in Figure 2, the input video and query are initially encoded into video and query features using frozen Feature Encoders. In addition to extracting the original video and text features, FlashVTG incorporates a dummy token to supplement semantic information beyond the query, enhancing alignment between the video and text.

Following the encoding process, the video and query features are integrated within the Feature Fusion Module, enabling precise alignment of the two modalities. The fused features are subsequently processed in two separate streams. For Highlight Detection, the HD head transforms the fused features into saliency scores, while the Temporal Feature Layering Module expands the features across multiple granularities.

The Adaptive Score Refinement Module then evaluates these multi-scale features to generate a confidence score, $s$. Concurrently, the Moment Retrieval prediction head outputs predictions across different scales, producing a series of tuples $(b_s, b_e)$. Finally, the predicted moments and confidence scores are combined into $(b_{s,i}, b_{e,i}, c_i) \mid i = 1, 2, \ldots, n$, which specify the predicted start and end points of moments, along with their associated confidence scores.

### 3.3. Feature Extraction and Fusion Module

**Feature Extraction**. Consistent with previous methodologies, we employ the CLIP [36] image encoder and SlowFast [6] to extract clip-level video features $\mathbb{V}$, and use the CLIP text encoder along with GloVe [35] to derive word-level text features $\mathbb{Q}$. Specifically, raw videos are segmented into clips at predetermined FPS, such as 0.5 or 1, using frozen pretrained models as feature extractors. This process transforms each clip into a distinct video feature $v_i$. Similarly, each word in the query is encoded into corresponding text features $q_i$.

**Feature Fusion.** To align video and query features for subsequent processing, we project them into the same dimensional space $d$ using two MLPs before fusion. We adopt the Adaptive Cross Attention (ACA) module from CG-DETR [32], which extends the traditional Cross Attention by incorporating the ability to supplement query information. A known limitation of standard Cross Attention, particularly with the softmax operation, is that it may not perfectly align all video clips with the query, as the query cannot fully encapsulate the semantic scope of an untrimmed video. In ACA module, we use learnable dummy tokens, $D = [D_1, ..., D_{L_d}]$, where $L_d$ is a hyperparameter. These tokens capture semantic information beyond the original query, complementing its content. The encoded dummy tokens $\tilde{D}$ are concatenated with the query features $\mathbb{Q}$ to form the Key, with the video features $\mathbb{V}$ as the Query and the query features $\mathbb{Q}$ as the Value, as shown in Equation (1).

$$\begin{aligned} \textit{Query} &= [p_Q(v_1), \ldots, p_Q(v_{L_v})] \\ \textit{Key} &= [p_K(q_1), \ldots, p_K(q_{L_q}), p_K(\tilde{D_1}), \ldots, p_K(\tilde{D_{L_d}})] \\ \textit{Value} &= [p_V(q_1), \ldots, p_V(q_{L_q})] \end{aligned} \tag{1}$$

Here, $p_Q(\cdot)$, $p_K(\cdot)$, and $p_V(\cdot)$ denote the projection functions used to transform inputs into the Query, Key, and Value formats, respectively. These Query, Key, Value are input into the ACA module to obtain the fused features $F \in \mathbb{R}^{L_v \times d}$, as shown in Equations (2) and (3).

$$F = \text{ACA}(v_i) = \sum_{j=1}^{L_q} W_{i,j} \odot V_j; \tag{2}$$

$$W_{i,j} = \frac{\exp\left(\frac{Q_i \odot K_j}{\sqrt{d}}\right)}{\sum_{k=1}^{L_q+L_d} \exp\left(\frac{Q_i \odot K_k}{\sqrt{d}}\right)}, \tag{3}$$

where $F$ denotes the fused feature and $\odot$ stands for the dot product. While $Q$, $K$, and $V$ represent the Query, Key, and Value, respectively. Through this adaptive cross-attention mechanism, the model more effectively aligns the query with the relevant video segments.

Following the ACA module, the fused features $F$ are passed through a Transformer Encoder to further refine the multi-modal representations. This additional processing step enables the model to enhance the interaction between the video and query features, allowing for a more comprehensive understanding of the temporal and semantic relationships across the entire sequence.

**Highlight Detection Head**. The refined features output from the Transformer Encoder are further processed to generate clip-level saliency scores $s$ for highlight detection. We aggregate the fused features to form a global contextual representation, which is then combined with the original fused

features through a linear projection. The operation can be expressed as:

$$s = \frac{\sum_i (\mathbf{W}_1 f_i \circ \mathbf{W}_2 g)}{\sqrt{d}}, \tag{4}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ represent two linear projection functions, $f_i$ denotes the $i$-th clip-wise fused features, and $g$ represents the global contextual features. The symbol $\circ$ represents the Hadamard operation. This operation identifies the key clips in the video, yielding a sequence of scores that quantify the relevance of each clip to the overall content.

### 3.4. Temporal Feature Layering Module

**Temporal Feature Layering.** This module transforms the fused features into a feature pyramid to enhance the model's capability to process features at various granularities. The feature pyramid is a commonly used structure in computer vision for extracting and utilizing vision features across multiple scales [9, 47]. This architecture emulates the hierarchical processing of the human visual system, while enabling the network to effectively perceive and process video information of varying lengths.

Building on this insight, we incorporate layered feature implementations into DETR-based models to overcome their limitations in short moment retrieval. This module separates moments of varying lengths, enabling the subsequent ASR module (Sec. 3.5) to provide more detailed supervision for short moments prediction. Specifically, we apply 1D convolution operations with various strides to the fused features $F$, constructing a feature pyramid composed of features at different granularities. Predictions of moment locations are then made at these various granularities using a uniform prediction head.

Let $F \in \mathbb{R}^{L_v \times d}$ represent the input fused feature, where $L_v$ is the length of the features and $d$ is the dimensionality of the features. The process of temporal feature layering can be represented as:

$$F_k = \begin{cases} F, & \text{if } k = 1, \\ \text{Conv1D}^{k-1}(F, \text{stride} = 2), & \text{if } k = 2, 3, \ldots, K. \end{cases}$$

By applying the temporal feature layering, we obtain a sets of features at different granularities, which can be represented as:

$$F_k \in \mathbb{R}^{\frac{L_v}{2^{k-1}} \times d}, \quad k = 1, 2, \ldots, K.$$

Here, $\{F_k | k = 1, 2, \ldots, K\}$ represent the feature pyramid obtained from the original fused feature $F$ through multiple convolution operations. This multi-scale processing approach enables the model to capture and process information at different temporal resolutions, thereby adapting to scene changes at various scales. As referenced in Sec. 4.5, this module significantly improves the model's ability to retrieve moments of varying lengths. This operation paves the way for enhanced supervision of short moment retrieval in subsequent steps.

**Moment Prediction Head.** This module primarily adjusts and reduces the feature dimension to 2, and through a series of transformations, it yields the predicted start and end points of moments. The specific process can be expressed as:

$$B_k = \left(\sigma\left(\text{Conv1D}\left(\sigma\left(\text{Conv1D}(F_k)\right)\right)^\top\right)\right)^\top \times C_k. \tag{5}$$

Here, $B_k \in \mathbb{R}^{\frac{L_v}{2^{k-1}} \times 2}$ represents a set of boundaries at scale $k$, $\sigma(\cdot)$ denotes the ReLU activation function. The term $C_k$ refers to a learnable parameter corresponding to each scale, which is used to adjust the influence of different scales on boundary prediction.

### 3.5. Adaptive Score Refinement Module

Adaptive Score Refinemen Module used to assign a confidence score $c \in [0, 1]$ to each predicted moment in MR. This score indicates the extent to which the given query is relevant to the predicted boundary. Compared with the previous method of generating scores on a single scale feature, we leverage both intra-scale and inter-scale scores to improve the final predictions.

For each level of the feature pyramid $F_k$, intra-scale scores are first generated through a score head, producing outputs corresponding to the varying dimensions of the pyramid levels. These outputs are then concatenated into a unified tensor along the length dimension. These steps are shown in Equation (6) and Equation (7), respectively.

$$c_k = \text{ScoreHead}_1(F_k) \in \mathbb{R}^{\frac{L_v}{2^{k-1}} \times 1}, k = 1, 2, \ldots, K. \tag{6}$$

$$c_{\text{intra}} = \text{Concat}(c_1, c_2, \ldots, c_K). \tag{7}$$

Our score head utilizes a 2D convolutional network with a kernel size of $1 \times 5$, which is effectively equivalent to a 1D convolution. Simultaneously, as shown in Equation (8), inter-scale scores are computed by concatenating the features from all pyramid levels and passing them through another score head, resulting in a tensor that matches the dimensions of the intra-scale output.

$$c_{\text{inter}} = \text{ScoreHead}_2(\text{Concat}(F_1, F_2, \ldots, F_K)). \tag{8}$$

The final prediction is obtained by a weighted combination of the intra-scale and inter-scale scores:

$$c_{\text{final}} = x \cdot c_{\text{intra}} + (1 - x) \cdot c_{\text{inter}}. \tag{9}$$

Here, the learnable weighting factor $x$ enables adaptive adjustment between $c_{\text{intra}}$ and $c_{\text{inter}}$, thereby resulting in a more comprehensive score prediction.

| Method | test | | | | | val | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | R1 | | mAP | | | R1 | | mAP | | |
| | @0.5 | @0.7 | @0.5 | @0.75 | Avg. | @0.5 | @0.7 | @0.5 | @0.75 | Avg. |
| M-DETR [16] *NeurIPS'21* | 52.89 | 33.02 | 54.82 | 29.17 | 30.73 | 53.94 | 34.84 | - | - | 32.20 |
| UMT [30] *CVPR'22* | 56.23 | 41.18 | 53.83 | 37.01 | 36.12 | 60.26 | 44.26 | 56.70 | 39.90 | 38.59 |
| QD-DETR [33] *CVPR'23* | 62.40 | 44.98 | 62.52 | 39.88 | 39.86 | 62.68 | 46.66 | 62.23 | 41.82 | 41.22 |
| UniVTG [20] *ICCV'23* | 58.86 | 40.86 | 57.60 | 35.59 | 35.47 | 59.74 | - | - | - | 36.13 |
| EaTR [14] *ICCV'23* | - | - | - | - | - | 61.36 | 45.79 | 61.86 | 41.91 | 41.74 |
| MomentDiff [18] *NeurIPS'23* | 57.42 | 39.66 | 54.02 | 35.73 | 35.95 | - | - | - | - | - |
| TR-DETR [42] *AAAI'23* | 64.66 | 48.96 | 63.98 | 43.73 | 42.62 | 67.10 | 51.48 | 66.27 | 46.42 | 45.09 |
| TaskWeave [56] *CVPR'24* | - | - | - | - | - | 64.26 | 50.06 | 65.39 | 46.47 | 45.38 |
| CG-DETR [32] *Arxiv'24* | 65.43 | 48.38 | 64.51 | 42.77 | 42.86 | 67.35 | 52.06 | 65.57 | 45.73 | 44.93 |
| UVCOM [51] *CVPR'24* | 63.55 | 47.47 | 63.37 | 42.67 | 43.18 | 65.10 | 51.81 | - | - | 45.79 |
| SFABD [13] *WACV'24* | - | - | 62.38 | 44.39 | 43.79 | - | - | - | - | - |
| LLMEPET [15] *MM'24* | 66.73 | 49.94 | 65.76 | 43.91 | 44.05 | 66.58 | 51.10 | - | - | 46.24 |
| $R^2$-Tunning [29] *ECCV'24* | 68.03 | 49.35 | 69.04 | 47.56 | 46.17 | 68.71 | 52.06 | - | - | 47.59 |
| **FlashVTG** (Ours) | 66.08 | 50.00 | 67.99 | 48.70 | 47.59 | 69.03 | 54.06 | 68.44 | 52.12 | 49.85 |
| **FlashVTG**† (Ours) | **70.69** | **53.96** | **72.33** | **53.85** | **52.00** | **73.10** | **57.29** | **72.75** | **54.33** | **52.84** |

Table 1. Performance comparison on the QVHighlights [16] Test and Validation Splits. In each column, the highest score is highlighted in **bold**, and the second highest score is underlined. The notation † indicates that the backbone used is InternVideo2 [48].

### 3.6. Training Objectives

In FlashVTG, we employ a series of loss functions to ensure the model converges towards the desired objectives. For MR, we use Focal Loss [21], L1 Loss, and Clip-Aware Score Loss to respectively optimize the classification labels, boundaries, and clip-level confidence scores of the predicted moments. For HD, we utilize SampledNCE Loss [29] and Saliency Loss to optimize the saliency scores for each clip. The overall loss can be expressed as:

$$\mathcal{L}_{\text{overall}} = \lambda_{\text{Reg}}\mathcal{L}_{\text{L1}} + \lambda_{\text{Cls}}\mathcal{L}_{\text{Focal}} + \lambda_{\text{CAS}}\mathcal{L}_{\text{CAS}} + \lambda_{\text{SNEC}}\mathcal{L}_{\text{SNCE}} + \lambda_{\text{Sal}}\mathcal{L}_{\text{Sal}},$$

where each $\lambda_*$ represents the corresponding weight for each loss component. Due to the space limitation, we will focus on the Clip-Aware Score Loss, further details on the other loss functions can be found in the supplementary materials.

**Clip-Aware Score Loss.** This Loss is designed to align the predicted confidence scores with the target saliency labels. Given the predicted clip-wise moment confidence score $c_{\text{final}}$ and the target saliency scores $s_{\text{gt}}$, we first normalize both sets of scores using min-max normalization to get $\hat{c}_{\text{final}}$ and $\hat{s}_{\text{gt}}$. The loss is then computed as the mean squared error between $\hat{c}_{\text{final}}$ and $\hat{s}_{\text{gt}}$:

$$\mathcal{L}_{\text{CAS}} = \text{MSE}(\hat{c}_{\text{final}}, \hat{s}_{\text{gt}}). \tag{10}$$

This loss encourages the model to produce confidence scores that not only match the target labels but also adhere to the relative distribution of saliency across clip-level, thereby specifically enhancing the model's performance when predicting short moments.

| Method | R@0.3 | R@0.5 | R@0.7 | mIoU |
|---|---|---|---|---|
| 2D-TAN [62] | 40.01 | 27.99 | 12.92 | 27.22 |
| VSLNet [61] | 35.54 | 23.54 | 13.15 | 24.99 |
| Moment-DETR [16] | 37.97 | 24.67 | 11.97 | 25.49 |
| UniVTG [20] | 51.44 | 34.97 | 17.35 | 33.60 |
| CG-DETR [32] | 52.23 | 39.61 | 22.23 | 36.48 |
| $R^2$-Tuning [29] | 49.71 | 38.72 | **25.12** | 35.92 |
| LLMEPET [15] | 52.73 | - | 22.78 | 36.55 |
| **FlashVTG** (Ours) | **53.71** | **41.76** | 24.74 | **37.61** |

Table 2. Performance Evaluation on TACoS [37]. All these methods utilize SlowFast [6] and CLIP [36] as backbones for TACoS. The highest score in each column is **bolded**, and the second highest is underlined.

## 4. Experiments

### 4.1. Datasets

We evaluated our model on five VTG task datasets, including QVHighlights, TACoS, Charades-STA, TVSum, and YouTube-HL.

QVHighlights [16] is the most widely used dataset for MR and HD tasks, as it provides annotations for both tasks. This dataset marked the beginning of a trend where these two tasks are increasingly studied together. It includes more than 10,000 daily vlogs and news videos with text queries. Our main experiments were conducted on this dataset, and we provide comprehensive comparisons with other methods on it. Charades-STA [7] and TACoS [37] were used to evaluate the model's performance on MR, containing daily

| Method | Backbone | R1@0.5 | R1@0.7 |
|---|---|---|---|
| SAP [5] | VGG | 27.42 | 13.36 |
| TripNet [11] | VGG | 36.61 | 14.50 |
| MAN [60] | VGG | 41.24 | 20.54 |
| 2D-TAN [62] | VGG | 40.94 | 22.85 |
| FVMR [17] | VGG | 42.36 | 24.14 |
| UMT† [30] | VGG | 48.31 | 29.25 |
| QD-DETR [33] | VGG | 52.77 | 31.13 |
| TR-DETR [42] | VGG | 53.47 | 30.81 |
| CG-DETR [32] | VGG | **55.22** | 34.19 |
| **FlashVTG** (ours) | VGG | 54.25 | **37.42** |
| 2D-TAN [62] | SF+C | 46.02 | 27.50 |
| VSLNet [61] | SF+C | 42.69 | 24.14 |
| Moment-DETR [16] | SF+C | 52.07 | 30.59 |
| QD-DETR [33] | SF+C | 57.31 | 32.55 |
| UniVTG [20] | SF+C | 58.01 | 35.65 |
| TR-DETR [42] | SF+C | 57.61 | 33.52 |
| CG-DETR [32] | SF+C | 58.44 | 36.34 |
| LLMEPET [15] | SF+C | - | 36.49 |
| **FlashVTG** (Ours) | SF+C | **60.11** | **38.01** |
| **FlashVTG** (Ours) | IV2 | **70.32** | **49.87** |

Table 3. Experimental results on the Charades-STA test set. "SF+C" refers to SlowFast R-50 [6] combined with CLIP-B/32 [36], and "IV2" denotes InternVideo2-6B [48].c Methods marked with "†" use audio features.

activities and cooking-related content, respectively. The other two datasets, TVSum [41] and YouTube-HL [43], are sports-related and were used for HD evaluation.

### 4.2. Evaluation Metrics

We follow previous works [15, 16, 29] and adopt consistent evaluation metrics: R1@X, mAP, and mIoU for MR, and mAP and Hit@1 for HD. Specifically, R1@X stands for "Recall 1 at X", which refers to selecting the predicted moment with the highest confidence score and checking whether its IoU with any ground truth moment exceeds the threshold X. If it does, the prediction is considered positive, and R@X is then calculated at thresholds $X \in \{0.3, 0.5, 0.7\}$. For MR, mAP serves as the primary metric, representing the mean of the average precision (AP) across all queries at thresholds [0.5:0.05:0.95]. We also use mean Intersection over Union (mIoU) to evaluate the average overlap between predicted moments and ground truth segments. For HD, mAP remains the key evaluation metric, with Hit@1 used to assess the hit ratio for the highest-scored clip.

### 4.3. Implementation Details

As in previous methods [16, 30, 33], we primarily used video and text features extracted by CLIP [36] and SlowFast [6] across all five datasets for a fair comparison. To further verify the generalizability of our model, we incorporated video and text features extracted by InternVideo2 [48] and LLaMA [44] for QVHighlights and Charades-STA, as well as features extracted by VGG [40] and GloVe [35] for Charades-STA. All feature dimensions were set to 256. The number of attention heads in the Feature Fusion Module was set to 8, with $K = 4, 5$ layers in temporal feature layering, and the number of 2D convolutional layers in the score head was set to 2. AdamW was used as the optimizer, and the NMS threshold during inference was set to 0.7. Training was conducted on a single Nvidia RTX 4090 GPU, taking approximately one and a half hours for 150 epochs on QVHighlights.

| Method | test | | val | |
|---|---|---|---|---|
| | mAP | HIT@1 | mAP | HIT@1 |
| M-DETR [16] | 35.69 | 55.60 | 35.65 | 55.55 |
| UMT [30] | 38.18 | 59.99 | 39.85 | 64.19 |
| QD-DETR [33] | 38.94 | 62.40 | 39.13 | 63.03 |
| UniVTG [20] | 38.20 | 60.96 | 38.83 | 61.81 |
| EaTR [14] | - | - | 37.15 | 58.65 |
| CG-DETR [32] | 40.33 | 66.21 | 40.79 | 66.71 |
| $R^2$-Tuning [29] | 40.75 | 64.20 | 39.45 | 64.13 |
| LLMEPET [15] | 40.33 | 65.69 | 40.52 | 65.03 |
| **FlashVTG** (Ours) | 41.07 | 66.15 | 41.39 | 67.61 |
| **FlashVTG**† (Ours) | **44.09** | **71.01** | **44.15** | **72.90** |

Table 4. Experimental results for Highlight detection on the QVHighlights [16]. All models used the same backbone, except for $R^2$-Tuning [29], which used CLIP [36] as the backbone, and FlashVTG marked with "†", which used InternVideo2 [48] as the backbone.

### 4.4. Comparison Results

The comparison of MR experimental results is shown in Tables 1, 2, 3, 7, where FlashVTG achieved state-of-the-art (SOTA) performance on nearly all metrics. Table 1, 7 presents the MR results on the QVHighlights [16]. Table 1 demonstrates that combining FlashVTG with a newly selected backbone [48] resulted in a significant performance improvement, evidenced by a 5.8% increase in mAP on the test set. Even when using the same backbone, mAP improved by 1.4%. Table 7 shows that FlashVTG improved the mAP of short moment retrieval to 125% of the previous SOTA method.These performance gains on one of the most widely used datasets demonstrate the effectiveness of FlashVTG compared to contemporaneous approaches.

Tables 2 and 3 present experimental results on two other MR datasets, TaCos and Charades-STA. Similarly, FlashVTG significantly outperformed previous methods on almost all metrics, achieving either SOTA or the second-highest performance. On Charades-STA, performance improvements were observed across all three different backbones, further validating the robustness of FlashVTG.

The comparison of HD experimental results is shown

| Method | VT | VU | GA | MS | PK | PR | FM | BK | BT | DS | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LIM-S [52] | 55.9 | 42.9 | 61.2 | 54.0 | 60.4 | 47.5 | 43.2 | 66.3 | 69.1 | 62.6 | 56.3 |
| Trailer [46] | 61.3 | 54.6 | 65.7 | 60.8 | 59.1 | 70.1 | 58.2 | 64.7 | 65.6 | 68.1 | 62.8 |
| SL-Module [54] | 86.5 | 68.7 | 74.9 | 86.2 | 79.0 | 63.2 | 58.9 | 72.6 | 78.9 | 64.0 | 73.3 |
| PLD [49] | 84.5 | 80.9 | 70.3 | 72.5 | 76.4 | 87.2 | 71.9 | 74.0 | 74.4 | 79.1 | 77.1 |
| UniVTG [20] | 83.9 | 85.1 | 89.0 | 80.1 | 84.6 | 81.4 | 70.9 | 91.7 | 73.5 | 69.3 | 81.0 |
| $R^2$-tunning [29] | 85.0 | 85.9 | 91.0 | 81.7 | **88.8** | 87.4 | 78.1 | 89.2 | 90.3 | 74.7 | 85.2 |
| LLMEPET [15] | **90.8** | 91.9 | **94.2** | **88.7** | 85.8 | 90.4 | **78.6** | **93.4** | 88.3 | 78.7 | **88.1** |
| MINI-Net† [12] | 80.6 | 68.3 | 78.2 | 81.8 | 78.1 | 65.8 | 57.8 | 75.0 | 80.2 | 65.5 | 73.2 |
| TCG† [57] | 85.0 | 71.4 | 81.9 | 78.6 | 80.2 | 75.5 | 71.6 | 77.3 | 78.6 | 68.1 | 76.8 |
| Joint-VA† [2] | 83.7 | 57.3 | 78.5 | 86.1 | 80.1 | 69.2 | 70.0 | 73.0 | **97.4** | 67.5 | 76.3 |
| CO-AV† [19] | **90.8** | 72.8 | 84.6 | 85.0 | 78.3 | 78.0 | 72.8 | 77.1 | 89.5 | 72.3 | 80.1 |
| UMT† [30] | 87.5 | 81.5 | 88.2 | 78.8 | 81.4 | 87.0 | 76.0 | 86.9 | 84.4 | 79.6 | 83.1 |
| **FlashVTG** (Ours) | 88.32 | **94.33** | 91.5 | 87.7 | 87.08 | **91.12** | 74.7 | **93.4** | 90.3 | **81.7** | 88.0 |

Table 5. Highlight detection results (Top-5 mAP) on TV-Sum [41] across different class. "†" denotes the methods that utilize the audio modality.

| Method | Dog | Gym. | Par. | Ska. | Ski. | Sur. | Avg. |
|---|---|---|---|---|---|---|---|
| RRAE [55] | 49.0 | 35.0 | 50.0 | 25.0 | 22.0 | 49.0 | 38.3 |
| GIFs [10] | 30.8 | 33.5 | 54.0 | 55.4 | 32.8 | 54.1 | 46.4 |
| LSVM [43] | 60.0 | 41.0 | 61.0 | 62.0 | 36.0 | 61.0 | 53.6 |
| LIM-S [52] | 57.9 | 41.7 | 67.0 | 57.8 | 48.6 | 65.1 | 56.4 |
| SL-Module [54] | 70.8 | 53.2 | 77.2 | 72.5 | 66.1 | 76.2 | 69.3 |
| QD-DETR [33] | 72.2 | **77.4** | 71.0 | 72.7 | 72.8 | 80.6 | 74.4 |
| LLMEPET [15] | 73.6 | 74.2 | 72.5 | **75.3** | **73.4** | 82.5 | 75.3 |
| MINI-Net† [12] | 58.2 | 61.7 | 70.2 | 72.2 | 58.7 | 65.1 | 64.4 |
| TCG† [57] | 55.4 | 62.7 | 70.9 | 69.1 | 60.1 | 59.8 | 63.0 |
| Joint-VA† [2] | 64.5 | 71.9 | 80.8 | 62.0 | 73.2 | 78.3 | 71.8 |
| UMT† [30] | 65.9 | 75.2 | **81.6** | 71.8 | 72.3 | 82.7 | 74.9 |
| UniVTG† [20] | 71.8 | 76.5 | 73.9 | 73.3 | 73.2 | 82.2 | 75.2 |
| **FlashVTG** (Ours) | **76.5** | 76.1 | 69.4 | 74.1 | 73.1 | **83.0** | **75.4** |

Table 6. Highlight detection performances (mAP) on YouTube-HL [43] across different class. "†" indicates the usage of audio modality.

| Method | MR-short-mAP | mAP |
|---|---|---|
| UMT [30] | 5.02 | 38.59 |
| UniVTG [20] | 8.64 | 36.13 |
| CG-DETR [32] | 10.58 | 44.93 |
| LLMEPET [15] | 10.96 | 46.24 |
| $R^2$-Tuning [29] | 12.62 | 47.86 |
| **FlashVTG** (Ours) | **15.73** | **49.85** |

Table 7. Performance comparison for short moment (<10s) retrieval on QVHighlights [16]. All methods, except $R^2$-tunning [29], use SlowFast [6] + Clip [36] as the backbone.

| Component | + TFL Module | | ✓ | ✓ |
|---|---|---|---|---|
| | + ASR Module | | | ✓ |
| Metrics | MR-R1@0.5 | 68.26 | 72.39 | **73.10** |
| | MR-R1@0.7 | 51.35 | 56.19 | **57.29** |
| | MR-mAP | 46.84 | 52.47 | **52.84** |
| | HL-mAP | 42.20 | 43.63 | **44.15** |
| | HL-Hit1 | 69.23 | 71.81 | **72.90** |

Table 8. Ablation study on different components. TFL stands for Temporal Feature Layering, ASR stands for Adaptive Score Refinement.

in Tables 4, 5, 6. FlashVTG achieved SOTA performance on both QVHighlights and YouTube-HL, and outperformed all methods that used additional audio features on TVSum, reaching performance comparable to the SOTA. This indicates that FlashVTG can deliver excellent performance even with relatively smaller datasets.

### 4.5. Ablation Study

We conducted an ablation study on the QVHighlights validation set to verify the effectiveness of FlashVTG. This dataset is currently the most widely used VTG benchmark and supports both MR and HD tasks, making it the most suitable for our research. We used the original single-layer feature and a method that generates confidence scores based solely on single predicted moments as our baseline.

**Effect of different components.** As shown in Table 8, we studied the effects of the Temporal Feature Layering and Adaptive Score Refinement modules. It can be observed that adding the TFL module to the baseline improved performance in both the MR and HD tasks, with the mAP for MR increasing by nearly 6%. Building on this, we further incorporated the ASR module to refine the confidence scores of the predicted moments. The improvements in MR-mAP, R@0.5, and R@0.7 indicate that the overall precision was maintained while enhancing Recall at 1, meaning the recall for the first predicted moment improved. Additionally, using the saliency score label from the HD task as a supervision signal indirectly boosted HD performance, with mAP increasing by 0.5% and Hit@1 improving by 1%.

## 5. Conclusion

This paper introduced FlashVTG, a novel architecture for Video Temporal Grounding tasks. The proposed Temporal Feature Layering and Adaptive Score Refinement modules improve the retrieval and selection of more accurate moment predictions across varying lengths. Extensive experiments on five VTG benchmarks demonstrate that FlashVTG consistently outperforms state-of-the-art methods in both Moment Retrieval and Highlight Detection, achieving substantial improvements. These results validate the robustness and effectiveness of our approach, establishing FlashVTG as a leading solution for VTG tasks.

## Acknowledgment

This work is supported by Australian Research Council (ARC) Discovery Project DP230101753.

## References


[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017. 3

[2] Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Joint visual and audio learning for video highlight detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8127–8137, 2021. 8

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[4] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. Rethinking the bottom-up framework for query-based video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10551–10558, 2020. 3

[5] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8199–8206, 2019. 7

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2, 4, 6, 7, 8

[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. 3, 6

[8] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. Mac: Mining activity concepts for language-based temporal localization. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 245–253. IEEE, 2019. 3

[9] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7036–7045, 2019. 5

[10] Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1001–1009, 2016. 3, 8

[11] Meera Hahn, Asim Kadav, James M Rehg, and Hans Peter Graf. Tripping through time: Efficient localization of activities in videos. *arXiv preprint arXiv:1904.09936*, 2019. 7

[12] Fa-Ting Hong, Xuanteng Huang, Wei-Hong Li, and Wei-Shi Zheng. Mini-net: Multiple instance ranking network for video highlight detection. In *European Conference on Computer Vision*, pages 345–360. Springer, 2020. 8

[13] Cheng Huang, Yi-Lun Wu, Hong-Han Shuai, and Ching-Chun Huang. Semantic fusion augmentation and semantic boundary detection: A novel approach to multi-target video moment retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6783–6792, 2024. 6

[14] Jinhyun Jang, Jungin Park, Jin Kim, Hyeongjun Kwon, and Kwanghoon Sohn. Knowing where to focus: Event-aware transformer for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13846–13856, 2023. 3, 6, 7

[15] Yiyang Jiang, Wengyu Zhang, Xulu Zhang, Xiaoyong Wei, Chang Wen Chen, and Qing Li. Prior knowledge integration via LLM encoding and pseudo event regulation for video moment retrieval. In *ACM Multimedia 2024*, 2024. 2, 3, 6, 7, 8

[16] 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. 1, 2, 3, 6, 7, 8

[17] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 7

[18] Pandeng Li, Chen-Wei Xie, Hongtao Xie, Liming Zhao, Lei Zhang, Yun Zheng, Deli Zhao, and Yongdong Zhang. Momentdiff: Generative video moment retrieval from random to real. *Advances in neural information processing systems*, 36, 2024. 6

[19] Shuaicheng Li, Feng Zhang, Kunlin Yang, Lingbo Liu, Shinan Liu, Jun Hou, and Shuai Yi. Probing visual-audio representation for video highlight detection via hard-pairs guided contrastive learning. *arXiv preprint arXiv:2206.10157*, 2022. 8

[20] 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*, pages 2794–2804, 2023. 2, 6, 7, 8

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[22] Chunxi Liu, Qingming Huang, and Shuqiang Jiang. Query sensitive dynamic web video thumbnail generation. In *2011 18th IEEE international conference on image processing*, pages 2449–2452. IEEE, 2011. 3

[23] Daizong Liu, Xiaoye Qu, Xing Di, Yu Cheng, Zichuan Xu, and Pan Zhou. Memory-guided semantic learning network for temporal sentence grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1665–1673, 2022. 3

[24] Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. Context-aware biaffine localizing network for temporal sentence grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11235–11244, 2021. 3

[25] Daizong Liu, Xiaoye Qu, and Wei Hu. Reducing the vision and language bias for temporal sentence grounding. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4092–4101, 2022. 3

[26] Daizong Liu, Xiaoye Qu, Xiao-Yang Liu, Jianfeng Dong, Pan Zhou, and Zichuan Xu. Jointly cross-and self-modal graph attention network for query-based moment localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4070–4078, 2020. 3

[27] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive moment retrieval in videos. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 15–24, 2018. 3

[28] Wu Liu, Tao Mei, Yongdong Zhang, Cherry Che, and Jiebo Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3707–3715, 2015. 3

[29] Ye Liu, Jixuan He, Wanhua Li, Junsik Kim, Donglai Wei, Hanspeter Pfister, and Chang Wen Chen. $r^2$-tuning: Efficient image-to-video transfer learning for video temporal grounding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2, 3, 6, 7, 8

[30] 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*, pages 3042–3051, 2022. 3, 6, 7, 8

[31] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. Debug: A dense bottom-up grounding approach for natural language video localization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5144–5153, 2019. 3

[32] 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. 2, 3, 4, 6, 7, 8

[33] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033, 2023. 1, 3, 6, 7, 8

[34] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10810–10819, 2020. 3

[35] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 4, 7

[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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 4, 6, 7, 8

[37] 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. 6

[38] Mrigank Rochan, Mahesh Kumar Krishna Reddy, Linwei Ye, and Yang Wang. Adaptive video highlight detection by learning from user history. In *European conference on computer vision*, pages 261–278. Springer, 2020. 3

[39] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 200–216, 2018. 3

[40] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[41] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015. 7, 8

[42] Hao Sun, Mingyao Zhou, Wenjing Chen, and Wei Xie. Tr-detr: Task-reciprocal transformer for joint moment retrieval and highlight detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4998–5007, 2024. 1, 6, 7

[43] Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 787–802. Springer, 2014. 3, 7, 8

[44] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 7

[45] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12168–12175, 2020. 3

[46] Lezi Wang, Dong Liu, Rohit Puri, and Dimitris N Metaxas. Learning trailer moments in full-length movies with co-contrastive attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 300–316. Springer, 2020. 8

[47] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 5

[48] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024. 2, 6, 7

[49] Fanyue Wei, Biao Wang, Tiezheng Ge, Yuning Jiang, Wen Li, and Lixin Duan. Learning pixel-level distinctions for video highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3073–3082, 2022. 8

[50] Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao. Boundary proposal network for two-stage natural language video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2986–2994, 2021. 3

[51] Yicheng Xiao, Zhuoyan Luo, Yong Liu, Yue Ma, Hengwei Bian, Yatai Ji, Yujiu Yang, and Xiu Li. Bridging the gap: A unified video comprehension framework for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18709–18719, 2024. 3, 6

[52] Bo Xiong, Yannis Kalantidis, Deepti Ghadiyaram, and Kristen Grauman. Less is more: Learning highlight detection from video duration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1258–1267, 2019. 8

[53] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9062–9069, 2019. 3

[54] Minghao Xu, Hang Wang, Bingbing Ni, Riheng Zhu, Zhenbang Sun, and Changhu Wang. Cross-category video highlight detection via set-based learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7970–7979, 2021. 8

[55] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE international conference on computer vision*, pages 4633–4641, 2015. 8

[56] Jin Yang, Ping Wei, Huan Li, and Ziyang Ren. Task-driven exploration: Decoupling and inter-task feedback for joint moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18308–18318, 2024. 6

[57] Qinghao Ye, Xiyue Shen, Yuan Gao, Zirui Wang, Qi Bi, Ping Li, and Guang Yang. Temporal cue guided video highlight detection with low-rank audio-visual fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7950–7959, 2021. 8

[58] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[59] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10287–10296, 2020. 3

[60] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1247–1257, 2019. 3, 7

[61] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*, 2020. 6, 7

[62] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12870–12877, 2020. 3, 6, 7

[63] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. Cross-modal interaction networks for query-based moment retrieval in videos. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 655–664, 2019. 3