SnAG: <u>S</u>calable a<u>n</u>d <u>A</u>ccurate Video <u>G</u>rounding

Fangzhou Mu[*], Sicheng Mo[*], Yin Li

University of Wisconsin-Madison

Abstract

Temporal grounding of text descriptions in videos is a central problem in vision-language learning and video understanding. Existing methods often prioritize accuracy over scalability — they have been optimized for grounding only a few text queries within short videos, and fail to scale up to long videos with hundreds of queries. In this paper, we study the effect of cross-modal fusion on the scalability of video grounding models. Our analysis establishes late fusion as a more cost-effective fusion scheme for long-form videos with many text queries. Moreover, it leads us to a novel, video-centric sampling scheme for efficient training. Based on these findings, we present SnAG, a simple baseline for scalable and accurate video grounding. Without bells and whistles, SnAG is 43% more accurate and 1.5× faster than CONE, a state of the art for long-form video grounding on the challenging MAD dataset, while achieving highly competitive results on short videos. Our code is available at https://github.com/fmu2/snag_release.*

1. Introduction

Localizing moments within an untrimmed video based on text descriptions, also known as temporal video grounding, involves joint reasoning about visual, textual and temporal information, and thus remains a challenging problem in video understanding. Video grounding is gaining traction amid the rapid advances in vision-language learning, with a range of applications such as intelligent personal assistants, human-robot interaction, and interactive video editing.

Despite recent progress, one largely overlooked aspect of video grounding is the *scalability* of models with respect to video duration and the number of text descriptions (*i.e.*, queries). Previous methods (*e.g.*, [46, 67]) are primarily designed for grounding only a few queries within short video snippets. However, the growing availability of long videos (*e.g.*, on streaming platforms) and demand to query their rich contents introduce a paradigm shift, necessitating efficient grounding of large volumes of queries in long videos.
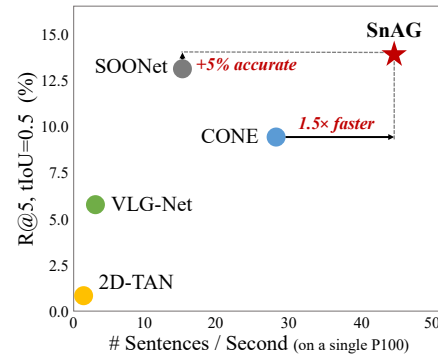
* indicates equal contribution

Figure 1. SnAG achieves the best accuracy and throughput simultaneously on the MAD dataset [45] for long-form video grounding.

On latest long-form video grounding benchmarks [16, 45], these methods struggle to scale up due to their high computational cost, and suffer major performance degradation.

In this paper, we take a principled approach to studying the problem of scalable video grounding, focusing on the design of cross-modal fusion—a key model component that combines visual and textual information. Specifically, we present an analysis that connects the computational cost of video grounding models to their fusion schemes. Our analysis and empirical results challenge the widely adopted early fusion, showing that late fusion scales better in inference by amortizing the cost of long video processing among text queries. Importantly, this insight further motivates the design of a novel, video-centric sampling scheme for scalable training, without compromising grounding accuracy.

Following these findings, we present SnAG, a scalable and accurate model for long-form video grounding. SnAG features a minimalist, late-fusion design for scalable inference, while supporting video-centric sampling for scalable training. Notwithstanding its simplicity, SnAG achieves superior results across all benchmarks designed for long-form video grounding. Without bells and whistles, SnAG attains 13.75% R@5, tIoU=0.5 on MAD [45] with an inference speed of 45.3 queries per second on a P100 GPU (Fig. 1), 43% more accurate and 1.5× faster than the concurrent method of CONE [17]. In the meantime, SnAG is highly competitive on short-video benchmarks. It achieves 46.26% R@1, tIoU=0.7 on Charades-STA [43], outperforming the

previous state of the art (SSRN [71]) by 3.61 absolute percentage points while being simpler in model design.

Contributions. **(1)** We present one of the first studies to understand the effect of fusion schemes on the computational cost of video grounding. **(2)** Our analysis offers new insights and leads to a new training scheme for scalable video grounding. **(3)** Based on our findings, we further deliver a simple model that attains superior results across major benchmarks for long-form video grounding while achieving significant efficiency gains in both training and inference.

2. Related Work

Video grounding. Previous works on video grounding can be categorized based on whether proposals are generated as moment candidates (two-stage *vs*. single-stage), and how text queries inform action localization (cross-modal fusion).

Two-stage methods first generate temporal segments as proposals, then score their "actionness" and optionally refine their boundaries. Early works [1, 13] densely sample proposals using sliding windows and score them independently. One line of work [8, 28, 54–56] subsequently conditions proposal generation on sentence queries and/or video context to avoid dense sampling. By contrast, another line of work [14, 46, 50, 66, 67] takes the opposite path of enumerating all segments, yet organizes them in a 2D adjacency map to reason about their relations.

Single-stage methods localize moments in a single shot without using proposals, and thus are often more efficient than two-stage methods. Some models decode moment boundaries from global features [23, 24, 33, 59, 70] or learnable queries [21, 25, 53]. Others densely predict onset and offset probabilities [15, 29, 34, 41, 63–65, 68], or classify and refine pre-set anchors [6, 30, 51, 58, 62] at every moment. Most relevant to our work are models that represent moment candidates as points [7, 11, 27, 31, 60]. Analogous to point-based object detectors [47], they classify every point in time and regress moment boundaries relative to the points. Despite their simplicity and efficiency, the performance of single-stage models lags behind two-stage models on standard benchmarks made up of short videos.

Cross-modal fusion combines video and text information for video grounding. Early models [1, 13] fuse video and text representations right before final prediction. This simple, late-fusion approach is abandoned by subsequent models, which turn to early fusion (*e.g*., [56, 60, 62, 67]) with sophisticated model design (*e.g*., LSTM [6, 15, 29, 51, 56, 65], GCN [7, 27, 46], memory bank [8, 27]), feature hierarchies [14, 33, 50], losses [23, 34, 56, 66], and inference procedures [68, 70]. Early fusion is believed to strengthen cross-modal reasoning, albeit at a price of model complexity. Recent Transformer-based models [3, 21, 53, 66] inherit this early-fusion design; video features and text embeddings

are concatenated at input, and attention-based feature fusion is carried out throughout the model.

Our method demonstrates a single-stage Transformer model with a simple late fusion mechanism based on cross-attention. Late fusion allows us to amortize the cost of video processing across many sentence queries, resulting in scalable training and inference on long-form videos. Without bells and whistles, our model outperforms all prior methods on long-video benchmarks, and attains competitive performance on standard, short-video benchmarks.

Long-form video grounding. Recent efforts [16, 45] put forth new challenges for sentence grounding in long videos. Previous methods tailored for short videos [46, 64, 67] fall short on these benchmarks. They have no access to long-range video context, require dense sliding window inference, and yield unsatisfactory results. Concurrent works [17, 36] present multi-stage approaches with sophisticated pre-processing, ranking and training procedures for efficient long-form video grounding. However, their efficiency stems from the restrictive assumption that moments are covered by short windows. By contrast, SnAG offers a simple, single-stage solution with superior accuracy, without making assumptions about moment duration.

Localizing moments in videos. Closely related to video grounding, temporal action localization (*e.g*., [32, 42, 44]) aims to identify moments of actions from within a closed set of action categories. Solutions to the two tasks often evolve in tandem and share remarkably similar model design. SnAG is inspired by our prior work [61], yet extends its design for the more challenging task of video grounding, with the goal of scaling up video grounding for long videos.

3. Scaling Up Video Grounding

Our goal is to scale up video grounding for long-form videos with many queries. Denote an input video $\mathbf{V}$ as a time-indexed sequence $[\mathbf{v}_t]_{t=1}^T = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_T]$, where $\mathbf{v}_t$ represents a short video clip centered at time t.[1] $\mathbf{V}$ is paired with N sentence queries $\{Q_j\}_{j=1}^N$, each describing a moment $\mathbf{y}_j = (s_j, e_j)$ in the video.

Long-form video grounding is characterized by a great number of time steps T (*e.g*., tens of thousands) and text queries N (hundreds) for an input video $\mathbf{V}$. With a large T, it is no longer feasible to feed the full video $\mathbf{V}$ into a model, and thus more viable to consider a sliding window setting. Specifically, a video $\mathbf{V}$ is further partitioned into a set of M snippets $\{\mathbf{V}_{i=1}^M\}$, each $\mathbf{V}_i = [\mathbf{v}_t]_{t=t_i}^{t_i+T_w}$ taken from a local time window of size T_w that may overlap with others. We consider the learning of a function $\mathcal{F}$, realized in a deep

---

[1]Like prior works, we consider video features and text embeddings extracted using pre-trained models (*e.g*., C3D [48] and BERT [10]). Yet the formulation can be easily extended to raw video frames and text tokens.
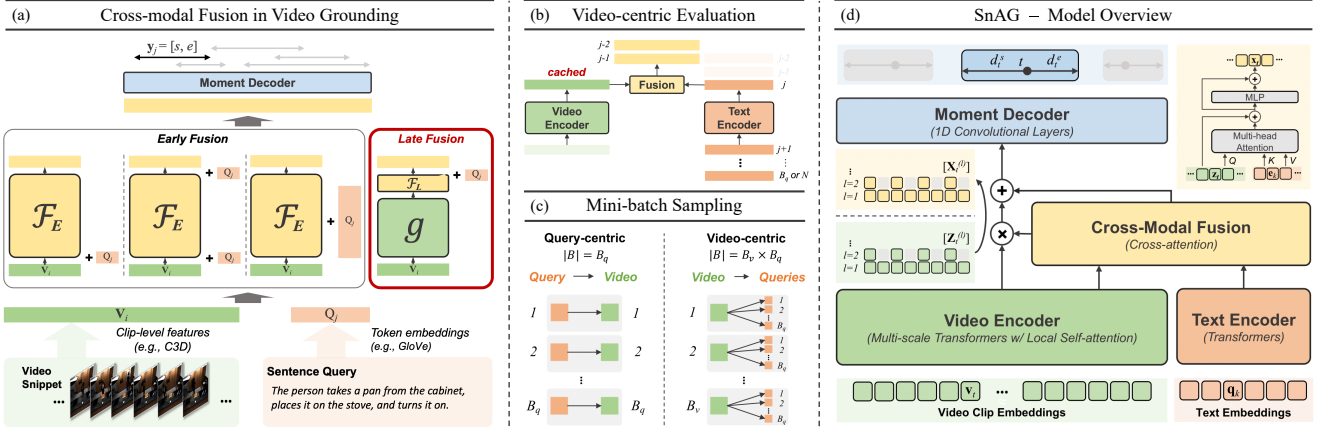
Figure 2. **(a) Cross-modal fusion** is key to a video grounding model $\mathcal{F}$. Models using *early fusion* jointly encode video and sentence query. SnAG revisits *late fusion* for scalable video grounding by decoupling expensive video encoding and inexpensive query encoding. **(b) Video-centric model evaluation.** With late fusion, the output of video encoder can be cached and re-used by queries of the same video in both training and inference. **(c) Mini-batch sampling in training**. Previous methods adopt *query-centric* sampling (query → video) whereas SnAG resorts to *video-centric* sampling (video → many queries) for efficient training. **(d) Model overview**. SnAG is a simple instantiation of late fusion and video-centric training for video grounding. It separately encodes a video and its queries using Transformers, applies simple cross-attention for cross-modal fusion, and decodes moments represented as points using lightweight convolutional heads.

network, to predict the start time s_j and end time e_j of the moment $\mathbf{y}_j$ given a snippet $\mathbf{V}_i$ and a query Q_j.

3.1. Background: Cross-Modal Fusion

We review cross-modal fusion — a key design choice of $\mathcal{F}$, and define the notation used for the rest of the paper.

Early fusion. A straightforward design is to fuse information from a video snippet $\mathbf{V}_i$ and a text query Q_j at input (Fig. 2(a)). This early fusion strategy is widely adopted in prior works [21, 53, 60, 62, 66, 67], and can be written as

$$\mathcal{F}_E(\mathbf{V}_i, Q_j) \mapsto \mathbf{y}_j, \tag{1}$$

where $\mapsto$ defines any necessary post-processing of the model outputs (*e.g.*, re-scaling, non-maximum suppression) in order to decode the moment $\mathbf{y}_j$.

Late fusion. Another possible design is to first embed the video snippet $\mathbf{V}_i$ and text query Q_j, then fuse their features for grounding (Fig. 2(a)). This late fusion strategy follows

$$\mathcal{F}_L(g(\mathbf{V}_i), h(Q_j)) \mapsto \mathbf{y}_j, \tag{2}$$

where $g(\cdot)$ is the video encoder and $h(\cdot)$ the text encoder. With slight abuse of the symbols, $\mathcal{F}_L$ here denotes the rest part of the grounding model, including feature fusion. The key is to keep $\mathcal{F}_L$ lightweight, *i.e.*, its cost should be a minor portion in comparison to the rest of the model.

Design choice. The conventional wisdom favors the design of early fusion for video grounding. Early fusion is believed to offer better interactions between visual and textual features, resulting in strong empirical results. In what follows, we analyze the computational cost of early and late fusion

during inference and training, and show that proper design of late fusion allows significantly improved scalability in video duration (M) and number of text queries (N).

3.2. Scalable Inference

We start by analyzing the computational cost of early *vs.* late fusion in inference. Denote the computational cost of a function by $\mathcal{C}(\cdot)$. For example, $\mathcal{C}(g)$ is the cost of the video encoder $g(\cdot)$, and $\mathcal{C}(h)$ the cost of the text encoder $h(\cdot)$.

Computational cost. We make two key observations regarding the computational cost of a video grounding model: (a) a lightweight $\mathcal{F}_L$ is used for late fusion, thus $\mathcal{C}(\mathcal{F}_L) \ll \mathcal{C}(g) + \mathcal{C}(h)$; and (b) the video encoder $g(\cdot)$ has a much higher cost than the text encoder, thus $\mathcal{C}(g) \gg \mathcal{C}(h)$.

The first observation stems from the design of late fusion. To justify the second observation, it is important to notice that $\mathcal{C}(g)$ scales w.r.t. the length of input video snippet (*i.e.*, the local window size T_w), whereas $\mathcal{C}(h)$ remains constant due to limited sequence length of text queries. For example, a realization of $g(\cdot)$ using a vanilla Transformer has a complexity of $\mathcal{O}(T_w^2)$. Thus, using a larger local window size naturally leads to $\mathcal{C}(g) \gg \mathcal{C}(h)$.

Cost analysis: early *vs*. late fusion. Evaluating an early-fusion model on a video with M snippets and N queries has a cost of $MN\,\mathcal{C}(\mathcal{F}_E)$, whereas late fusion has a cost of $MN\,\mathcal{C}(\mathcal{F}_L) + M\mathcal{C}(g) + N\mathcal{C}(h)$. This is because an early-fusion model has to compare each snippet to every query, yet late fusion allows re-using video features for all queries of the video (*i.e.*, *video-centric* evaluation, see Fig. 2(b)).

Let us assume that the cost of both models can be controlled by setting $\mathcal{C}(\mathcal{F}_E) = \mathcal{C}(\mathcal{F}_L) + \mathcal{C}(g) + \mathcal{C}(h)$, *i.e.*, eval-

uating either model on a pair of video snippet and sentence query has the same cost (post-processing excluded). With $\mathcal{C}(\mathcal{F}_L) = \alpha(\mathcal{C}(g) + \mathcal{C}(h))$ for some small α, the ratio between the cost of early and late fusion in inference is

$$\mathbf{R}_{\text{inf}} = \frac{\mathcal{C}(\text{inference with early fusion})}{\mathcal{C}(\text{inference with late fusion})} \tag{3}$$

$$= \frac{MN\,\mathcal{C}(\mathcal{F}_L) + MN\,\mathcal{C}(g) + MN\,\mathcal{C}(h)}{MN\,\mathcal{C}(\mathcal{F}_L) + M\,\mathcal{C}(g) + N\,\mathcal{C}(h)} \tag{4}$$

$$= \frac{MN(1+\alpha)}{MN\alpha + M\mathbf{R}_{g/(g+h)} + N(1 - \mathbf{R}_{g/(g+h)})} \tag{5}$$

$$\approx \frac{1+\alpha}{\frac{1}{N}+\alpha} \to \begin{cases} N, & \alpha \to 0 \\ 1 + \frac{1}{\alpha}, & N \to \infty \end{cases}, \tag{6}$$

where $\mathbf{R}_{g/(g+h)} = \mathcal{C}(g)/(\mathcal{C}(g)+\mathcal{C}(h))$, and the approximation in Eq. 6 is a result of $\mathbf{R}_{g/(g+h)} \approx 1$ with $\mathcal{C}(g) \gg \mathcal{C}(h)$. Eq. 6 reveals that early fusion is N times the cost of late fusion as snippet length grows ($\alpha \to 0$), and is particularly inefficient for videos with many queries ($N \to \infty$).

3.3. Scalable Training

We now show that late fusion further unlocks a novel training scheme that scales well for long-form video grounding.

Training and mini-batch sampling. Given a set of paired videos, text queries, and their moments $\{\mathbf{V}^{(k)}, \mathbf{Q}^{(k)}, \mathbf{y}^{(k)}\}$, training a model $\mathcal{F}$ amounts to minimizing a loss function $\mathcal{L}$ over mini-batches B sampled from the dataset:

$$\Sigma_{(k,i,j)\in B}\ \mathcal{L}\left(\mathcal{F}\left(\mathbf{V}_i^{(k)}, \mathbf{Q}_j^{(k)}\right), \mathbf{y}_j^{(k)}\right). \tag{7}$$

A key design largely overlooked in video grounding is the mini-batch sampling scheme during training. With multiple queries associated with a video snippet, a combination of late fusion and proper sampling scheme allows us to re-use video representations in training as is done in inference.

Query-centric sampling. Virtually all prior works sample mini-batches uniformly at random from all pairs of valid video snippets and text queries.[2] This can be interpreted as ancestral sampling: first draw a text query (and its moment), then select a video snippet covering the moment:

$$Q_{j'}^{(k')}, \mathbf{y}_{j'}^{(k')} \sim U\left(\{(Q_j^{(k)}, \mathbf{y}_j^{(k)})\}_{j\times k}\right),$$
$$\mathbf{V}_{i'}^{(k')} \sim U\left(\{(\mathbf{V}_i^{(k')}, Q_{j'}^{(k')}, \mathbf{y}_{j'}^{(k')})_+\}\right), \tag{8}$$

where $U(\cdot)$ denotes a discrete uniform distribution and $\{(\mathbf{V}_i^{(k')}, Q_{j'}^{(k')}, \mathbf{y}_{j'}^{(k')})_+\}$ is the union of all valid triplets: a moment $\mathbf{y}_{j'}^{(k')}$ of a text query $Q_{j'}^{(k')}$ in a video snippet $\mathbf{V}_{i'}^{(k')}$.

This sampling process is repeated $|B|$ times to construct a mini-batch, with $|B|$ the batch size. With a large

---

[2]A video snippet without any query is often discarded during training.

---

dataset, the resulting mini-batch will contain $|B|$ snippet-query pairs, each likely from a different video.

Video-centric sampling. To speed up the training of late fusion models, we propose video-centric sampling that considers every video snippet $\mathbf{V}_i^{(k)}$ as a sample. Specifically, we first draw a video based on its importance score, then sample a snippet within the video, and finally select *multiple* text queries associated with the snippet. This is defined as the following ancestral sampling:

$$k' \sim P(k) = \frac{M_+^{(k)}}{\Sigma_k M_+^{(k)}} \quad i' \sim P(i;k') = \frac{N_i^{(k')}}{\Sigma_k N_i^{(k')}} \tag{9}$$
$$\{Q_{j'}^{(k')}, \mathbf{y}_{j'}^{(k')}\}^{B_q} \sim U\left(\{(\mathbf{V}_{i'}^{(k')}, Q_j^{(k')}, \mathbf{y}_j^{(k')})_+\}\right),$$

where $M_+^{(k)}$ denotes the number of valid snippets in a video $\mathbf{V}^{(k)}$, and $N_i^{(k')}$ the number of queries in a snippet $\mathbf{V}_i^{(k')}$. Importantly, a total of B_q queries are drawn for the selected video snippet $\mathbf{V}_{i'}^{(k')}$.

This sampling process is repeated $|B|/B_q$ times for a mini-batch, leading to $|B|/B_q$ video snippets, each likely from a different video yet with multiple (B_q) text queries. This mini-batch thus allows us to re-use the video representations for the sampled snippets.

Cost analysis: query-centric *vs.* video-centric sampling. Fig. 2(c) illustrates the key difference between query-centric and video-centric sampling. Similar to our analysis in Section 3.2, it is easy to show that late fusion combined with video-centric training enables substantial speedup for long snippets ($\alpha \to 0$) and large B_q:

$$\mathbf{R}_{\text{train}} = \frac{\mathcal{C}(\text{late fusion + video-centric training})}{\mathcal{C}(\text{late fusion + query-centric training})} \tag{10}$$

$$\approx \frac{1+\alpha}{\frac{1}{B_q}+\alpha} \to \begin{cases} B_q, & \alpha \to 0 \\ 1 + \frac{1}{\alpha}, & B_q \to \infty \end{cases}. \tag{11}$$

3.4. Scalable Video Grounding: An Instantiation

We now present an instantiation with late fusion and video-centric training for scalable video grounding. Our instantiation, dubbed SnAG, is shown in Fig. 2(d). SnAG is a single-stage Transformer model where every time step represents a moment candidate. It combines (a) a multi-scale Transformer-based video encoder; (b) a Transformer-based text encoder; (c) cross-attentions for late fusion; and (d) convolutional heads for moment decoding.

Video encoder g. We use the ActionFormer backbone [61] as our video encoder, given its superior performance on the closely related task of temporal action localization. It transforms input video features into a multi-scale representation $[\mathbf{Z}^{(l)}]_{l=1}^L = [\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, ..., \mathbf{Z}^{(L)}]$ using Transformer layers with local self-attentions [9]. We refer readers to the original paper [61] for more details on model architecture.

Text encoder h**.** Our text encoder is a vanilla Transformer. It takes a query Q with K tokens $[q_1, ..., q_K]$ and outputs their embeddings $\mathbf{E} = [\mathbf{e}_1, ...\mathbf{e}_K]$. This computation is light compared to video encoding as $K \ll T_w$ for long videos.

Cross-modal late fusion $\mathcal{F}_L$**.** Our model uses a simple variant of cross-attention for fusion. Specifically, it modulates $[\mathbf{Z}^{(l)}]_{l=1}^L$ with an affine transformation given by $\mathbf{E}$:

$$
\begin{aligned}
(\mathbf{W}, \mathbf{B})^{(l)} &= \mathrm{MCA}(\mathrm{LN}(\mathbf{Z}^{(l)}), \mathrm{LN}(\mathbf{E})), \\
\tilde{\mathbf{Z}}^{(l)} &= \mathbf{W}^{(l)} \odot \mathbf{Z}^{(l)} + \mathbf{B}^{(l)}, \\
\mathbf{X}^{(l)} &= \tilde{\mathbf{c}} \odot \mathrm{MLP}(\mathrm{LN}(\tilde{\mathbf{Z}}^{(l)})) + \tilde{\mathbf{Z}}^{(l)},
\end{aligned} \quad (12)
$$

where LN is Layer Normalization [2], MLP is Multi-Layer Perceptron, and MCA is Multi-head Cross-Attention [49], whose output is split into affine weights $\mathbf{W}^{(l)}$ and biases $\mathbf{B}^{(l)}$. $\tilde{\mathbf{c}}$ are learnable per-channel scales, and $\mathbf{X}^{(l)}$ the modulated video representations prior to moment decoding.

Moment decoding. After fusion, our model decodes each time step into a moment candidate, similar to previous video grounding models [7, 27, 31, 60]. Briefly, a classification head predicts a score $p_t^{(l)}$ for all $\mathbf{x}_t^{(l)}$ within $[\mathbf{X}^{(l)}]_{l=1}^L$, and a regression head predicts the normalized distances from t to the moment boundaries (d_t^s, d_t^e) if $\mathbf{x}_t^{(l)}$ is classified positive. Both heads are lightweight, each comprising three 1D convolutions. The decoded moment $\hat{\mathbf{y}} = (\hat{s}, \hat{e})$ is given by

$$
\begin{aligned}
(t, l) &= \arg\max_{t,l} p_t^{(l)}, \\
\hat{s} = 2^{l-1} \cdot (t - d_t^s), \quad &\hat{e} = 2^{l-1} \cdot (t + d_t^e).
\end{aligned} \quad (13)
$$

At inference time, we further apply Soft-NMS [4] to merge overlapping moment predictions.

Training and loss function. We use video-centric training as described in Section 3.3. The loss function for a pair of video snippet and text query is the sum of two terms: a Focal loss [26] $\mathcal{L}_{cls}$ for moment classification and a Distance-IoU loss [69] $\mathcal{L}_{reg}$ defined on positive time steps for distance regression. Formally,

$$
\mathcal{L} = \frac{1}{C}(\mathcal{L}_{cls} + \lambda_{reg} \mathbb{1}_t^{(l)} \mathcal{L}_{reg}), \quad (14)
$$

where $\mathbb{1}_t^{(l)}$ is an indicator function that evaluates to one on positive time steps, $C = \sum_{t,l} \mathbb{1}_t^{(l)}$ is the total number of positive samples within the snippet, and λ_{reg} balances the two loss terms. Center sampling [61] is used for selecting positive time steps around the center of a target moment.

4. Experiments

Our main experiments include extensive comparisons of SnAG to strong baselines on five challenging benchmarks for video grounding. We further provide ablations on cross-modal fusion and analyze the efficiency of SnAG in both training and inference. More results are in the supplement.
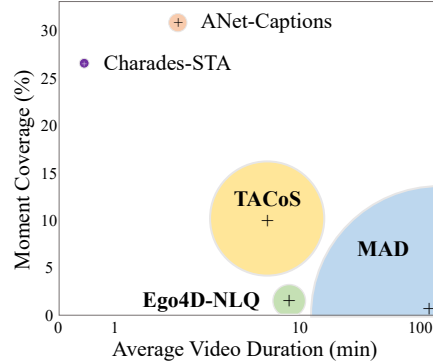

Figure 3. **Visualization of dataset statistics.** Circle radius is in proportion to average number of queries per video. Long-video benchmarks (Ego4D-NLQ, MAD and TACoS) consist of longer videos with more queries and exhibit low moment coverage compared to short-video benchmarks (ANet-Captions and Charades).

	Model	R@1			R@5		
		0.3	0.5	Avg	0.3	0.5	Avg
SF + BERT	2D-TAN [67]	5.04	2.02	3.53	12.89	5.88	9.39
	VSLNet [64]	5.45	3.12	4.29	10.74	6.63	8.69
	CONE [17]	**10.40**	5.03	7.72	22.74	11.87	17.31
	SOONet [36]	8.00	3.76	5.88	22.40	11.09	16.75
	SnAG (video-centric)	_9.83_	_6.83_	_8.33_	**27.93**	**19.27**	**23.60**
	SnAG (query-centric)	10.43	**7.15**	**8.79**	_27.59_	_19.33_	_23.46_
EgoVLP	VSLNet [64]	10.84	6.81	8.83	18.84	13.45	16.15
	CONE [17]	14.15	8.18	11.17	30.33	18.02	24.18
	SnAG (video-centric)	_15.72_	_10.78_	_13.25_	**38.39**	**27.44**	**32.92**
	SnAG (query-centric)	**15.87**	**11.26**	**13.57**	_38.26_	_27.16_	_32.71_

Table 1. **Results on Ego4D-NLQ.** SnAG outperforms all baselines. Best results are in **bold** and second best underlined.

Datasets. We categorize five benchmark datasets into two groups based on their video and query statistics (Fig. 3).

Long videos, many queries. Ego4D-NLQ [16] [3] is a large collection of egocentric videos on daily human activities. The videos are 3.5 to 20 minutes long with an average of 11.6 queries. MAD [45] contains 1.2K hours of movies with 384K queries transcribed from audio description. The videos are 47 to 202 minutes long. TACoS [40] is a conventional benchmark with 10.1 hours of cooking videos, with an average of 143.5 queries per video. These datasets expose the scalability challenges that motivate our work and thus are the main focus of our experiments.

Short videos, few queries. Charades-STA [43] is an action recognition dataset later annotated for video grounding [13]. The videos are ∼30 seconds long with ∼2.4 queries per video. ActivityNet-Captions [20] was collected for dense video captioning and later repurposed for video grounding. The videos are two minutes long on average with a small set of ∼3.65 queries. Several works [35, 57] advise against using these datasets as benchmarks for the

---

[3] All experiments are conducted using the Ego4D *v1* dataset.

Model	R@1			R@5			R@10			R@50		
	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
CLIP [39]	6.57	3.13	1.39	15.05	9.85	5.44	20.26	14.13	8.38	37.92	28.71	18.80
VLG-Net [46]	3.64	2.76	1.65	11.66	9.31	5.99	17.39	14.56	9.77	39.78	34.27	24.93
Moment-DETR [21]	0.31	0.24	0.16	1.52	1.14	0.28	2.79	2.06	1.20	11.08	7.97	4.71
CONE [17]	8.90	6.87	4.10	20.51	16.11	9.59	27.20	21.53	12.82	43.36	34.73	20.56
SOONet [36]	**11.26**	**9.00**	5.32	23.21	19.64	13.14	30.36	26.00	17.84	50.32	44.78	32.59
SnAG (video-centric)	<u>10.28</u>	<u>8.46</u>	**5.55**	**24.42**	**20.60**	**13.75**	**32.23**	**27.50**	**19.00**	**52.28**	**46.68**	**35.24**
SnAG (query-centric)	10.18	8.37	<u>5.44</u>	<u>24.30</u>	<u>20.35</u>	<u>13.51</u>	<u>32.05</u>	<u>27.25</u>	<u>18.75</u>	<u>52.05</u>	<u>46.53</u>	<u>34.83</u>

Table 2. **Results on MAD.** SnAG outperforms all baselines. Best results are in **bold** and second best <u>underlined</u>.
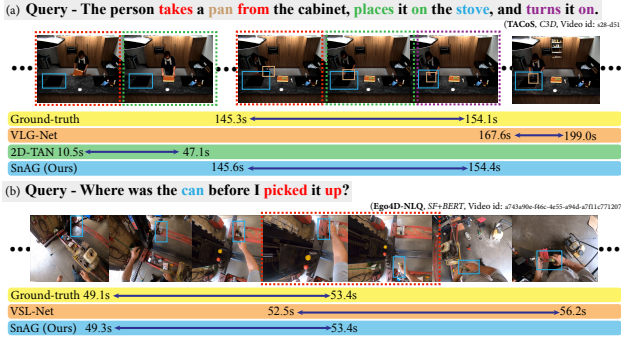

Figure 4. **Visualization of moment predictions.** SnAG can **(a)** comprehend complex text queries with multiple objects and actions; **(b)** reason about temporal ordering of events.

temporal biases in their moment annotations. Nevertheless, we report results on these datasets for completeness.

Baselines. On TACoS, Charades and ANet-Captions, we compare to strong baselines including two-stage [28, 55, 56] and single-stage [30, 50, 70] models with various fusion operations (LSTM [29, 51], GCN [27, 46], co-attention [24, 68], *etc*.). On Ego4D and MAD, we compare to their official baselines (2D-TAN [67] and VSLNet [64] for Ego4D, CLIP [39] and VLG-Net [46] for MAD) as well as the concurrent method of CONE [17] and SOONet [36] that similarly address scalable video grounding. We discuss additional baselines in the supplement.

Evaluation metric. We report Recall@k at various temporal intersection-over-union threshold θ (R@k, tIoU=θ) for all datasets. It measures the percentage of sentence queries with at least one of the top-k moment predictions whose temporal overlap with the ground-truth moment exceeds θ.

4.1. Long Videos, Many Queries

Experiment setup. For Ego4D, We consider (1) Slow-Fast [12] video features with BERT [10] text features, and (2) EgoVLP [38] video and text features. We train on the *train* split and report R@{1, 5}, tIoU={0.3, 0.5} on the *val* split. For MAD, we use CLIP [39] features for both videos and text. We train on the *train* split and report R@{1, 5, 10, 50}, tIoU={0.1, 0.3, 0.5} on the *test* split. For TACoS, we use C3D video features [48] and GloVe [37] for

sentence embeddings. We train on the *train* split and report R@{1, 5}, tIoU={0.5, 0.7} on the *test* split. We report results using both video-centric and query-centric training on all datasets.

Results on Ego4D-NLQ (Table 1). SnAG sets the new state of the art, achieving mean R@1 / R@5 scores of 8.79% / 23.60% with SlowFast/BERT features, and 13.57% / 32.92% with EgoVLP features, doubling or even tripling the results of the official baselines [64, 67]. Compared to the strong baselines of CONE [17] and SOONet [36], SnAG improves mean R@1 and R@5 respectively by more than 1.05 and 6.29 absolute percentage points without taking advantage of external features.[4] Further, SnAG demonstrates a larger performance gain on the more stringent tIoU threshold of 0.5, indicating more precise moment localization.

Results on MAD (Table 2). SnAG sets new records on the challenging MAD dataset. It improves over CLIP [39] and VLG-Net [46] on mean R@1 / R@5 at tIoU=0.5 by 3.90 / 7.76 absolute percentage points. In the meantime, it outperforms CONE [17] and SOONet [36], with larger gain for more stringent tIoU thresholds. Further, SnAG achieves stronger results as input snippet length increases, reaching the best accuracy with T_w =16384 (Supplement Fig. A). This scaling behavior highlights the strength of our scalable model design for long-form video grounding.

Results on TACoS (Table 4 (left)). SnAG achieves an R@1 / R@5 of 44.86% / 70.66% at tIoU=0.5, outperforming the strongest baseline of MATN [66] by a significant margin (+7.5% R@1, +12.1% R@5). This confirms that our simple single-stage design with late fusion is a strong alternative to multi-stage models with sophisticated fusion mechanisms. Importantly, our scalable design enables learning from full-resolution video features, whereas the baselines often aggressively reduce feature resolution to contain their training and inference cost on long-form videos. We hypothesize that full-resolution video features capture detailed event dynamics and are key to the strong performance of SnAG.

Result visualization. We visualize predictions of SnAG and the baselines on TACoS and Ego4D-NLQ videos.[5] In

---

[4]CONE [17] leverages external CLIP [39] features in several steps.
[5]MAD videos are not publicly available due to copyright issues.

Model	R@1		R@5	
	0.3	0.5	0.3	0.5
(a) Fusion stage (early *vs.* late fusion)				
No fusion	10.85	7.85	46.46	31.19
SnAG-E	53.34	44.01	79.56	69.41
SnAG-EL	54.84	44.54	**81.13**	70.11
SnAG-M	51.41	40.79	77.56	65.71
SnAG (ours)	**55.14**	**44.94**	81.10	**70.66**
(b) Fusion operation (global *vs.* token-wise)				
Cat	49.39	39.24	77.31	65.36
Add	51.91	41.81	77.98	66.51
Scale	55.01	43.96	80.63	70.11
Add + Scale	53.36	43.09	80.43	69.58
XAttn	54.84	**45.06**	79.96	70.08
XAttn-affine (ours)	**55.14**	44.94	**81.10**	**70.66**

Table 3. **Ablations results on TACoS. (a)** Late fusion yields comparable results to early fusion (E, EL, M); **(b)** Fusion with token-wise sentence embeddings (XAttn) produces better results than fusion with global sentence embeddings (Cat, Add, Scale).

Fig. 4(a), a long query describes the complex interplay between two objects (pan and stove) and three actions ("take from", "place on", and "turn on"). VLG-Net [46] recognized both objects yet none of the actions, while 2D-TAN [67] identified one action ("take from") but at the same time missed one object (pan). By contrast, SnAG followed the query semantics and produced accurate localization. The other query in Fig. 4(b) tests the understanding of temporal context. SnAG leveraged the keyword "*before*" to identify the correct moment before the action ("pick up"), whereas VSLNet [64] only attended to the object (can), thus mistakenly predicting a moment *after* the action.

4.2. Short Videos, Few Queries

Experiment setup. We extract video features using both C3D [48] (pretrained on Sports-1M [18]) and I3D [5] (pretrained on Kinetics [19]) for Charades-STA, and use the official C3D features for ActivityNet-Captions. We use GloVe word embeddings [37] to represent sentence queries for both datasets. We train on their respective *train* splits, report R@{1, 5}, tIoU={0.5, 0.7} on the *test* split of Charades-STA, and R@{1, 5}, tIoU={0.3, 0.5} on the *val_2* split of ActivityNet-Captions. Following standard practice, we resize all video features to a uniform length of 256.

Results on Charades-STA (Table 4 (middle)). SnAG outperforms all previous methods by a wide margin. It obtains 33.52% R@1 and 64.11% R@5 at tIoU=0.7 with C3D features, 2.10 and 4.52 absolute percentage points higher than the latest SSRN [71] baseline. It further attains 46.26% R@1 and 71.94% R@5 at tIoU=0.7 with I3D features, setting new state of the art on this competitive benchmark.

Results on ActivityNet-Captions (Table 4 (right)). SnAG achieves competitive results, reaching an R@1 / R@5 score of 30.56% / 63.41% at tIoU=0.7. It is only outperformed by a few latest models [22, 27, 71] while being much simpler.

4.3. Ablation Studies

We conduct extensive ablations on TACoS to study key design choices for cross-modal fusion.

Early *vs*. late fusion. Does late fusion enable better scalability at the expense of accuracy? We study three variants of SnAG with common early-fusion designs: (1) *SnAG-E* with fusion before the pyramid; (2) *SnAG-EL* with fusion before and after the pyramid; (3) *SnAG-M* with repeated fusion within the pyramid. Table 3 confirms that early and late fusion yield comparable results.

Choice of fusion operation. We study fusion operations used by strong baselines, including point-wise addition, multiplication, and concatenation of video representations with global sentence embeddings, as well as vanilla cross-attention with token-level sentence embeddings. According to Table 3, token-level embeddings produce better results, and our cross-attention variant further boosts performance.

4.4. Efficiency Analysis

Finally, we conduct an efficiency analysis of our method.

Breakdown of model capacity. A key design choice of SnAG is to carefully balance the model capacity among its video encoder, text encoder, and moment decoder including the fusion module and the localization heads. For a detailed breakdown of model capacity, we measure their respective parameter size and MACs in percentage. Fig. 5(a) indicates that SnAG's video encoder absorbs a significant portion of parameters and compute, such that per-query evaluation (*i.e.*, text encoding, fusion, and decoding) occupies ~50% of the total computation. This design is naturally in favor of video-centric training and inference, where the costly video encoding is shared among many queries.

Video-centric *vs*. query-centric training. Fig. 5(b) presents the training time and GPU memory usage of video-centric training relative to conventional query-centric training. The results show that the amount of saving correlates with snippet length and number of queries. Video-centric training is thus preferred for long videos with many queries.

B_q **in video-centric training.** We further study how B_q, the number of text query samples per video, trades off accuracy for efficiency in video-centric training. On MAD, we fix the batch size $|B| = 32$ for a mini-batch B, vary B_q between 1 and $|B|$, and report training time, GPU memory usage and test accuracy relative to query-centric training (*i.e.*, $B_q = 1$). Fig. 5(d) reveals that test accuracy is unaffected even with a large $B_q = 32$ (*i.e.*, a single video per mini-batch), while training time and GPU memory usage

Method	TACoS (C3D)				Charades-STA (I3D)				Charades-STA (C3D)				ActivityNet-Captions (C3D)			
	R@1		R@5		R@1		R@5		R@1		R@5		R@1		R@5	
	0.3	0.5	0.3	0.5	0.5	0.7	0.5	0.7	0.5	0.7	0.5	0.7	0.5	0.7	0.5	0.7
CTRL ICCV 17 [13]	18.32	13.30	36.69	25.42	-	-	-	-	23.63	8.89	58.92	29.52	-	-	-	-
QSPN AAAI 19 [56]	-	-	-	-	-	-	-	-	35.60	15.80	79.40	45.40	27.70	13.60	59.20	38.30
SCDM NeurIPS 19 [58]	27.64	23.27	40.06	33.49	54.92	34.26	76.50	60.02	-	-	-	-	36.90	20.28	66.84	42.92
2D-TAN AAAI 20 [67]	45.61	35.77	69.11	57.31	56.64	36.21	89.14	61.13	41.10	23.25	81.53	48.55	46.16	29.21	78.80	60.85
CBP AAAI 20 [51]	27.31	24.79	43.64	37.40	-	-	-	-	36.80	18.87	70.94	50.19	35.76	17.80	65.89	46.20
DRN CVPR 20 [60]	-	23.17	-	33.36	53.09	31.75	89.06	60.05	45.40	26.40	88.01	55.38	45.45	24.36	77.97	50.30
BPNet AAAI 21 [55]	25.96	20.96	-	-	50.75	31.64	-	-	38.25	20.51	-	-	42.07	24.69	-	-
CPNet AAAI 21 [24]	42.61	28.29	-	-	60.27	38.74	-	-	40.32	22.47	-	-	40.56	21.63	-	-
SMIN CVPR 21 [50]	48.01	35.24	65.18	53.36	64.06	40.75	89.49	68.09	-	-	-	-	48.46	30.34	81.16	62.11
CBLN CVPR 21 [29]	38.98	27.65	73.12	46.24	61.13	38.22	90.33	61.69	47.94	28.22	88.20	57.47	48.12	27.60	79.32	63.41
CPN CVPR 21 [68]	47.69	36.33	-	-	51.07	31.54	-	-	-	-	-	-	45.10	28.10	-	-
DeNet CVPR 21 [70]	-	-	-	-	59.70	38.52	91.24	66.83	-	-	-	-	43.79	-	74.13	-
MATN CVPR 21 [66]	48.79	37.57	67.63	57.91	-	-	-	-	-	-	-	-	48.02	31.78	78.02	63.18
VLG-Net ICCVW 21 [46]	45.46	34.19	70.38	56.56	-	-	-	-	-	-	-	-	46.32	29.82	77.15	63.33
APGN EMNLP 21 [28]	40.47	27.86	59.98	47.12	62.58	38.86	91.24	62.11	48.20	29.37	89.05	58.49	48.92	28.64	78.87	63.19
IA-Net EMNLP 21 [30]	37.91	26.27	57.62	46.39	61.29	37.91	89.78	62.04	-	-	-	-	48.57	27.95	78.99	63.12
RaNet EMNLP 21 [14]	43.34	33.54	67.33	55.09	60.40	39.65	89.57	64.54	-	-	-	-	45.59	28.67	75.93	62.97
MGSL-Net AAAI 22 [30]	42.54	32.27	63.39	50.13	63.98	41.03	93.21	63.85	-	-	-	-	51.87	31.42	82.60	66.71
MMN AAAI 22 [52]	39.24	26.17	62.03	47.39	-	-	-	-	-	-	-	-	48.59	29.26	79.50	64.76
SSRN EMNLP 22 [71]	45.10	34.33	65.26	51.85	65.59	42.65	94.76	65.48	50.39	31.42	90.68	59.94	54.49	33.15	84.72	68.48
G2L ICCV 23 [22]	42.74	30.95	65.83	49.86	-	-	-	-	-	-	-	-	51.68	33.35	81.32	67.60
SnAG (video-centric)	**56.44**	**44.86**	**81.15**	**70.66**	64.62	**46.26**	92.55	**71.94**	**51.72**	**33.52**	**92.55**	**64.11**	48.55	30.56	81.71	63.41
SnAG (query-centric)	55.01	44.51	80.68	70.13	65.13	45.91	92.80	71.75	51.10	31.77	90.22	62.15	47.55	29.65	81.55	62.21

Table 4. **Results on TACoS, Charades-STA and ActivityNet-Captions.** SnAG outperforms all baselines on TACoS and Charade-STA by a large margin, while being highly competitive on ActivityNet-Captions. Best results are in **bold** and second best underlined.
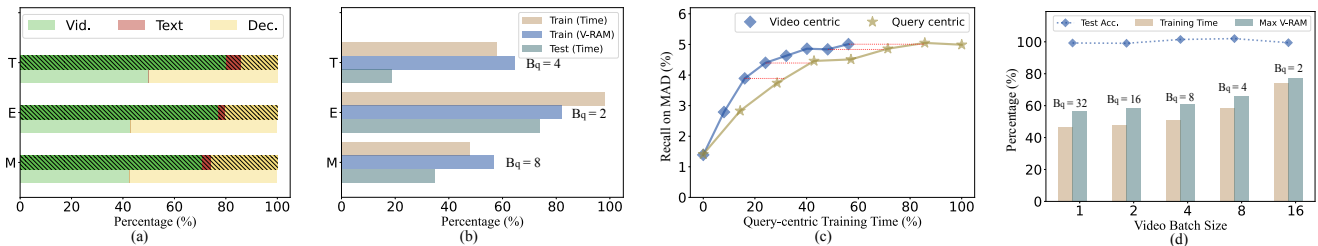

Figure 5. **(a) Model capacity.** T, E, M denotes TACoS, Ego4D and MAD, respectively (also in (b)). Shaded bar: % of parameters, Colored bar: % of MACs. SnAG places more parameters and compute in video encoder for less expensive per-query evaluation (fusion + decoding). **(b, c) Video-centric** *vs.* **query-centric inference (b) and training (c).** SnAG saves up to 50% training time, 40% GPU memory and 80% test time, and delivers up to 40% faster convergence relative to query-centric training / inference. **(d) Effect of** B_q **on training efficiency and test accuracy on MAD.** Training on MAD is faster and takes less GPU memory as B_q grows, while the test accuracy is unaffected.

drop by a significant 55% and 42%, approaching the upper bound by our analysis.

Video-centric *vs.* **query-centric inference.** We now ask to what extent the re-use of video representations enabled by SnAG's late fusion design accelerates inference. Fig. 5(b) presents the percentage of time spent by our video-centric inference scheme relative to query-centric inference, where the full model is evaluated on each and every video-query pair. Consistent with earlier findings on video-centric training, longer videos with more queries result in more saving in inference time. On the challenging MAD dataset, this alone brings a major reduction (65%) in test time.

Inference speed. With window filtering [17], inference of SnAG on the *val* split of MAD takes 12 minutes on a single P100 GPU. This is 33% faster than CONE [17] (∼18 mins) and 63% faster than SOONet [36] (∼32 mins).[6]

---

[6]SOONet reports an inference time of 8.2 mins on a single A100 GPU.

5. Conclusion

In this paper, we presented a simple and accurate model towards scaling up video grounding. We provided extensive analysis on the complexity of video grounding models from the perspective of cross-modal fusion, and argued that late fusion is key to scalability in video duration and query size. Our model, SnAG, is a simple instantiation of late fusion with minimalist design. It achieves superior accuracy and efficiency on a spectrum of challenging benchmarks and compares favorably to strong baselines including those dedicated to efficient inference on long videos. We hope our analysis and model design can shed light on video grounding and more broadly scalable video understanding.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[3] Wayner Barrios, Mattia Soldan, Alberto Mario Ceballos-Arroyo, Fabian Caba Heilbron, and Bernard Ghanem. Localizing moments in long video via multimodal guidance. In *ICCV*, 2023. 2

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *ICCV*, 2017. 5

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 7

[6] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *EMNLP*, 2018. 2

[7] 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 *AAAI*, 2020. 2, 5

[8] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In *AAAI*, 2019. 2

[9] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *ICLR*, 2021. 4

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 2, 6

[11] Xiang Fang, Daizong Liu, Pan Zhou, and Guoshun Nan. You can ground earlier than see: An effective and efficient pipeline for temporal sentence grounding in compressed videos. In *CVPR*, 2023. 2

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019. 6

[13] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *ICCV*, 2017. 2, 5, 8

[14] Jialin Gao, Xin Sun, Mengmeng Xu, Xi Zhou, and Bernard Ghanem. Relation-aware video reading comprehension for temporal language grounding. In *EMNLP*, 2021. 2, 8

[15] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. ExCL: Extractive clip localization using natural language descriptions. In *ACL*, 2019. 2

[16] 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 *CVPR*, 2022. 1, 2, 5

[17] 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. In *ACL*, 2023. 1, 2, 5, 6, 8

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 7

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 7

[20] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 5

[21] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *NeurIPS*, 2021. 2, 3, 6

[22] Hongxiang Li, Meng Cao, Xuxin Cheng, Yaowei Li, Zhihong Zhu, and Yuexian Zou. G2L: Semantically aligned and uniform video grounding via geodesic and game theory. In *ICCV*, 2023. 7, 8

[23] Juncheng Li, Junlin Xie, Long Qian, Linchao Zhu, Siliang Tang, Fei Wu, Yi Yang, Yueting Zhuang, and Xin Eric Wang. Compositional temporal grounding with structured variational cross-graph correspondence learning. In *CVPR*, 2022. 2

[24] Kun Li, Dan Guo, and Meng Wang. Proposal-free video grounding with contextual pyramid network. In *AAAI*, 2021. 2, 6, 8

[25] 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 *ICCV*, 2023. 2

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[27] Daizong Liu, Xiaoye Qu, Xing Di, Yu Cheng, Zichuan Xu, and Pan Zhou. Memory-guided semantic learning network for temporal sentence grounding. In *AAAI*, 2022. 2, 5, 6, 7

[28] Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. Adaptive proposal generation network for temporal sentence localization in videos. In *EMNLP*, 2021. 2, 6, 8

[29] 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 *CVPR*, 2021. 2, 6, 8

[30] Daizong Liu, Xiaoye Qu, and Pan Zhou. Progressively guide to attend: An iterative alignment framework for temporal sentence grounding. In *EMNLP*, 2021. 2, 6, 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 *EMNLP*, 2019. 2, 5

[32] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, 2016. 2

[33] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *CVPR*, 2020. 2

[34] Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, Hao Zhang, and Wei Lu. Interventional video grounding with dual contrastive learning. In *CVPR*, 2021. 2

[35] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkilä. Uncovering hidden challenges in query-based video moment retrieval. In *BMVC*, 2020. 5

[36] Yulin Pan, Xiangteng He, Biao Gong, Yiliang Lv, Yujun Shen, Yuxin Peng, and Deli Zhao. Scanning only once: An end-to-end framework for fast temporal grounding in long videos. In *ICCV*, 2023. 2, 5, 6, 8

[37] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *EMNLP*, 2014. 6, 7

[38] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *NeurIPS*, 2022. 6

[39] 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 *ICML*, 2021. 6

[40] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *TACL*, 2013. 5

[41] Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *WACV*, 2020. 2

[42] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2

[43] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 1, 5

[44] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, 2016. 2

[45] 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 *CVPR*, 2022. 1, 2, 5

[46] Mattia Soldan, Mengmeng Xu, Sisi Qu, Jesper Tegner, and Bernard Ghanem. VLG-net: Video-language graph matching network for video grounding. In *ICCVW*, 2021. 1, 2, 6, 7, 8

[47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019. 2

[48] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2, 6, 7

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 5

[50] 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 *CVPR*, 2021. 2, 6, 8

[51] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *AAAI*, 2020. 2, 6, 8

[52] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding. In *AAAI*, 2022. 8

[53] Sangmin Woo, Jinyoung Park, Inyong Koo, Sumin Lee, Minki Jeong, and Changick Kim. Explore and match: End-to-end video grounding with transformer. *arXiv preprint arXiv:2201.10168*, 2022. 2, 3

[54] Shaoning Xiao, Long Chen, Jian Shao, Yueting Zhuang, and Jun Xiao. Natural language video localization with learnable moment proposals. In *EMNLP*, 2021. 2

[55] 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 *AAAI*, 2021. 6, 8

[56] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Si-

gal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, 2019. 2, 6, 8

[57] Yitian Yuan, Xiaohan Lan, Xin Wang, Long Chen, Zhi Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In *ACM MM-HUMA Workshop*, 2021. 5

[58] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *NeurIPS*, 2019. 2, 8

[59] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*, 2019. 2

[60] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *CVPR*, 2020. 2, 3, 5, 8

[61] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Action-Former: Localizing moments of actions with transformers. In *ECCV*, 2022. 2, 4, 5

[62] 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 *CVPR*, 2019. 2, 3

[63] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Siow Mong Rick Goh. Parallel attention network with sequence matching for video grounding. In *Findings of ACL*, 2021. 2

[64] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *ACL*, 2020. 2, 5, 6, 7

[65] Lingyu Zhang and Richard J Radke. Natural language video moment localization through query-controlled temporal convolution. In *WACV*, 2022. 2

[66] Mingxing Zhang, Yang Yang, Xinghan Chen, Yanli Ji, Xing Xu, Jingjing Li, and Heng Tao Shen. Multistage aggregated transformer network for temporal language localization in videos. In *CVPR*, 2021. 2, 3, 6, 8

[67] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, 2020. 1, 2, 3, 5, 6, 7, 8

[68] Yang Zhao, Zhou Zhao, Zhu Zhang, and Zhijie Lin. Cascaded prediction network via segment tree for temporal video grounding. In *CVPR*, 2021. 2, 6, 8

[69] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-IoU loss: Faster and better learning for bounding box regression. In *AAAI*, 2020. 5

[70] Hao Zhou, Chongyang Zhang, Yan Luo, Yanjun Chen, and Chuanping Hu. Embracing uncertainty: Decoupling and de-bias for robust temporal grounding. In

CVPR, 2021. 2, 6, 8

[71] Jiahao Zhu, Daizong Liu, Pan Zhou, Xing Di, Yu Cheng, Song Yang, Wenzheng Xu, Zichuan Xu, Yao Wan, Lichao Sun, et al. Rethinking the video sampling and reasoning strategies for temporal sentence grounding. In *EMNLP Findings*, 2022. 2, 7, 8