

Scanning Only Once: An End-to-end Framework for Fast Temporal Grounding in Long Videos

Yulin Pan¹ Xiangteng He^{2,4*} Biao Gong¹ Yiliang Lv¹
Yujun Shen³ Yuxin Peng^{2,4} Deli Zhao^{1*}

¹Alibaba Group ²Wangxuan Institute of Computer Technology, Peking University

³Ant Group ⁴National Key Laboratory for Multimedia Information Processing, Peking University

Abstract

Video temporal grounding aims to pinpoint a video segment that matches the query description. Despite the recent advance in short-form videos (e.g., in minutes), temporal grounding in long videos (e.g., in hours) is still at its early stage. To address this challenge, a common practice is to employ a sliding window, yet can be inefficient and inflexible due to the limited number of frames within the window. In this work, we propose an end-to-end framework for fast temporal grounding, which is able to model an hours-long video with **one-time** network execution. Our pipeline is formulated in a coarse-to-fine manner, where we first extract context knowledge from non-overlapped video clips (i.e., anchors), and then supplement the anchors that highly response to the query with detailed content knowledge. Besides the remarkably high pipeline efficiency, another advantage of our approach is the capability of capturing long-range temporal correlation, thanks to modeling the entire video as a whole, and hence facilitates more accurate grounding. Experimental results suggest that, on the long-form video datasets MAD and Ego4d, our method significantly outperforms state-of-the-arts, and achieves $14.6\times / 102.8\times$ higher efficiency respectively. Project can be found at <https://github.com/afcedf/SOONet.git>.

1. Introduction

Video temporal grounding [5, 6, 13, 18, 19, 29, 31, 32], which aims to localize a specific moment in the video corresponding to a natural language description, has found its applications in many real-world scenarios, such as video retrieval [11, 22], video highlight detection [23, 24], and video question answering [9, 26].

Despite the rapid advance in recent years, existing methods for temporal grounding usually target short-form

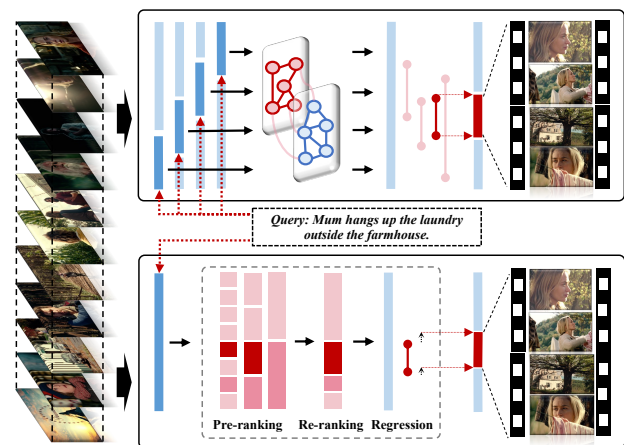


Figure 1. **Pipeline comparison** between sliding window-based methods (top) [19, 29, 31, 32] and our SOONet (bottom). It is noteworthy that the sliding window pipeline requires repeated inference on *overlapped clips* and the final *result aggregation*, while ours can deliver the result with *one-time* network execution. Detailed discussion can be found in Sec. 4.5.

videos (e.g., in minutes) and characterize the input video with a small number of frames (e.g., 128) [19, 27, 29, 30, 31, 32]. When it comes to the case of long-form video temporal grounding (LVTG) [6, 18], however, temporally downsampling a video (e.g., in hours) to so few frames could cause severe information loss and further result in drastic performance degradation [6].

A straightforward solution is to reorganize a long video to a sequence of short videos using a sliding window and perform temporal grounding within each window [6, 8, 18]. However, such a solution as shown in the top half of Fig. 1 has three main drawbacks. (1) Inference inefficiency: The overlap between adjacent windows brings redundant computations. Besides, the large amounts of highly overlapped predictions cause post-processing (e.g., non-maximum suppression) time-consuming. It is noteworthy by saying efficiency, we mean *pipeline efficiency* instead of

*Corresponding authors.

model efficiency, which considers the total execution time from data input to final results output, including data pre-processing, model forward running and post-processing.¹ (2) Training insufficiency: The network with a sliding window can only scan the video contents within a local time range at one time, yet ignore the long-range temporal correlation. (3) Prediction inflexibility: The prediction is restricted inside a single window, making it hard to generalize to segments with long duration.

In this work, we propose an anchor-based end-to-end framework, termed as **SOONet**, which facilitates efficient and accurate LVTG by **Scanning a long-form video Only Once**. As shown in the bottom half of Fig. 1, SOONet follows a pipeline of *pre-ranking*, *re-ranking*, *regression*, via leveraging both the inter-anchor context knowledge and the intra-anchor content knowledge.

Specifically, we first produce non-overlapped anchor sequence via anchor partition layer, then three procedures are implemented to obtain final predictions: (1) Multi-scale context-based anchor features are acquired by modeling inter-anchor context knowledge via cascaded temporal swin transformer blocks [12]. Meanwhile, a coarse anchor rank is obtained via sorting the context-based matching scores with respect to query. (2) Content-based anchor features and a content-enhanced anchor rank can be obtained by supplementing anchors with detailed intra-anchor content knowledge. We pick out the top- m anchors that highly corresponds to query from each scale to form an anchor subset, then implement re-ranking within subset to reduce the computational complexity. (3) Boundary regression is adopted to achieve flexible predictions, leveraging both inter-anchor and intra-anchor knowledge. To take full advantage of the abundant cross-modal semantic relationship in long videos, we sample one video with a batch of queries grounded in this video at one training step, then optimize the full-length anchor rank and query rank simultaneously with the help of proposed dual-form approximate rank loss, which achieves superior cross-modal alignment.

Extensive experiments are conducted on two long-form video datasets, *i.e.*, MAD [18] and Ego4d [6]. Our method significantly outperforms state-of-the-arts, and achieves $14.6\times / 102.8\times$ higher pipeline efficiency, which demonstrate the effectiveness.

2. Related Work

2.1. Short-form Video Temporal Grounding

Existing methods mainly focus on short-form video temporal grounding and can be categorized into *proposal-based* and *proposal-free* methods. Methods in proposal-based category adopt a two-stage pipeline, which first generate proposal candidates by various proposal generation

methods, such as sliding window and proposal generation network, then they rank these candidates and output the proposal with the highest matching score as final prediction. [5] propose CTRL, a pioneer work in video grounding. CTRL produces various-length proposal candidates via sliding window and uses the visual-textual fusion modules combined with three operators, *i.e.*, add, multiply and fully-connected layer, to obtain multi-modal fused representation. MAN [30] and SCDM [27] leverage multiple cascaded temporal convolution layers to generate proposal candidates hierarchically. TGN [2] temporally captures the evolving fine-grained frame-by-word interactions and uses pre-set anchors to produce multi-scale proposal candidates ending at each time step. Subsequently [15, 21, 33, 34] follow the anchor-based framework and propose various multi-modal reasoning strategies to achieve precise moment localization. In addition, 2D-TAN [32] enumerate all possible segments as proposal candidates and convert them into 2D feature map, then a temporal adjacent network is proposed to obtain multi-modal representation and encode the video context information. Following this, [19, 20, 35] design more complicated cross-modal reasoning strategies to learn the video-language semantic alignment from both coarse and fine-grained granularities.

Methods in proposal-free category predict the start and end boundaries by computing the time pair directly, or output the confidence scores of being the start and end positions of target moment for each snippet in video. [28] propose ABLR, which performs cross-modal reasoning with a multi-modal co-attention interaction modules and outputs target moments by feeding the multi-modal features to regressor. Attention weight-based regression and attention feature-based regression are considered together to achieve precise boundary regression. Concurrently, DRN [29] considers the data imbalance issue and only uses the frame in ground-truth moment to mitigate the sparsity issue. LGI [13] aligns the video and language from phrase-level and propose a local-global interaction network that models the cross-modal relationship considering local and global context information simultaneously.

However, directly applying these methods on long-form videos results in drastic performance degradation, as temporally downsampling a long video to so few frames causes severe temporal information loss.

2.2. Long-form Video Temporal Grounding

Recently MAD [18] and Ego4d [6] pose the challenge of long-form video temporal grounding, and give some baselines that integrate sliding window and temporal downsampling into some short video-fit methods, such as 2D-TAN [32], VLG-Net [19] and VSLNet [31]. However, all these methods achieve inferior performance, considering both accuracy and efficiency. Recently [8] propose CONE,

¹The concrete explanations of each part can be found in Sec. 4.5.

which pre-filters the candidate windows to address the inference inefficiency and learns the cross-modal alignment from proposal-level and frame-level. Nevertheless, it adopts sparse sampling strategy at training stage, which does not explore the potential of long-form video adequately.

3. Method

This section presents a detailed introduction to our proposed framework. As depicted in Fig. 2, our method takes a long-form video and a sentence query as input, and predicts the video moment that is semantically related to the query in an end-to-end manner. Specifically, our framework consists of three modules: (1) *Pre-ranking with Anchor Knowledge* aims to encode the inter-anchor context by employing cascaded temporal swin transformer blocks. Then, a coarse anchor rank is obtained by sorting the context-based matching scores concerning the query. (2) *Re-ranking with Frame Knowledge* is designed to model the intra-anchor content knowledge, and calculate the content-based matching scores concerning the query. The anchor candidates are re-ranked by summing the context-based and content-based matching scores. (3) *Boundary Regression* aims to adjust anchor boundaries, leveraging both inter-anchor context and intra-anchor content. Our method outputs the adjusted boundaries of the top- n anchors as the final predictions.

3.1. Feature Extractor

Given an untrimmed video $V = \{f_t\}_{t=1}^T$ and a sentence query $Q = \{w_m\}_{m=1}^M$, where T and M represent the number of frames and words respectively, the LVTG task requires to localize the target moment (τ_s, τ_e) that corresponds to query. To achieve this, we adopt off-the-shelf pretrained models to extract visual features $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\} \in \mathbb{R}^{N \times D}$ as well as textual features $\mathbf{Q} = \{\mathbf{q}_{\text{cls}}, \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M\} \in \mathbb{R}^{(M+1) \times D}$. N, M represent the numbers of extracted frame features and word features respectively, and D represents the feature dimension. The query feature \mathbf{q} is extracted in different ways depended on the type of pretrained model. For models pretrained with multi-modalities (e.g., CLIP [16]), we take out the class token embedding \mathbf{q}_{cls} as query feature. While for other models (e.g., BERT [3]), we pass the word embeddings through a trainable LSTM [7] layer to acquire the query feature. We then feed the video features \mathbf{V} and query feature \mathbf{q} into our network for next process.

3.2. Pre-ranking with Anchor Knowledge

Multi-scale anchor generation. Due to the computational complexity of global self-attention is quadratic to the sequence length, the standard transformer is heavily computational on modeling full-length frame sequences of long-form video. To mitigate the computational burden,

we first employs a single convolutional layer to produce non-overlapping base anchors from successive frames. The formulation is as follow:

$$\mathbf{E}^0 = \text{Conv1d}(\mathbf{V}), \quad (1)$$

where $\mathbf{E}^0 \in \mathbb{R}^{\frac{N}{C_0} \times D}$, and C_0 denotes the length of base anchor. Then we adopt L cascaded temporal swin transformer blocks with pooling layers to encode inter-anchor context knowledge and obtain multi-scale context-based anchor features $\mathbf{E} = [\mathbf{E}^1; \mathbf{E}^2; \dots; \mathbf{E}^L]$, where L represents the number of scales. Each anchor feature $\mathbf{e}_i \in \mathbf{E}$ corresponds to a unique clip proposal (t_s^i, t_e^i) . For the anchors of l -th scale, the corresponding anchor length is

$$C_l = C_{l-1}r_l, \quad (2)$$

where r_l denotes the receptive field of l -th pooling layer.

Temporal swin transformer block. We have incorporated the shifted window-based self-attention approach, as proposed in Swin Transformer [12], into 1-dimensional sequence encoding. This technique effectively implements self-attention in local windows, while also establishes connections between consecutive windows to bolster the modeling capabilities. In this way, the computational complexity is linearly scaling with the sequence length. Specifically, each temporal swin transformer block consists of a local-window self-attention layer (W-MSA), a shifted-window self-attention layer (SW-MSA) and two multi-layer perceptrons (MLP), which can be formulated as:

$$\begin{aligned} \hat{z}^l &= \text{W-MSA}(\text{LN}(z^l)) + z^l, \\ \tilde{z}^l &= \text{MLP}(\text{LN}(\hat{z}^l)) + \hat{z}^l, \\ \tilde{z}^{l+1} &= \text{SW-MSA}(\text{LN}(\tilde{z}^l)) + \tilde{z}^l, \\ z^{l+1} &= \text{MLP}(\text{LN}(\tilde{z}^{l+1})) + \tilde{z}^{l+1}, \end{aligned} \quad (3)$$

where LN represents the LayerNorm [1] operation.

For each context-based anchor feature $\mathbf{e}_i \in \mathbf{E}$, the context-based matching score is obtained by computing the cosine similarity between anchor feature and query feature, then scaling it to $[0, 1]$ via Sigmoid function:

$$S_{\text{ctx}}^i = \text{Sigmoid}\left(\frac{\mathbf{e}_i \cdot \mathbf{q}}{\|\mathbf{e}_i\| \|\mathbf{q}\|}\right), 1 \leq i \leq \sum_{l=1}^L \frac{N}{C_l}. \quad (4)$$

Finally a coarse anchor rank can be acquired by sorting S_{ctx} in a descending order.

3.3. Re-ranking with Frame Knowledge

To mitigate the temporal information loss caused by the anchor partition and pooling operation in *pre-ranking module*, the *re-ranking module* models the detailed content inside anchors and re-rank anchor candidates. Given the

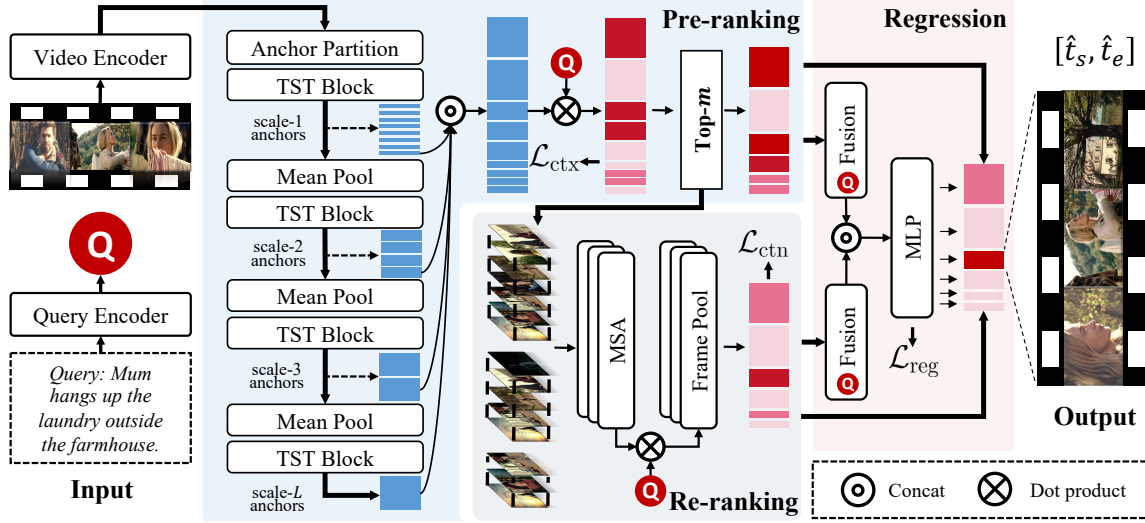


Figure 2. **Overall architecture of our algorithm.** The whole framework consists of three modules: the *pre-ranking module* aims to obtain coarse anchor rank by modeling inter-anchor context; the *re-ranking module* aims to obtain content-enhanced anchor rank by supplementing anchors with detailed content; the *regression module* aims to adjust anchor boundaries.

coarse anchor rank, we first collect the indices of the top- m anchors from each scale separately to set up an anchor subset, then for i -th anchor of l -th scale in this subset, we fetch the intra-anchor frame features $\mathbf{V}_i = \{\mathbf{v}_i^k\}_{k=1}^{C_l}$ and adopt standard multi-head self-attention module (MSA) to model the intra-anchor frame correlation:

$$\hat{\mathbf{V}}_i = \text{MSA}(\text{LN}(\mathbf{V}_i + f_{\text{pos}}(\mathbf{V}_i)) + \mathbf{V}_i, \quad (5)$$

where f_{pos} is trainable positional embeddings used to inject positional information. The content-based matching score of i -th anchor is obtained by first computing cosine similarity between each frame feature and query feature, then pooling frame-wise similarities and scaling it to $[0, 1]$ via Sigmoid function:

$$S_{\text{ctn}}^i = \text{Sigmoid}\left(\text{Pool}\left(\frac{\hat{\mathbf{V}}_i \cdot \mathbf{q}}{\|\hat{\mathbf{V}}_i\| \|\mathbf{q}\|}\right)\right), 1 \leq i \leq mL. \quad (6)$$

We sum the context-based score and content-based score as the final matching score for re-ranking:

$$S = \tilde{S}_{\text{ctx}} + S_{\text{ctn}}, \quad (7)$$

where $\tilde{S}_{\text{ctx}} \subseteq S_{\text{ctx}}$ is the context-based scores of subset.

3.4. Boundary Regression

To achieve flexible localization, the *boundary regression module* is employed to adjust anchor boundaries inward or outward. For i -th anchor of l -th scale in anchor subset, given context-based anchor feature \mathbf{e}_i and content-based anchor feature $\hat{\mathbf{V}}_i$, we fuse them with query to obtain multi-modal fused feature, and pass it through a MLP header to

predict the start and end bias:

$$\begin{aligned} \mathbf{f}^i &= [\mathbf{e}_i \odot \mathbf{q}; \text{Att}(\hat{\mathbf{V}}_i) \odot \mathbf{q}], \\ (\delta_s^i, \delta_e^i) &= \text{MLP}(\mathbf{f}^i), \end{aligned} \quad (8)$$

where \odot is element-wise multiplication. $\text{Att}(\hat{\mathbf{V}}_i)$ represents the self-attentive accumulation of $\hat{\mathbf{V}}_i$:

$$\begin{aligned} \alpha_i^k &= \mathbf{W} \hat{\mathbf{v}}_i^k, \\ \mathbf{a}_i &= \text{Softmax}([\alpha_i^1, \alpha_i^2, \dots, \alpha_i^{C_l}]), \\ \text{Att}(\hat{\mathbf{V}}_i) &= \sum_{k=1}^{C_l} \mathbf{a}_i^k \hat{\mathbf{v}}_i^k, \end{aligned} \quad (9)$$

where $\mathbf{W} \in \mathbb{R}^{1 \times D}$ is a learnable weight matrix. Then given the original anchor boundaries (t_s^i, t_e^i) , we add the predicted start and end bias respectively to obtain adjusted boundaries:

$$\begin{aligned} \hat{t}_s^i &= t_s^i + \delta_s^i \times (t_e^i - t_s^i), \\ \hat{t}_e^i &= t_e^i + \delta_e^i \times (t_e^i - t_s^i). \end{aligned} \quad (10)$$

Finally we output the adjusted boundaries (\hat{t}_s, \hat{t}_e) of the top- n anchors as final predictions.

3.5. Training

Two loss terms are adopted to optimize the network: (1) Cross-modal alignment loss $\mathcal{L}_{\text{align}}$, and (2) Boundary regression loss \mathcal{L}_{reg} . The total loss is a weighted combination of the two loss terms:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{align}} + \lambda_2 \mathcal{L}_{\text{reg}}, \quad (11)$$

where λ_1 and λ_2 are hyper-parameters used to control the contribution of $\mathcal{L}_{\text{align}}$ and \mathcal{L}_{reg} respectively.

3.5.1 Cross-modal Alignment Loss

We define the cross-modal alignment loss as a combination of context-based alignment loss \mathcal{L}_{ctx} and content-based alignment loss \mathcal{L}_{ctn} :

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{ctx}} + \mathcal{L}_{\text{ctn}}. \quad (12)$$

For \mathcal{L}_{ctx} and \mathcal{L}_{ctn} , we propose a dual-form approximate rank loss that adopts two ApproxNDCG [14] loss terms to optimize the anchor rank and query rank simultaneously. We first revisit the ApproxNDCG loss and introduce the dual-form approximate rank loss, then give out formal definitions of \mathcal{L}_{ctx} and \mathcal{L}_{ctn} .

ApproxNDCG loss. Given large amounts of anchor candidates, we aim to obtain such an anchor rank: the anchor semantically related to query should be ranked in front of the unrelated ones. To achieve this goal, rather than point-wise or pair-wise rank losses which are commonly used in existing methods, we adopt the list-wise ApproxNDCG loss to optimize the anchor rank from the global perspective:

$$\mathcal{L}_{\text{ar}}(S, y) = 1 - Z_m^{-1} \sum_{i=1}^K \frac{2^{y_i} - 1}{\log(1 + \hat{\pi}_i)}, \quad (13)$$

where S denotes the matching scores of anchor candidates, K is the number of anchor candidates and Z_m refers to the discounted cumulative gain of the best rank. y_i represents the matching degree between the i -th anchor and query that equals to the temporal IoU of their bounding boxes:

$$y_i = \text{IoU}((t_s^i, t_e^i), (\tau_s, \tau_e)). \quad (14)$$

$\hat{\pi}_i$ is a differentiable approximation to the rank of i -th anchor:

$$\hat{\pi}_i = 1 + \sum_{u \neq i} \frac{\exp(-\alpha(S_i - S_u))}{1 + \exp(-\alpha(S_i - S_u))}, \quad (15)$$

where α denotes a temperature parameter. For each anchor, the ApproxNDCG loss compares it with all other anchors to decide its rank, taking full advantage of the semantic relationship in long-form videos.

Dual-form approximate rank loss. Besides the anchor rank optimization, considering the unique characteristic of long-form video dataset, we introduce an “one video with batch queries” data sampling strategy that samples one video with a batch of queries grounded in this long video at one training step, and employ another ApproxNDCG loss to optimize the query rank simultaneously:

$$\mathcal{L}_{\text{dar}}(S^a, S^q, y) = \mathcal{L}_{\text{ar}}(S^a, y) + \mathcal{L}_{\text{ar}}(S^q, y), \quad (16)$$

where S^a and S^q denotes the matching scores of anchor candidates and query candidates, respectively. Now, we

define the context-based alignment loss \mathcal{L}_{ctx} and content-based alignment loss \mathcal{L}_{ctn} as :

$$\begin{aligned} \mathcal{L}_{\text{ctx}} &= \mathcal{L}_{\text{dar}}(S_{\text{ctx}}^a, S_{\text{ctx}}^q, y), \\ \mathcal{L}_{\text{ctn}} &= \mathcal{L}_{\text{dar}}(S_{\text{ctn}}^a, S_{\text{ctn}}^q, y), \end{aligned} \quad (17)$$

where S_{ctx}^a and S_{ctn}^a represents the full-length context-based anchor matching scores and mL -length content-based anchor matching scores respectively. Likewise, S_{ctx}^q and S_{ctn}^q denotes the context-based and content-based query matching scores respectively.

3.5.2 Boundary Regression Loss

We define the boundary regression loss as follows:

$$\mathcal{L}_{\text{reg}} = \frac{1}{mL} \sum_{i=1}^{mL} \mathcal{L}_{\text{iou}}((\hat{t}_s^i, \hat{t}_e^i)), \quad (18)$$

where the $(\hat{t}_s^i, \hat{t}_e^i)$ is the adjusted boundaries of i -th anchor. IoU loss [25] is adopted to regress the start and end bias between anchor boundaries and groundtruth moment:

$$\mathcal{L}_{\text{iou}}((\hat{t}_s^i, \hat{t}_e^i)) = -\ln(\text{IoU}((\hat{t}_s^i, \hat{t}_e^i), (\tau_s, \tau_e))). \quad (19)$$

4. Experiments

4.1. Datasets

We conduct experiments on two long-form video datasets MAD [18] (avg. 110.8 min / video) and Ego4d [6] (avg. 25.7 min / video), in which videos are much longer than those in previous datasets, such as ActivityNet Captions [10] (avg. 2.0 min / video) and Charades-STA [17] (avg. 0.5 min / video).

MAD is a large-scale benchmark for long-form video temporal grounding, which contains over 384K natural language queries that derived from high-quality audio description of mainstream movies and grounded in over 1.2K hours of videos with very low coverage (an average duration of 4.1s). The length of videos in MAD ranges from 47 minutes to 202 minutes, which are orders of magnitude longer than previous datasets.

Ego4d is an egocentric video dataset, containing 3,670 hours of daily-life activity videos collected by 931 world-wide participants. The **Ego4d-NLQ** is the official subtask of Ego4d which is to retrieve the most relevant video moment from truncated video clips, given a natural language question that generated via filling pre-defined query templates. However, the average duration of video clips is only 8.25 minutes, which is too short to be used as LVTG evaluation benchmark. To verify the effectiveness of our method on long-form video grounding, we introduce a new evaluation setting and name it **Ego4d-Video-NLQ**, where we replace the truncated video clips with full-length

Table 1. Performance on the test set of MAD dataset. All of three baselines are sliding window-based methods.

Model	IoU = 0.1					IoU = 0.3					IoU = 0.5				
	R@1	R@5	R@10	R@50	R@100	R@1	R@5	R@10	R@50	R@100	R@1	R@5	R@10	R@50	R@100
VLG-Net [19]	3.64	11.66	17.89	39.78	51.24	2.76	9.31	14.65	34.27	44.87	1.65	5.99	9.77	24.93	33.95
CLIP [16]	6.57	15.05	20.26	37.92	47.73	3.13	9.85	14.13	28.71	36.98	1.39	5.44	8.38	18.80	24.99
CONE [8]	8.90	20.51	27.20	43.36	-	6.87	16.11	21.53	34.73	-	4.10	9.59	12.82	20.56	-
SOONet (Ours)	11.26	23.21	30.36	50.32	58.66	9.00	19.64	26.00	44.78	53.18	5.32	13.14	17.84	32.59	39.62

Table 2. Performance on the val set of Ego4d dataset, under Ego4d-NLQ and Ego4d-Video-NLQ settings. Noted that 2D-TAN and CONE are sliding window-based methods while VSLNet is downsampling-based method.

Model	IoU = 0.3		IoU = 0.5	
	R@1	R@5	R@1	R@5
Ego4d-NLQ (avg. 8.25 min / video)				
2D-TAN [32]	5.04	12.89	2.02	5.88
VSLNet [31]	5.45	10.74	3.12	6.63
CONE ² [8]	10.40	22.74	5.03	11.87
SOONet (Ours)	8.00	22.40	3.76	11.09
Ego4d-Video-NLQ (avg. 25.7 min / video)				
2D-TAN [32]	1.70	4.59	0.82	2.77
VSLNet [31]	1.57	4.44	0.75	2.22
SOONet (Ours)	3.90	10.71	1.80	5.09

video, therefore the average duration of videos reaches 25.7 minutes. We report the performance on the validation set of Ego4d, under both Ego4d-NLQ and Ego4d-Video-NLQ settings.

4.2. Metrics

Following [8, 18], we adopt the standard metric “Recall@ n , IoU= m ” (R@ n - m) for evaluation. Specifically, it represents the percentage of testing samples that have at least one grounding prediction whose IoU with groundtruth is larger than m among top- n predictions.

4.3. Implementation Details

Following [18], we use CLIP [16] to extract visual features and textual features for MAD dataset. We set $C_0 = 10$, $L = 4$ for multi-scale anchor generation. λ_1, λ_2 are set to 1 and 20 respectively. m is set to 100 for filtering. The temperature α for \mathcal{L}_{ctx} and \mathcal{L}_{ctn} are both set to 0.01. We train the network for 100k steps with an initial learning rate of 0.001, and decay it by a factor of 10 after 40k steps. When training, we set batch size as 32 (1 video with 32 queries grounded in this video at one step) and use AdamW as the optimizer. The feature dimension D is set to 512.

²As for CONE, its code has not been released as of the completion of this paper, so for fair comparison we only report its result on Ego4d-NLQ provided by its original paper.

Table 3. Details about sliding windows. Numbers in brackets stand for the window duration (or stride duration). Number of windows is averaged over the entire validation set.

Method	# Frames per Window	Slide Stride	# Windows
CLIP	128 (25.6s)	64 (12.8s)	566.3
VLG-Net	128 (25.6s)	64 (12.8s)	566.3
2D-TAN	75 (40s)	37.5 (20s)	89.5

For Ego4d-NLQ and Ego4d-Video-NLQ, we use the pre-extracted SlowFast features [4] and Bert features [3] as the visual and textual features, following [6]. We set $C_0 = 1$, $L = 7$ on Ego4d-NLQ and $C_0 = 6$, $L = 2$ on Ego4d-Video-NLQ. λ_1, λ_2 are set to 1 and 5 respectively. m is set to 200. We train the network for 30k steps with an initial learning rate of 0.0001, and decay it by 10 after 15k steps. Early stop is adopted to mitigate overfitting. All experiments are implemented on one A100 GPU with 80GB memory.

4.4. Accuracy Comparison with SOTAs

We first compare our model with several state-of-the-art methods. Tab. 1 reports the performance results on long video dataset MAD (the average video duration is around **110.8 minutes**) with three methods: CLIP [16], VLG-Net [19] and CONE [8]. All of them are sliding window-based methods. From Tab. 1 we can observe that our method outperforms all other methods, achieving **2.13%** and **1.22%** performance gains, in terms of R@1-0.3 and R@1-0.5 respectively. Thanks to modeling the entire video as a whole, our method can capture long-range temporal correlation, and learn cross-modal alignment with abundant context information, which facilitates more accurate grounding. We also conduct experiments on Ego4d dataset and summarize the results on Tab. 2. We first compare performance under Ego4d-NLQ setting with three methods: 2D-TAN [32], VSLNet [31] and CONE [8]. It is noted that, to improve the efficiency and save the memory cost, our approach does not include fine-grained cross-modality semantic alignment (e.g., align text and image from the token level), which can be very computation intensive along with the video length growing. Although such fine-grained knowledge can be beneficial to short-form video temporal

Table 4. **Efficiency comparison** on MAD and Ego4d-Video-NLQ. The total time is a summation of time of three parts: pre-processing, model forward, and post-processing. For fair comparison, we feed one video and one query to the system at each time, and report the total running time over the entire test set. Compared to sliding window-based methods, which require repeated inference on overlapped clips and the final result aggregation (*i.e.*, post-processing), our one-time execution pipeline is far more efficient.

Dataset	Method	Method Type	Trainable Parameters	FLOPs	GPU Memory	Execution Time (second)			
						Pre	Model	Post	Total
MAD	CLIP [16]	Slide Window	0	0.2G	2.9G	630.9s	15.7s	6741.2s	7387.8s
	VLG-Net [19]	Slide Window	5,330,435	1757.3G	20.0G	3350.3s	10659.0s	15546.7s	29556.0s
	SOONet (Ours)	End-to-end	22,970,947	70.2G	2.4G	42.4s	438.9s	23.7s	505.0s
Ego4d-Video-NLQ	2D-TAN [32]	Slide Window	86,773,761	6160.0G	3.9G	442.1s	2625.3s	1153.7s	4225.2s
	VSLNet [31]	Down-sampling	866,435	0.9G	2.8G	10.7s	56.9s	1.4s	69.0s
	SOONet (Ours)	End-to-end	25,203,779	5.4G	1.8G	16.7s	23.6s	0.8s	41.1s

grounding, we use a unified framework for both short video and long video settings. Tab. 2 suggests that our method achieves competitive performance on Ego4d-NLQ, even though it tests on short-form videos (the average video duration is **8.25 minutes**). We then test the performance on Ego4d-Video-NLQ, where the average video duration is **25.7 minutes**. We re-implement the 2D-TAN and VSLNet with the public code released by [6]: it combines the 2D-TAN with sliding window to fit long-form video while adopts the downsampling strategy for VSLNet to reduce the sequence length to 128. From Tab. 2 we observe our SOONet achieves **2.20%** / **0.98%** performance gains in terms of R@1-0.3 and R@1-0.5 respectively, which demonstrates the effectiveness of our method on long-form video temporal grounding.

4.5. Efficiency Comparison with SOTAs

To evaluate the efficiency of our method, we compare SOONet with 3 sliding window-based methods (*i.e.*, CLIP, VLG-Net and 2D-TAN) and 1 downsampling-based method (*i.e.*, VSLNet) on MAD and Ego4d-Video-NLQ. We report the details about sliding windows in Tab. 3, regarding CLIP and VLG-Net on MAD dataset, and 2D-TAN on Ego4d-Video-NLQ dataset. Numbers in brackets stand for the window duration (or stride duration). Number of windows is averaged over the entire validation set. Recall that the code of CONE is not publicly available so we can't make a fair comparison with it. As mentioned in Sec. 1, the efficiency here means pipeline efficiency, which considers the execution time of three parts: (1) *Pre-processing* (denoted as Pre). For all competing methods, the pre-processing time refers to the time of loading snippets' features to GPU memory. It is worth noting that the snippets' features are offline extracted and, therefore, are excluded from the pre-processing stage. Moreover, sliding window-based approaches require reorganizing the snippets' features into overlapped window features, which is more time-consuming than our method. (2) *Model forward* (denoted as Model), which refers to network calculation.

Table 5. Ablation study on various modules in SOONet. PR, RR, and BR denote Pre-ranking module, Re-ranking module, and Boundary Regression module, respectively

PR	RR	BR	IoU = 0.1		IoU = 0.3		IoU = 0.5	
			R@1	R@5	R@1	R@5	R@1	R@5
✓			9.41	20.68	7.07	17.02	4.10	11.08
✓	✓		10.17	21.94	7.65	17.98	4.43	11.37
✓		✓	10.79	22.37	8.52	18.73	4.79	12.00
✓	✓	✓	11.03	22.99	8.83	19.48	5.23	13.18

Table 6. Ablation study on dual-form approximate rank loss.

Loss	IoU = 0.1		IoU = 0.3		IoU = 0.5	
	R@1	R@5	R@1	R@5	R@1	R@5
\mathcal{L}_{bce}	0.05	0.51	0.01	0.10	0.00	0.01
\mathcal{L}_{nce}	5.26	13.65	4.09	10.90	2.32	6.73
\mathcal{L}_{ar}	10.08	22.02	8.15	18.47	4.80	12.04
\mathcal{L}_{dar}	11.03	22.99	8.83	19.48	5.23	13.18

(3) *Post-processing* (denoted as Post). For sliding window-based methods, it is necessary to first apply non-maximum suppression (NMS) to discard a significant number of highly overlapping predictions, and then obtain the final segments by sorting their scores. However, our approach minimizes the generation of numerous highly overlapped candidates, making the NMS operation optional. In all experiments conducted in this paper, we did not employ NMS in our method.

Tab. 4 reports the number of parameters, FLOPs, GPU memory usage of models and gives a detailed breakdown of execution time. For FLOPs and GPU memory usage, we measure them using same samples as input because they change with the length of input video. From Tab. 4 we observe the GPU memory usages of sliding window-based methods surpass our SOONet obviously, because batch inference on local windows is adopted to accelerate the model forward. For execution time, at each time we feed one video and one sentence to the system and report

the total execution time of each part separately over the entire test set. From Tab. 4 we observe that compared with sliding window-based methods, our SOONet makes huge improvement on pipeline efficiency, achieving $14.6\times / 58.5\times / 102.8\times$ higher inference speed, compared with CLIP, VLG-Net and 2D-TAN respectively. It is noteworthy that model FLOPs only affects the model forward time. Though CLIP contains only a matrix multiplication operation that needs few FLOPs, it suffers from both the slow pre-processing, which needs to split an entire video into lots of overlapped windows as well as gather window features, and the slow post-processing, which employs NMS (*i.e.*, non-maximum suppression) to abandon large amounts of highly overlapped predictions. In addition to the slow pre-processing and post-processing, another efficiency bottleneck of sliding window-based methods lies in the redundant computation on overlapped windows, which increases the model FLOPs greatly, causing the model forward part time-consuming. Compared with downsampling-based VSLNet, our SOONet achieves competitive inference speed whereas a far superior accuracy. Despite a bit more FLOPs, our network spends less time on model forward running than VSLNet. These results demonstrate the efficiency of our method.

4.6. Ablation Studies

Effectiveness of Each Module. We conduct experiments on MAD to verify the effectiveness of each module employed in our framework: (1) Pre-ranking with Anchor Knowledge, (2) Re-ranking with Frame Knowledge, and (3) Boundary Regression. We report the ablation results on Tab. 5, where PR, RR, BR represent the three modules respectively. Tab. 5 suggest that, equipped with Pre-ranking module only, our method achieves 9.41% / 7.07% / 4.10% performance in terms of R@1-0.1, R@1-0.3, R@1-0.5 respectively, which is a competitive result compared with state-of-the-arts. Benefit from the long-range context encoding and global-view rank learning, the Pre-ranking module explore the cross-modal semantic relationship in long videos adequately, thus facilitates accurate grounding. Upon this, integrating Re-ranking module achieves improvements of +0.76%/+0.58%/+0.33%, because the detailed frame knowledge supplement fine-grained semantics, *e.g.*, the scene and objects occurred in few frames, that generally perturbed by many unrelated frames. Integrating Boundary Regression module achieves improvements of +1.38%/+1.45%/+0.69%, which benefits from the flexible adjustments. The combination of the three modules achieves improvements of +1.62%/+1.76%/+1.13%, which demonstrates the complementary of proposed modules.

Impact of Dual-form Approximate Rank Loss. To make clear the contribution of the proposed dual-form approximate rank Loss \mathcal{L}_{dar} , we compare it with three

Table 7. Ablation study on TST number on MAD.

# TST	IoU = 0.3			IoU = 0.5			Infer Time
	R@1	R@10	R@100	R@1	R@10	R@100	
1	9.06	26.44	53.23	5.41	17.96	38.53	178.93s
2	9.11	26.62	53.37	5.36	17.98	39.05	273.08s
4	9.09	26.29	54.54	5.32	17.90	39.69	475.90s

Table 8. Ablation study on TST number on Ego4d-Video-NLQ.

# TST	IoU = 0.3		IoU = 0.5		Infer Time
	R@1	R@5	R@1	R@5	
1	2.90	9.44	1.41	4.82	34.47s
2	3.84	10.52	1.98	5.08	36.42s
4	3.61	10.67	1.72	5.02	38.32s

loss functions: (1) Binary cross entropy loss \mathcal{L}_{bce} , which uses IoU as labels to optimize the query-anchor matching scores; (2) Noise contrastive estimation loss \mathcal{L}_{nce} , which optimizes a hidden space where positive pairs are assigned close and negative pairs are pushed away. We select the anchor with highest IoU as positive samples and others as negative samples; (3) Single ApproxNDCG loss \mathcal{L}_{ar} , which optimizes the anchor rank only. The results are summarized in Tab. 6. \mathcal{L}_{bce} achieves poor performance in all metrics, mainly caused by the extremely imbalance of positive (which has IoU > 0) and negative (which has IoU = 0) samples, even though we have enlarge the weight of positive samples. Besides, \mathcal{L}_{ar} and \mathcal{L}_{dar} both surpass \mathcal{L}_{nce} by a large margin, because \mathcal{L}_{nce} only tries to distinguish the anchor with highest IoU from large amounts of anchor candidates, while \mathcal{L}_{ar} and \mathcal{L}_{dar} implement the anchor rank optimization from the global perspective, which needs to consider the relationship between each anchor pair. Finally, \mathcal{L}_{dar} outperforms \mathcal{L}_{ar} by 0.68% / 0.43% in terms of R@1-0.3 and R@1-0.5, demonstrating the complementary of query rank optimization and anchor rank optimization.

Multi-scale mechanism. We conduct experiments on MAD and Ego4d-Video-NLQ to study the impact of the number of TST blocks (*i.e.*, the number of anchor scales). Results are shown in Tab. 7 and Tab. 8, respectively. ‘‘Infer Time’’ refers to the total time cost over the entire test set. Upon analyzing the tables, it is evident that incorporating a multi-scale mechanism significantly improves performance. By utilizing 2 scales, consistent enhancements are observed across all metrics of MAD and Ego4d-Video-NLQ, with the exception of R@1-0.5 on MAD. Furthermore, employing 4 scales yields a substantial performance gain on the R@100 metric on MAD. However, it should be noted that the introduction of a multi-scale mechanism does lead to a slight decrease in efficiency. Comparatively, a 4-scale network requires $2.7\times$ and $1.1\times$ more time than a single-scale network for MAD and Ego4d-Video-NLQ, respectively.

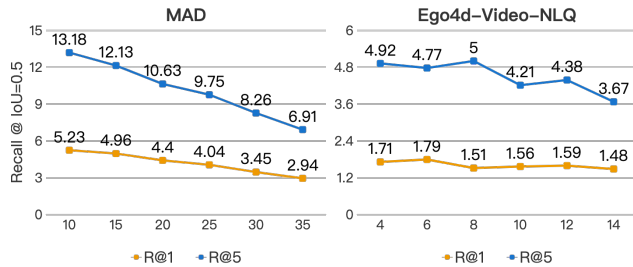


Figure 3. Ablation study on the base anchor length, C_0 .

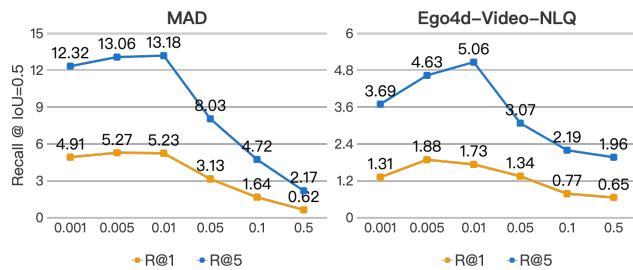


Figure 4. Ablation study on the temperature value, α .

Nevertheless, these time costs are relatively modest when compared to the computational demands of sliding window-based approaches.

Base anchor length C_0 . We vary the value of C_0 to study the impact of anchor length and summarize the results in Fig. 3. We observe that the performance decreases greatly on MAD as C_0 grows, while not changes obviously on Ego4d-Video-NLQ. This intriguing finding can be attributed to the characteristics of the ground-truth moments in each dataset. In the case of MAD, the majority of ground-truth moments have a very short duration. Consequently, it becomes challenging to accurately align a long anchor with the query and accurately regress the boundaries. On the contrary, the ground-truth moments in Ego4d-Video-NLQ exhibit a more-balanced length distribution, rendering them less sensitive to variations in anchor length.

Temperature α . We vary the value of α in \mathcal{L}_{dar} across from 0.001 to 0.5 to study the impact. The results are presented in Fig. 4. It is evident from the results that the optimization process is highly sensitive to the choice of α . We observe that the performance reaches its peak when α falls within the range of [0.005, 0.01]. However, as the value of α increases beyond this optimal range, we observe a significant decline in performance.

4.7. Qualitative Analysis

We provide qualitative results to illustrate the contributions of Pre-ranking and Re-ranking modules. Fig. 5 displays the predictions of SOONet without Re-ranking (first line) and with Re-ranking (second line), as well as the corresponding groundtruth (third line). It suggests, equipped

Query: what seasoning did I use?

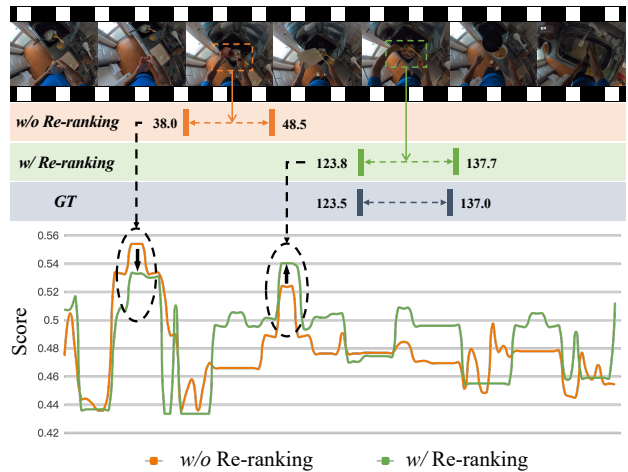


Figure 5. Qualitative analysis on the re-ranking module with full-length anchor matching scores, where re-ranking helps localize the moment of interest more precisely.

with only Pre-ranking module, our method achieves coarse localization (two humps showed on orange line) but loses some fine-grained details so that it can not distinguish food and seasoning. However, when combined with Re-ranking module, our method succeeds in recognizing the seasoning, and raises the confidence of the right moment as well as decreases the matching score of wrong moment. More qualitative results are provided in *Supplementary Material*.

5. Conclusion and Discussion

We propose an end-to-end framework, SOONet, for fast temporal grounding in long videos, which is able to model an hours-long video with one-time network execution, alleviating the inefficiency issue caused by the sliding window pipeline. Extensive experiments on MAD and Ego4d datasets demonstrate the superiority of our SOONet regarding both accuracy and efficiency.

Still, some further explorations can be expected. Due to the fact that the sentence feature is pre-extracted by a pre-trained language model, some fine-grained cross-modal fusion (e.g., word-to-object semantic alignment) may get lost. We visualize some failure cases in *Supplementary Material* and leave it as a future work.

Acknowledgement

This work was supported by the grants from the National Natural Science Foundation of China (62272013, 61925201, 62132001) and Beijing Natural Science Foundation (4232005).

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. **3**
- [2] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 162–171, 2018. **2**
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. **3, 6**
- [4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Int. Conf. Comput. Vis.*, pages 6202–6211, 2019. **6**
- [5] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Int. Conf. Comput. Vis.*, pages 5267–5275, 2017. **1, 2**
- [6] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18995–19012, 2022. **1, 2, 5, 6, 7**
- [7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, pages 1735–1780, 1997. **3**
- [8] Zhijian Hou, Wanjun Zhong, Lei Ji, Difei Gao, Kun Yan, Wing-Kwong Chan, Chong-Wah Ngo, Zheng Shou, and Nan Duan. Cone: An efficient coarse-to-fine alignment framework for long video temporal grounding. *arXiv preprint arXiv:2209.10918*, 2022. **1, 2, 6**
- [9] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. Location-aware graph convolutional networks for video question answering. In *Assoc. Adv. Artif. Intell.*, pages 11021–11028, 2020. **1**
- [10] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Int. Conf. Comput. Vis.*, pages 706–715, 2017. **5**
- [11] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Eur. Conf. Comput. Vis.*, pages 447–463, 2020. **1**
- [12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. **2, 3**
- [13] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10810–10819, 2020. **1, 2**
- [14] Tao Qin, Tie-Yan Liu, and Hang Li. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval*, pages 375–397, 2010. **5**
- [15] Xiaoye Qu, Pengwei Tang, Zhikang Zou, Yu Cheng, Jianfeng Dong, Pan Zhou, and Zichuan Xu. Fine-grained iterative attention network for temporal language localization in videos. In *ACM Int. Conf. Multimedia*, pages 4280–4288, 2020. **2**
- [16] 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 *Int. Conf. Mach. Learn.*, pages 8748–8763, 2021. **3, 6, 7**
- [17] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Eur. Conf. Comput. Vis.*, pages 510–526, 2016. **5**
- [18] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5026–5035, 2022. **1, 2, 5, 6**
- [19] Mattia Soldan, Mengmeng Xu, Sisi Qu, Jesper Tegner, and Bernard Ghanem. Vlg-net: Video-language graph matching network for video grounding. In *Int. Conf. Comput. Vis.*, pages 3224–3234, 2021. **1, 2, 6, 7**
- [20] Hao Wang, Zheng-Jun Zha, Liang Li, Dong Liu, and Jiebo Luo. Structured multi-level interaction network for video moment localization via language query. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7026–7035, 2021. **2**
- [21] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *Assoc. Adv. Artif. Intell.*, pages 12168–12175, 2020. **2**
- [22] Peng Wu, Xiangteng He, Mingqian Tang, Yiliang Lv, and Jing Liu. Hanet: Hierarchical alignment networks for video-text retrieval. In *ACM Int. Conf. Multimedia*, pages 3518–3527, 2021. **1**
- [23] Bo Xiong, Yannis Kalantidis, Deepti Ghadiyaram, and Kristen Grauman. Less is more: Learning highlight detection from video duration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1258–1267, 2019. **1**
- [24] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 982–990, 2016. **1**
- [25] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *ACM Int. Conf. Multimedia*, pages 516–520, 2016. **5**
- [26] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Eur. Conf. Comput. Vis.*, pages 471–487, 2018. **1**
- [27] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *Adv. Neural Inform. Process. Syst.*, 2019. **1, 2**
- [28] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Assoc. Adv. Artif. Intell.*, pages 9159–9166, 2019. **2**
- [29] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10287–10296, 2020. **1, 2**
- [30] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1247–1257, 2019. 1, 2
- [31] 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. 1, 2, 6, 7
- [32] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Assoc. Adv. Artif. Intell.*, pages 12870–12877, 2020. 1, 2, 6, 7
- [33] Zongmeng Zhang, Xianjing Han, Xuemeng Song, Yan Yan, and Liqiang Nie. Multi-modal interaction graph convolutional network for temporal language localization in videos. *IEEE Trans. Image Process.*, pages 8265–8277, 2021. 2
- [34] 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. 2
- [35] Qi Zheng, Jianfeng Dong, Xiaoye Qu, Xun Yang, Shouling Ji, and Xun Wang. Progressive localization networks for language-based moment localization. *arXiv preprint arXiv:2102.01282*, 2021. 2