Temporal Sentence Grounding in Videos: A Survey and Future Directions

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou

Abstract—Temporal sentence grounding in videos (TSGV), *a.k.a.,* natural language video localization (NLVL) or video moment retrieval (VMR), aims to retrieve a temporal moment that semantically corresponds to a language query from an untrimmed video. Connecting computer vision and natural language, TSGV has drawn significant attention from researchers in both communities. This survey attempts to provide a summary of fundamental concepts in TSGV and current research status, as well as future research directions. As the background, we present a common structure of functional components in TSGV, in a tutorial style: from feature extraction from raw video and language query, to answer prediction of the target moment. Then we review the techniques for multimodal understanding and interaction, which is the key focus of TSGV for effective alignment between the two modalities. We construct a taxonomy of TSGV techniques and elaborate the methods in different categories with their strengths and weaknesses. Lastly, we discuss issues with the current TSGV research and share our insights about promising research directions.

Index Terms—Temporal Sentence Grounding in Video, Natural Language Video Localization, Video Moment Retrieval, Temporal Video Grounding, Multimodal Retrieval, Cross-modal Video Retrieval, Multimodal Learning, Video Understanding, Vision and Language.

1 Introduction

VIDEO has gradually become a major type of information transmission media, thanks to the fast development and innovation in communication and media creation technologies. A video is formed from a sequence of continuous image frames possibly accompanied by audio and subtitles. Compared to image and text, video conveys richer semantic knowledge, as well as more diverse and complex activities. Despite the strengths of video, searching for content from the video is challenging. Thus, there is a high demand for techniques that could quickly retrieve video segments of user interest, specified in natural language.

1.1 Definition and History

Given an untrimmed video, temporal sentence grounding in videos (TSGV) is to retrieve a video segment, also known as a temporal moment, that semantically corresponds to a query in natural language *i.e.,* sentence. As illustrated in Fig. 1, for the query "*A person is putting clothes in the washing machine.*", TSGV needs to return the start and end timestamps (*i.e.,* $9.6s$ and $24.5s$) of a video moment from the input video as the answer. The answer moment should contain actions or events described by the query.

As a fundamental vision-language problem, TSGV also serves as an intermediate step for various downstream vision-language tasks, such as video question answering and video-grounded dialogue[1]. These tasks require localizing relevant moments about questions, then discovering or generating answers to the input questions by analyzing the retrieved moments. Naturally, TSGV connects computer vision (CV) and natural language processing (NLP) and benefits from the advancements made in both areas.

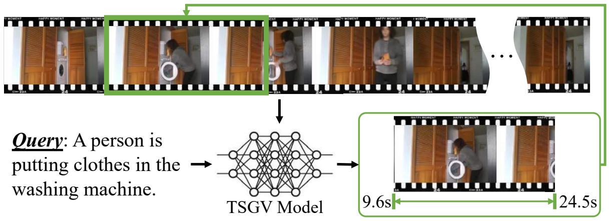

Query: A person is putting clothes in the washing machine.
TSGV Model
9.6s
24.5s

Fig. 1. An illustration of temporal sentence grounding in videos (TSGV).

TSGV also shares similarities with some classical tasks in both CV and NLP. For instance, video action recognition (VAR) [1]–[4] in CV is to detect video segments, which perform specific actions in video. Although VAR localizes temporal segments with activity information, it is constrained by the predefined action categories. TSGV is more flexible and aims to retrieve complicated and diverse activities from video via arbitrary language queries. In this sense, TSGV needs a semantic understanding of both video and language, as well as the multimodal interaction between them. TSGV is similar to the reading comprehension (RC) task in NLP [5]–[8], which is to retrieve a span of words from the text to answer a question. The core of RC is the interaction between text passages and query. TSGV models the interaction between two different modalities, making it more arduous and challenging.

TSGV was proposed in 2017 [9], [10]; the task immediately drew significant attention from researchers. Early solutions mainly adopt an ineffective two-stage approach, first to sample moments as candidate answers, then to score these candidates [9]–[13]. Subsequent solutions focus more on effective and efficient multimodal interactions between video and query. A lot of methods are developed, including proposal-based [14]–[18], proposal-free [19]–[23], reinforcement learning-based [24]–[26], and weakly-supervised [27]–[31] methods, etc.

- *H. Zhang is with School of Computer Science and Engineering, Nanyang Technological University, Singapore, 639798.*
- *A. Sun is with S-Lab, Nanyang Technological University, Singapore, 639798.*
- *W. Jing is with Alibaba Group, China, 311121.*
- *J.T. Zhou is with Centre for Frontier AI Research, A*STAR, Singapore, 138632.*
- *Corresponding author: A. Sun (Email: axsun@ntu.edu.sg).*

[1] For detailed relations between TSGV and other vision-language tasks, please refer to Appendix Section A.2.

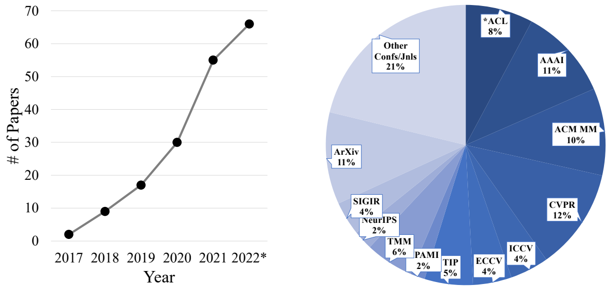


Fig. 2. Statistics of the collected papers in this survey. Left: number of papers published each year (till September 2022). Right: distribution of papers by venue, where *ACL denotes the series of conferences hosted by the Association for Computational Linguistics.

In this survey, we aim to provide a comprehensive and systematic review of TSGV research. We collect papers from reputable conferences and journals in CV, NLP, MM, IR, and machine learning areas, *e.g.,* CVPR, ECCV, ICCV, WACV, BMVC, ACL, EMNLP, NAACL, SIGIR, ACM MM, NeurIPS, AAAI, IJCAI, and TPAMI, TMM, TIP, etc. The papers were mainly published from 2017 to 2022[2]. For the paper collection, we primarily rely on academic search engines and digital libraries, such as IEEE Xplore, ACM Digital Library, ScienceDirect, Springer, ACL Anthology, CVF Open Access, etc. We also adopt Google Scholar to collect papers in other conferences/journals, and open-sourced articles.[3] Fig. 2 summarizes the statistics of the collected papers.

1.2 The User's Dilemma and the Role of Expertise

The availability of a vast collection of TSGV methods easily confounds a researcher or practitioner attempting to select or design an algorithm suitable for a specific problem at hand. Existing surveys [32]–[34] summarize the progress of TSGV research and establish taxonomies of methods based on their task formulation and architecture. Being the first survey, the taxonomy presented in Yang *et al.* [32] is relatively incomplete and coarse. Liu *et al.* [33] propose a pipeline of the TSGV model by partitioning it into three components and categorizing the existing solutions into supervised and weakly-supervised groups. However, their taxonomy is unable to cover various TSGV approaches as well. Lan *et al.* [34] present a more complete taxonomy, with detailed illustrations and comparison between different categories of methods. Benchmark datasets and evaluation metrics are also covered. The most recent survey by Liu *et al.* [35] covers more TSGV methods and provides an efficiency comparison among methods. Similar to prior work, this survey lists current research but does not provide an in-depth critical analysis of methods and insights into future directions.

Our survey covers more recent developments in TSGV research. By abstracting common generalities in all methods, we summarize different types of TSGV methodologies and reveal a common pipeline of the TSGV model. We also establish a more comprehensive taxonomy and conclude more concrete and promising future research directions. All existing surveys focus on summarizing existing TSGV methods and stating future research directions. However, they do not provide a critical analysis of existing TSGV methods. More importantly, common questions from researchers/practitioners are not well addressed in existing surveys: (i) How should TSGV data be processed? (ii) How should the data be used in a particular TSGV method? (iii) What does a TSGV method generally look like and how it works? and (iv) Which model assessment is appropriate for a particular TSGV method? Our aim is to provide these perspectives on the composition of TSGV methods and state-of-the-art TSGV research. With such perspectives, an informed practitioner is able to confidently assess the trade-offs of various TSGV methods and make a competent decision on designing a TSGV solution with a suite of techniques.

This survey is organized as follows. In Section 2, we present a general pipeline of TSGV methods and interpret the technical details in a tutorial style. It provides readers with background on what a TSGV model generally looks like, its I/O, and functional components. Section 3 summarizes the major benchmark datasets and evaluation metrics. Section 4 classifies TSGV solutions into categories, elaborates on the methods in each category, and discusses their pros and cons. Section 5 summarizes the current research progress. Section 6 discusses open issues and further research directions. Section 7 concludes this paper.

2 Background

There are no theoretical guidelines that reveal a common structure or pipeline of a TSGV method. Despite various sophisticated architectures in different methods, conceptually, a TSGV method generally contains six components shown in Fig. 3. The dotted line in the figure indicates that the proposal generator is an optional component, and it may be placed at different stages. We brief these main components to provide the necessary background to the readers before we zoom into the technical details in Section 4.

A TSGV method takes a video-query pair as input, where the video is a collection of consecutive image frames, and the query is a sequence of words. The preprocessor prepares inputs for feature extraction, *e.g.,* downsampling and resizing image frames in the video, and tokenizing words in the query sentence. Feature extractor converts the video frames and query words into their corresponding vector feature representations. Then the encoder module maps the video and query features to the same dimension and aggregates contextual information to enhance the feature representation. The interactor module, an essential component in TSGV, learns the multimodal representations by modeling the cross-modal interaction between video and query. Finally, the answer predictor generates moment predictions based on the learned multimodal representations. For proposal-based methods, the answer predictor makes predictions based on the proposals generated by the proposal generator. A proposal can be considered as a candidate answer moment, which can be generated at different stages. An example proposal is a video segment sampled from the input video. Proposal-free methods predict answers directly without the need of generating candidate answers.

Before we elaborate the details of each component, we define the following notations. Given a TSGV dataset, we denote its video corpus as $\mathcal{V} = \{V^1, V^2, \dots, V^N\}$ and its query set as $\mathcal{Q} = \{Q^1, Q^2, \dots, Q^M\}$, where N and M are the number of videos and queries, respectively. Note that multiple queries can be posed on the same video with its different moments as answers; typically $M \geq N$ in TSGV datasets. Given a video-query pair, a

[2] The paper collection was conducted lastly on 2022-09-18.

[3] A number of keywords and their combinations are utilized for paper searching, including moment, grounding, localization, language query, video retrieval, moment retrieval, video grounding, temporal grounding, moment localization, video localization, temporal localization, temporal language grounding, temporal sentence grounding, etc.

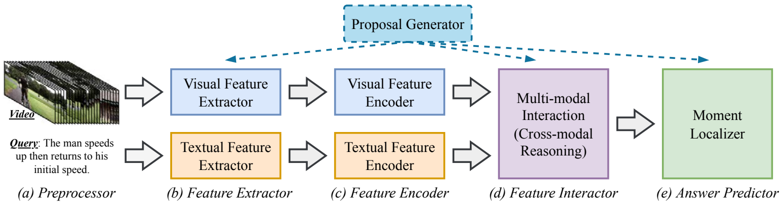


Fig. 3. A general pipeline for temporal sentence grounding in videos.

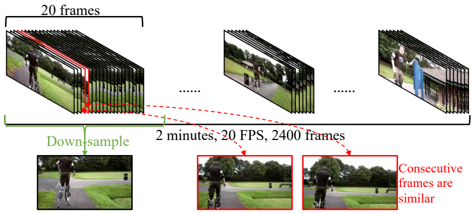

20 frames
......
......
Down-sample
2 minutes, 20 FPS, 2400 frames
Consecutive frames are similar

Fig. 4. An example of video frames down-sampling.

video V contains T frames $V=[f_1,f_2,\dots,f_T]$ and a query Q has m words $Q=[q_1,q_2,\dots,q_m]$, the start and end time of the ground truth moment are denoted by τ_s and τ_e, $1\leq\tau_s<\tau_e\leq T$. Here, we use the frame index to represent time points, based on a fixed frame rate or fps. Mathematically, TSGV is to retrieve the target moment starting from τ_s and ending at τ_e by giving a video V and query Q, *i.e.,* $\mathcal{F}_{TSGV}:(V,Q)\mapsto(\tau_s,\tau_e)$.

2.1 Preprocessor

Video is a series of still images and the number of frames can be very large. For instance, a 2-minute video with 20 fps has $2,400$ frames in total. Thus, it is infeasible (and often unnecessary) to process every frame in a video due to computational cost. Besides, video is continuous, *i.e.,* changes between consecutive frames are usually small and smooth. Hence, it is reasonable to downsample video for efficient computation. As shown in Fig. 4, if we sample 1 frame from every 20 consecutive frames, we only need to process 120 frames instead of 2400 frames for this 2-minute video. With a downsample rate r_{ds}, the number of video frames becomes $T'=T/r_{ds}$. Downsample rate has a direct impact on video quality and should be carefully selected depending on the dataset.

Language query is discrete and words in a sentence demonstrate syntactic structure. Different word combinations lead to very different semantic meanings. For instance, in a query sentence "*The man speeds up then returns to his initial speed.*", the words "*initial*" and "*speed*" carry different meanings, and their combination describes a specific scene. For preprocessing, a query is typically tokenized into word tokens. If a query contains too many words, a common strategy is truncation, *i.e.,* taking a fixed number of words from the beginning and discarding the rest.

2.2 Feature Extractor

The feature extractor bridges the raw inputs and the model by converting inputs into feature representations.

Textual Feature Extractor maps a query sentence to an embedding space, which can be categorized into token-level and sentence-level extractors. Token-level extractor converts each word into its corresponding word embedding by using pre-trained word embeddings (PWE), *e.g.,* Word2Vec [36] and GloVe [37], or pre-trained language models (PLM), *e.g.,* BERT [38] and RoBERTa [39]. We represent token-level extraction as:

$$Q=[q_1,\dots,q_m]\xmapsto{\text{PWE/PLM}}\mathbf{Q}=[\mathbf{q}_1,\dots,\mathbf{q}_m]\in\mathbb{R}^{m\times d_q}, \quad (1)$$

where d_q denotes the word embedding dimension.

Sentence-level extractor encodes the entire query into a sentence feature in d_s dimension, by using pre-trained sentence encoder (PSE), *e.g.,* Skip-Thought [40], InferSent [41], Sentence-BERT [42], or PWE/PLM with a trainable sentence encoder (TSE). We represent the process as:

$$\begin{aligned} &Q=[q_1,\dots,q_m]\xmapsto{\text{PSE}}\mathbf{q}_s\in\mathbb{R}^{d_s}, \text{ or}\\ &Q=[q_1,\dots,q_m]\xmapsto{\text{PWE/PLM}}\mathbf{Q}\in\mathbb{R}^{m\times d_q}\xmapsto{\text{TSE}}\mathbf{q}_s\in\mathbb{R}^{d_s} \end{aligned} \quad (2)$$

Visual Feature Extractor converts video frames to a sequence of visual features. Depending on whether proposals are generated directly on the input video, there are two types of feature extraction.

Recall that a proposal is a candidate answer. A straightforward approach is to sample video segments from the input video as proposals. Proposals may contain a different number of frames. Suppose there are n_{seg} video segments as proposals, the feature extraction process is described as:

$$\begin{aligned} &V\in\mathbb{R}^{T'\times frame}\xmapsto{\text{proposals}}\{\text{segment}_i\in\mathbb{R}^{\chi\times frame}\}_{i=1}^{n_{\text{seg}}}\\ &\xmapsto[\text{extractor}]{\text{visual feature}}\mathbf{V}=\{\mathbf{v}_{p,i}\in\mathbb{R}^{d_v}\}_{i=1}^{n_{\text{seg}}}, \end{aligned} \quad (3)$$

where χ is the number of frames in a proposal, and d_v denotes the dimension of extracted features. The task becomes to determine whether a proposal represented by $\mathbf{v}_{p,i}$ is the correct answer.

If proposals are not generated directly from the input video, then the video is uniformly decomposed into a sequence of non-overlapping snippets. Suppose there are n_{snp} video snippets and each snippet contains ξ frames, the extraction process is:

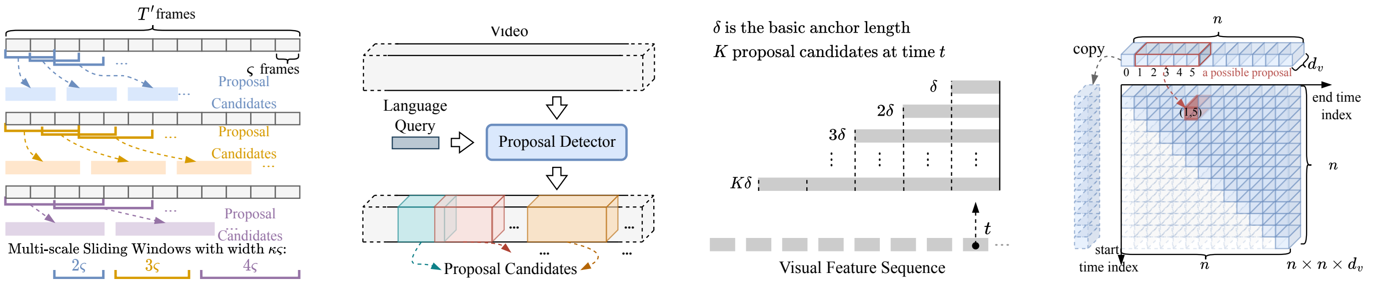
$$\begin{aligned} &V\in\mathbb{R}^{T'\times frame}\xmapsto{\text{decompose}}[\text{snippet}_i]_{i=1}^{n_{\text{snp}}}\in\mathbb{R}^{n_{\text{snp}}\times\xi\times frame}\\ &\xmapsto[\text{extractor}]{\text{visual feature}}\mathbf{V}=[\mathbf{v}_i]_{i=1}^{n_{\text{snp}}}\in\mathbb{R}^{n_{\text{snp}}\times d_v}. \end{aligned} \quad (4)$$

Here we distinguish "video snippet" from "video segment". A video segment is sampled as a proposal to match the target moment, and a video snippet is a very short clip that only contains a few frames, *i.e.,* $\xi<<\chi$ in general. Furthermore, as each video segment is one candidate answer, the video segments are irrelevant to each other and they are further processed separately in TSGV. As very short clips, video snippets are maintained in sequence, and are jointly processed in later stages.

Each frame is a still image. From frames to features, the commonly used pre-trained visual feature extractors are (i) 3D-ConvNet for action recognition, *e.g.,* C3D [1] or I3D [3], and (ii) 2D-ConvNet for object detection, *e.g.,* VGG [43] or ResNet [44].

2.3 Feature Encoder and Feature Interactor

Feature encoder maps visual and textual features to the same dimension, and refines their feature representations by encoding their corresponding contextual information. Existing TSGV methods use various feature encoders, from simple multi-layer perceptrons to complex transformers and graph neural networks.



(a) Sliding window (SW) strategy (b) Proposal generated (PG) strategy (c) Anchor-based strategy (d) 2D-Map strategy

Fig. 5. Illustration of sliding window, proposal generated, anchor-based, and 2D-Map strategies.

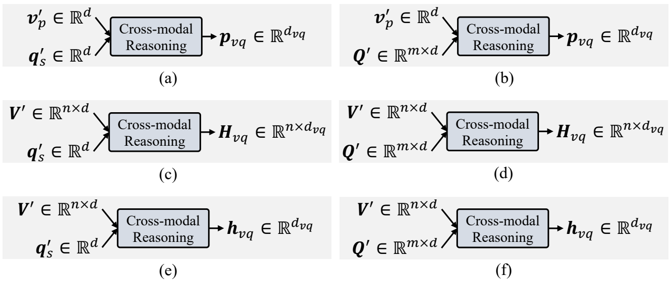

$v'_p \in \mathbb{R}^d$
$q'_s \in \mathbb{R}^d$
Cross-modal Reasoning
$p_{vq} \in \mathbb{R}^{d_{vq}}$
(a)
$v'_p \in \mathbb{R}^d$
$Q' \in \mathbb{R}^{m \times d}$
Cross-modal Reasoning
$p_{vq} \in \mathbb{R}^{d_{vq}}$
(b)
$V' \in \mathbb{R}^{n \times d}$
$q'_s \in \mathbb{R}^d$
Cross-modal Reasoning
$H_{vq} \in \mathbb{R}^{n \times d_{vq}}$
(c)
$V' \in \mathbb{R}^{n \times d}$
$Q' \in \mathbb{R}^{m \times d}$
Cross-modal Reasoning
$H_{vq} \in \mathbb{R}^{n \times d_{vq}}$
(d)
$V' \in \mathbb{R}^{n \times d}$
$q'_s \in \mathbb{R}^d$
Cross-modal Reasoning
$h_{vq} \in \mathbb{R}^{d_{vq}}$
(e)
$V' \in \mathbb{R}^{n \times d}$
$Q' \in \mathbb{R}^{m \times d}$
Cross-modal Reasoning
$h_{vq} \in \mathbb{R}^{d_{vq}}$
(f)

Fig. 6. The common input/output feature formats of feature interactor in TSGV. $\mathbf{p}_{vq} \in \mathbb{R}^{d_{vq}}$ denotes the learned multimodal proposal feature; $\mathbf{H}_{vq} = [\mathbf{h}^1_{vq}, \dots, \mathbf{h}^n_{vq}] \in \mathbb{R}^{n \times d_{vq}}$ is the multimodal snippet feature sequence; $\mathbf{h}_{vq} \in \mathbb{R}^{d_{vq}}$ is the pooled multimodal snippet feature. d_{vq} denotes the dimension of output multimodal feature.

The design of the feature encoder highly depends on the model architecture.

Briefed in Section 2.1, there are token-level and sentence-level query features. There are also two types of visual features, depending on whether a proposal generator is applied on input video, *i.e.,* proposal feature and video snippet feature sequence. Let d be the target dimension for both visual and textual features. Mapping of sentence-level and token-level query features is defined as:

$$\begin{aligned} \mathbf{q}_s \in \mathbb{R}^{d_s} &\xrightarrow[\text{encoder}]{\text{textual feature}} \mathbf{q}'_s \in \mathbb{R}^{d}, \text{ and} \\ \mathbf{Q} \in \mathbb{R}^{m \times d_q} &\xrightarrow[\text{encoder}]{\text{textual feature}} \mathbf{Q}' \in \mathbb{R}^{m \times d}. \end{aligned} \tag{5}$$

For the proposal feature and video snippet feature sequence, the mapping is written as:

$$\begin{aligned} \mathbf{v}_p \in \mathbb{R}^{d_v} &\xrightarrow[\text{encoder}]{\text{visual feature}} \mathbf{v}'_p \in \mathbb{R}^{d}, \text{ and} \\ \mathbf{V} \in \mathbb{R}^{n \times d_v} &\xrightarrow[\text{encoder}]{\text{visual feature}} \mathbf{V}' \in \mathbb{R}^{n \times d}, \end{aligned} \tag{6}$$

where we simply use $\mathbf{v}_p \in \mathbb{R}^{d_v}$ to represent the visual feature of a proposal, and n to replace n_{snp}.

Feature interactor, an essential component in any TSGV method, aims to learn cross-interaction between video and query. Recall the goal of TSGV is to retrieve a target moment from the video that *semantically corresponds* to the query. Thus, the feature interactor requires to understand the semantic meaning of query and to recognize various activities in the video simultaneously. It then performs to fuse query and video representations by emphasizing the portion of the video content that is most relevant to the query semantics. In general, the quality of the feature interactor determines the performance of a TSGV method to a large extent.

Fig. 6 summarizes the various input and output formats of different feature interactors among existing TSGV methods. The input is determined by the types of query features (token-level or sentence-level), and the types of visual features (proposal or snippet sequence). The common output feature formats include (i) the learned multimodal proposal feature $\mathbf{p}_{vq} \in \mathbb{R}^{d_{vq}}$, (ii) the multimodal snippet feature sequence $\mathbf{H}_{vq} = [\mathbf{h}^1_{vq}, \dots, \mathbf{h}^n_{vq}] \in \mathbb{R}^{n \times d_{vq}}$, and (iii) the pooled multimodal snippet feature $\mathbf{h}_{vq} \in \mathbb{R}^{d_{vq}}$. Here, d_{vq} is the dimension of the multimodal feature.

Output format of the feature interactor is highly correlated with the answer predictor in a TSGV method. Answer predictor may depend on proposals that can be generated at different stages. We next brief the proposal generation before the answer predictor.

2.4 Proposal Generation

Depending on whether a proposal generation module is used, existing TSGV methods can be roughly categorized into *proposal-based* and *proposal-free* methods. As shown in Fig. 3, the proposal generator can be integrated into the model at various positions. For instance, proposals can be directly sampled from the input video. Proposals can also be generated before or after the feature interactor based on the visual features. Anchor-based methods generate proposals during answer prediction. A method may also engage multiple proposal generation strategies.

Sliding window-based (SW) strategy [9]–[13], [45]–[51] generates proposal candidates by densely sampling fixed-length video segments on the input video, using pre-defined multi-scale sliding windows. SW strategy is usually performed directly on video frames. Illustrated in Fig. 5(a), given a downsampled video with T' frames and a set of sliding windows, each sliding window samples video segments sequentially, with a preset overlap ratio (r_o). In our illustration, we use three different sliding windows $sw \in \{\kappa\varsigma\}_{\kappa=2,3,4}$ (ς is a basic window size) and set $r_o = 0.5$. Overlap ratio is necessary to increase the chance of covering the target moment. Then we have a set of video segments as proposals.

Proposal-generated (PG) strategy [14], [52]–[58] produces proposals by utilizing auxiliary modules, *e.g.,* pre-trained segment proposal network (SPN) [52] or carefully designed proposal detector. The PG strategy is usually performed on visual features, but it involves the query as input to guide its proposal generation process, illustrated in Fig. 5(b). Hence, the proposals generated are related to the query. Depending on the position of the proposal detector, PG strategy may also involve feature encoder and interactor.

Anchor-based strategy [15]–[17], [59]–[69] generates proposals with pre-set multi-scale anchors. Different from the SW strategy, it is performed on the encoded visual features and is integrated in the answer predictor. Suppose we have K different scale anchors, and the length of a basic anchor is δ. Fig. 5(c) plots a commonly used anchor-based strategy. This strategy applies K preset anchors to generate proposals, ended at a time step t, where t is the index of the multimodal visual feature in the feature sequence.

Another version of anchor-based strategy is 2D-Map strategy [4], [18], [70]–[82]. Different from the standard anchor-based strategy above, 2D-Map strategy is usually applied after the feature extractor, *i.e.,* before answer predictor. It generates proposals by modeling the temporal relations between video moments through a two-dimensional map. One dimension indicates the starting time of a moment; the other indicates the end time. Given a visual feature sequence with $n \times d_v$, all possible proposal candidates are computed based on a 2D temporal feature map. Shown in Fig. 5(d), a candidate proposal representation can be computed by max-pooling the corresponding visual features across a specific time span, resulting in the 2D feature map with $n \times n \times d_v$. Note the start (a) and end (b) timestamps of a proposal candidate should satisfy $a \leq b$. Therefore, only proposal candidates that locate in the upper triangular part of the 2D map are valid.

2.5 Answer Predictor and Objective

Answer predictor is responsible for predicting the position of a target moment based on the learned multimodal features. Next, we brief the commonly used answer predictors and their corresponding objectives, for both proposal-based and proposal-free methods. Methods may combine multiple answer predictors or incorporate various auxiliary objectives to boost performance. In this background section, we only focus on the main objectives.

For proposal-based methods, the answer predictor computes a score for each proposal. Ideally, a proposal gets a higher score if it is closer to the ground truth moment. Specifically, given a multimodal proposal feature $\mathbf{p}_{vq}$, its score is computed as $s = \sigma(\mathcal{A}(\mathbf{p}_{vq})) \in \mathbb{R}^1$, where $\mathcal{A}$ is answer predictor and σ is an (optional) activation function. Then, the proposal with the highest score is selected as the answer. If proposals are generated by anchor-based strategy, the score is computed based on the multimodal snippet feature sequence $\mathbf{H}_{vq}$ by applying multi-scale anchors in the answer predictor.

Various learning objectives have been developed for proposal-based methods. The alignment loss [9], [11]–[13], [45], [46], [48]–[51], [54], [56] is commonly used for SW and PG strategies, which is defined as:

$$\mathcal{L}_{aln} = \gamma \log(1 + e^{-s_{i,i}}) + \sum_{j=0, j\neq i}^{N_{neg}} \log(1 + e^{s_{i,j}}), \qquad (7)$$

where $s_{i,i}$ is the score of aligned (or positive) proposal-query pair, and $s_{i,j}$ is the score of misaligned (or negative) pair; γ is a hyper-parameter to control the weight between positive and negative pairs; N_{neg} is the number of negative pairs. For a given query, a proposal is considered positive if it has a good overlap with the ground truth moment, measured by IoU (intersection area over union area). Otherwise, it is negative. Nevertheless, a negative pair can also be constructed by replacing a random query or pairing random but unmatched proposals and queries. In general, $\mathcal{L}_{aln}$ encourages aligned proposal-query pairs to have positive scores and misaligned pairs to have negative scores. Besides, triple-based ranking loss [10], [14], [45], [47], [53], [58], [70], [83] has also been used for SW and PG strategies:

$$\mathcal{L}_{triple} = \max(0, \eta + s' - s) \qquad (8)$$

where s denotes the score of matched proposal-query pair and s' is the score of mismatched proposal-query pair. Similarly, $\mathcal{L}_{triple}$ encourages similarities between aligned pairs to be greater than misaligned pairs by some margin $\eta > 0$.

For anchor-based and 2D-Map strategies, binary cross-entropy loss [15]–[18], [56], [59]–[69], [71]–[77], [79]–[82], [84]–[87] is usually adopted, which is defined as:

$$\mathcal{L}_{bce} = \gamma \cdot y \cdot \log s + (1 - y) \cdot \log(1 - s) \qquad (9)$$

where γ is an optional balance weight, determined based on the number of positive and negative samples. y is the corresponding anchor label for the proposal; $y = 1$ if the proposal candidate has IoU with ground truth moment larger than a threshold θ, *i.e.,* positive. Otherwise $y = 0$. y may also be defined as the scaled IoU value between the proposal and the ground truth moment.

Proposal-free methods do not generate proposals. Instead, they use a regressor or a span predictor as the answer predictor. Specifically, regression-based predictor aims to regress the start and end times of the target moment directly. It takes the pooled multimodal snippet feature $\mathbf{h}_{vq}$ as input and predicts the temporal positions (t_s, t_e). Mathematically, we represent this process as $(t_s, t_e) = \sigma(\mathcal{A}(\mathbf{h}_{vq})) \in \mathbb{R}^2$, where $\mathcal{A}$ denotes the regressor, and σ is (optional) Sigmoid activation to normalize the output to $[0, 1]$. Given a predicted (t_s, t_e) and the normalized ground truth (τ_s, τ_e), the smoothed L_1 loss [17], [19], [56], [60], [66], [88]–[94], MSE loss [22], [55], [57], [85], [95]–[97] or Huber loss [98], [99], *i.e.,* $R \in \{\text{smooth}_{L_1}, \text{MSE}, \text{Huber}\}$, is commonly used as learning objectives:

$$\mathcal{L}_{reg} = R(t_s - \tau_s) + R(t_e - \tau_e), \qquad (10)$$

Span predictor also predicts the start and end boundaries of the target moment directly. Different from repression-based predictor, span predictor computes the probability of each video snippet being the start and end points of the target moment. Specifically, it takes the multimodal snippet feature sequence $\mathbf{H}_{vq}$, and computes the start and end boundary scores as $(\mathbf{S}_s, \mathbf{S}_e) = \mathcal{A}(\mathbf{H}_{vq}) \in \mathbb{R}^{n\times 2}$. Then, the probability distributions of boundaries are computed by $\mathbf{P}_s = \text{softmax}(\mathbf{S}_s) \in \mathbb{R}^n$ and $\mathbf{P}_e = \text{softmax}(\mathbf{S}_e) \in \mathbb{R}^n$, where $\mathbf{P}^t_{s/e}$ denotes the probability of t-th snippet be the start/end boundary. Cross-entropy loss [20], [21], [23], [55], [97], [100]–[109] and Kullback-Leibler (KL) divergence [57], [85], [110]–[113] are both commonly used for span prediction. The cross-entropy objective is defined as:

$$\mathcal{L}_{span} = f_{XE}(\mathbf{P}_s, \mathbf{Y}_s) + f_{XE}(\mathbf{P}_e, \mathbf{Y}_e), \qquad (11)$$

where f_{XE} is the cross-entropy loss; $\mathbf{Y}_s$ and $\mathbf{Y}_e$ denote the ground truth labels for the start and end boundaries, respectively. $\mathbf{Y}_{s/e}$ is a n-dim one-hot vector, which is generated by setting the index of the snippet contains $\tau_{s/e}$ as 1, and others as 0. Similarly, the KL-divergence objective is defined as:

$$\mathcal{L}_{span} = D_{KL}(\mathbf{P}_s || \hat{\mathbf{Y}}_s) + D_{KL}(\mathbf{P}_e || \hat{\mathbf{Y}}_e), \qquad (12)$$

where D_{KL} denotes KL-divergence; $\hat{\mathbf{Y}}_s$ and $\hat{\mathbf{Y}}_e$ are the ground truth start and end boundary distributions. Not specified in an one-hot $\mathbf{Y}_{s/e}$, the ground truth boundary distribution is formulated

TABLE 1
Statistics of the TSGV benchmark datasets. Different queries may correspond to the same moment.

Dataset	DiDeMo	Charades-STA	ActivityNet Captions	TACoS$_{\text{org}}$	TACoS$_{\text{2DTAN}}$	MAD
Video Source	Flickr	Homes	YouTube	Lab Kitchen		Movie
Domain	Open	Indoor Activity	Open	Cooking		Open
# Videos	10,464	6,672	14,926	127		650
# Moments	26,892	11,767	71,953	3,290	7,069	-
# Queries (or Annotations)	40,543	16,124	71,953	18,818	18,227	384,600
Average # Annotations per Video	3.87	2.42	4.82	148.17	143.52	-
Vocabulary Size	7,785	1,303	15,505	2,344	2,287	61,400
Average Video Length (seconds)	30.00	30.60	117.60	286.59		6,646.20
Min / Max Video Length (seconds)	-	5.50 / 194.33	1.58 / 755.11	48.30 / 1,402.18		-
Average Moment Length (seconds)	-	8.09	37.14	6.10	27.88	4.10
Min / Max Moment Length (seconds)	-	1.68 / 80.80	0.05 / 408.80	0.31 / 166.97	0.48 / 843.20	-
Average Query Length (words)	-	7.22	14.41	10.05	9.42	12.70
Min / Max Query Length (words)	-	3 / 13	4 / 91	2 / 229	2 / 69	-

as $\hat{\mathbf{Y}}_{s/e} \sim \mathcal{N}(\tau_{s/e}, \sigma^2_{std})$, where $\mathcal{N}(\mu, \sigma^2_{std})$ is the normal distribution with expectation μ and standard deviation σ_{std}.

To summarize, we brief the main components of the TSGV method from input processing to answer prediction. Although existing TSGV methods may contain more sophisticated structures and diverse ancillary modules, their model frameworks generally follow this pipeline. Among the components, the effectiveness of the feature interactor highly affects TSGV performance. Proposal generation strategies are highly correlated with the design of the answer predictor, and each strategy has its own advantages and drawbacks. Lastly, all methods rely on effective feature extractors, mainly developed in computer vision and natural language processing areas.

3 Datasets and Measures

Datasets are essential resources for building and evaluating TSGV methods. We review benchmark datasets and evaluation metrics.

3.1 Benchmark Datasets

A TSGV dataset typically contains a collection of videos. Each video may come with one or more annotations, *i.e.,* moment-query pairs. Each annotation has a query corresponding to a moment in the video. A few TSGV datasets have been developed, covering various scenarios with distinct characteristics *e.g.,* different scenes, and activity complexities, summarized in Table 1.[4]

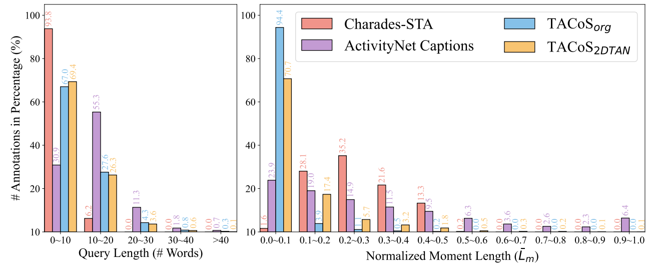
DiDeMo has its root in YFCC100M [114] dataset, and the latter contains over $100k$ Flickr videos about various human activities. Hendricks *et al.* [10] randomly select over $14,000$ videos, then split and label video segments. Each segment is a five-second video clip, hence the length of ground truth moment is five seconds. DiDeMo dataset consists of $10,464$ videos and $40,543$ annotations in total, on average 3.87 annotations per video. Note that the videos are released in the form of extracted visual features, hence we cannot provide detailed statistics in Table 1. Hendricks *et al.* [45] further collect a TEMPO dataset, which is built on top of DiDeMo, by augmenting language queries via the template model (template language) and human annotators (human language). Compared to DiDeMo, TEMPO contains more complex human-language queries.

Charades-STA is built by Gao *et al.* [9] from the Charades dataset [115]. The Charades dataset contains $9,848$ annotated videos about human daily indoor activities for video activity recognition. The original dataset provides $27,847$ video-level sentence descriptions, $66,500$ temporally localized intervals for 157 action categories, and $41,104$ labels for 46 object categories. Based on Charades, Gao *et al.* [9] design a semi-automatic way to construct Charades-STA. They first parse the activity labels from video descriptions using Stanford CoreNLP [116], then match the labels with sub-sentences, and finally align sub-sentences with the original label-indicated temporal intervals. A collection of (sentence query, target moment) pairs are generated as annotations. Because the original descriptions are quite short, Gao *et al.* [9] further combine consecutive descriptions into a more complex sentence to enhance the description complexity for test. Charades-STA contains $6,672$ videos and $16,124$ annotations. Average video length, moment length, and query length are 30.60 seconds, 8.09 seconds and 7.22 words, respectively.

ActivityNet Captions is developed by Krishna *et al.* [117] for dense video captioning task. However, the sentence-moment pairs in this dataset can naturally be adopted for TSGV task. The videos are taken from ActivityNet [118] dataset, a human activity understanding benchmark. ActivityNet provides samples from 203 activity classes, with an average of 137 untrimmed videos per class and 1.41 activity instances per video [118]. The official test set of ActivityNet Captions is withheld for competition, existing TSGV methods mainly use the official "val1" and/or "val2" development sets as test sets. Thus, statistics of ActivityNet Captions in Table 1 does not consider its official test set. In total, there are $14,926$ videos and $71,953$ annotations in ActivityNet Captions, where each video contains 4.82 temporally localized sentences on average. Average video and moment lengths are 117.60 and 37.14 seconds, respectively. Average query length is about 14.41 words.

TACoS dataset [119] is selected from the MPII Cooking Composite Activities dataset [120], originally developed for human activity recognition under specific scene, *i.e.,* composite cooking activities in lab kitchen. TACoS contains 127 videos, and each video is associated with two types of annotations: (1) fine-grained activity labels with temporal location, and (2) natural language descriptions with temporal locations. The natural language descriptions are from crowd-sourcing annotators, who describe the video content by sentences [119]. TACoS has $18,818$ moment-

[4] For DiDeMo and MAD datasets, we directly obtain their statistical results from the original papers. For others, we conduct statistics on raw datasets. We also filter out or modify some invalid annotations in each dataset.



(a) Query Length Distribution (b) Normalized Moment Length Distribution

Fig. 7. Statistics of query length and normalized moment length over benchmark datasets.

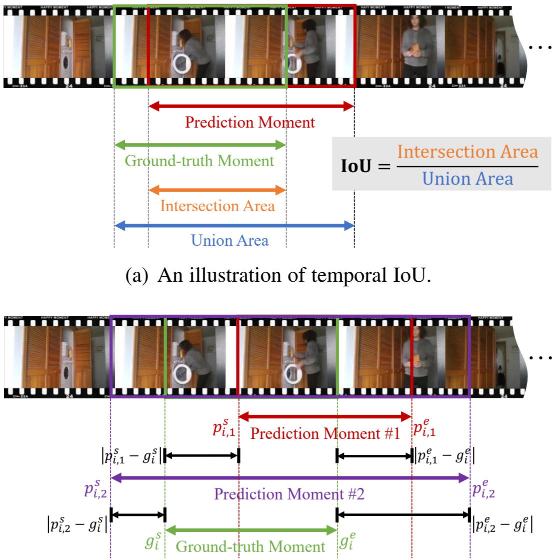
query pairs. Average video and moment lengths are 286.59 and 6.10 seconds, and average query length is 10.05 words. Each video in TACoS contains 148.17 annotations on average. We name this dataset TACoS$_{\text{org}}$ in Table 1. A modified version TACoS$_{\text{2DTAN}}$ is made available by Zhang *et al.* [18]. TACoS$_{\text{2DTAN}}$ has $18,227$ annotations. On average, there are 143.52 annotations per video. The average moment length and query length after modification are 27.88 seconds and 9.42 words, respectively.

MAD [121] is a large-scale dataset containing mainstream movies. Compared to previous datasets, MAD aims to avoid the hidden biases (detailed in Section 6.1) and provide accurate and unbiased annotations for TSGV. Instead of relying on crowd-sourced annotations, Soldan *et al.* [121] adopt a scalable data collection strategy. They transcribe the audio description track of a movie and remove sentences associated with actor's speech, to obtain highly descriptive sentences that are grounded in long-form videos. MAD contains 650 movies with over $1,200$ hours of video length in total. Average video duration is around 110 minutes. Each video in MAD is a full movie without pruning. MAD has $348,600$ queries with vocabulary size of $61,400$. Average query length is 12.7 words. Average length of temporal moment in MAD is merely 4.1 seconds, making the localization process more challenging.[5]

Videos in the aforementioned datasets may be from open domain or constrained in narrow and specific scenes (see Table 1). Open domain videos contain more diverse and complex activities, making them more challenging, but are closer to real-world scenarios. Although DiDeMo videos are from open domain, the answers in this dataset are in fixed-length, *i.e.,* five-second. The fixed length considerably reduces the complexity of finding answers in DiDeMo.

ActivityNet Captions and DiDeMo have a much larger vocabulary size than Charades-STA and TACoS, suggesting that the former two datasets provide rich variations in language queries. From the perspective of query length (see Fig. 7 (a)), a large portion of queries in Charades-STA (93.8%) and TACoS ($> 67.0\%$) has fewer than 10 words. Query length distribution indicates that ActivityNet Captions contain more queries with complicated expressions. Fig. 7 (b) depicts the normalized moment length ($\bar{L}_m$) distribution, against the length of its source video. A small $\bar{L}_m$ means the moment is difficult to retrieve due to moment sparsity [100]. The figure shows more than 70.7% of the moments in TACoS has $\bar{L}_m \leq 0.1$, while 70.1% moments in Charades-STA are in the range of $0.2 < \bar{L}_m \leq 0.5$.

[5] At the time of writing, MAD dataset is not publicly available.



(a) An illustration of temporal IoU.

(b) An illustration of dR@n,IoU@m.

Fig. 8. The temporal intersection over union (IoU), and the discounted-R@n,IoU@m (dR@n,IoU@m). p^s_i and p^e_i are start and timestamps of predicted moments, $g^{s/e}$ is start/end timestamp of ground-truth moment. $|\cdot|$ denotes absolute operation.

3.2 Evaluation Metrics

TSGV are generally evaluated by comparing predictions with ground truth annotations. The widely used measures include: mean IoU (mIoU), $\langle$R@n, IoU@$m\rangle$, and $\langle$dR@n, IoU@$m\rangle$.

Intersection over Union (IoU) is a metric commonly used in object detection [122]–[124] for measuring the similarity between two bounding boxes. Hence, the standard IoU in object detection is defined on a two-dimensional spatial space. TSGV focuses on temporal dimension only. The temporal IoU is adopted to measure similarity between the ground truth and predicted moments in TSGV, illustrated in Fig. 8(a). IoU is computed as the ratio of intersection area over union area between two moments, in the range of 0.0 to 1.0. A larger IoU means the two moments match better, and IoU $= 1.0$ denotes the exact match. The mIoU metric is the average temporal IoUs among all annotations in the test set. Mathematically, mIoU is defined as:

$$\text{mIoU} = \frac{1}{N_q} \sum_{i=1}^{N_q} \text{IoU}_i \tag{13}$$

where N_q denotes the total number of annotations or query samples, and IoU$_i$ is the IoU value of i-th sample.

The mIoU is computed based on the single top-ranked prediction for each query. However, given a query, the top-ranked prediction by a TSGV model may not always have the best match with ground truth. It is reasonable to relax the evaluation by considering top-n retrieved moments for each query. The $\langle$R@n, IoU@$m\rangle$ [125] is the percentage of queries, having at least one result whose temporal IoU with ground truth is larger than m among the top-n retrieved moments. For query q_i, among its top-n retrieved moments, if there exists at least one moment whose IoU with ground truth is larger than m, then q_i is considered as positive, denoted by $r(n, m, q_i) = 1$. Otherwise, $r(n, m, q_i) = 0$. Thus,

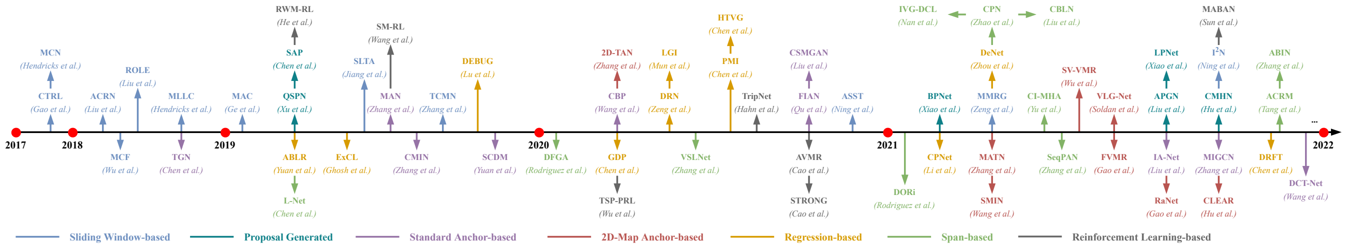


Fig. 9. Chronological overview of selected supervised TSGV methods in different categories. The methods plotted at the same position on the timeline are published in the same venue.

⟨R@n, IoU@m⟩ is calculated as:

$$\text{R@}n\text{,IoU@}m = \frac{1}{N_q}\sum_{i=1}^{N_q} r(n, m, q_i) \tag{14}$$

Yuan *et al.* [126] reveal that ⟨R@n, IoU@m⟩ is unreliable for small IoU thresholds. A method tends to generate long predictions if a substantial proportion of ground truth moments are long in a dataset. The method increases its chance of correct prediction under small IoU thresholds. Discounted-R@n, IoU@m ⟨dR@n, IoU@m⟩ is proposed to alleviate this problem [126]. This new measure leverages "temporal distance" between the predicted and ground truth moments to discount $r(n, m, q_i)$ value. ⟨dR@n, IoU@m⟩ is calculated as:

$$\text{dR@}n\text{,IoU@}m = \frac{1}{N_q}\sum_{i=1}^{N_q} r(n, m, q_i) \cdot \alpha_i^s \cdot \alpha_i^e \tag{15}$$

where discounted ratio $\alpha_i^* = 1 - |p_i^* - g_i^*|, * \in \{s, e\}$. $|p_i^* - g_i^*|$ is the absolute distance between the boundaries of the predicted and the ground truth moments (see Fig. 8(b)). Note that both p_i^* and g_i^* are normalized in 0.0 to 1.0 by dividing the corresponding whole video length. If the predicted moment exactly matches ground truth, then the discounted ratio $\alpha_i^* = 1$, and the metric degrades to ⟨R@n, IoU@m⟩. Otherwise, even if IoU threshold is met, $r(n, m, q_i)$ is discounted by α_i^*, which helps to restrain overlong predictions.

In Fig. 8(b), $(p_{i,1}^s, p_{i,1}^e)$ and $(p_{i,2}^s, p_{i,2}^e)$ are two example predicted moments of query q_i, and (g_i^s, g_i^e) is ground truth moment. Suppose both Predictions 1 and 2 in Fig. 8(b) have the same IoU value which satisfies IoU $\geq m$, ($m \leq 0.5$ here), ⟨dR@n, IoU@m⟩ penalizes more on Prediction 2 since its temporal boundaries are farther from ground truth. With respect to ⟨R@n,IoU@m⟩ and ⟨dR@n,IoU@m⟩ metrics, community is habituated to set $n \in \{1, 5, 10\}$ and $m \in \{0.3, 0.5, 0.7\}$.

4 TSGV Methods

The majority of solutions proposed for TSGV belong to the supervised learning paradigm. Early solutions mainly rely on sliding windows or segment proposal networks to pre-sample proposal candidates from the input video. Then, the proposals are paired with the query to generate the best answers through cross-modal matching. However, this two-stage "*propose-and-rank*" pipeline is inefficient, because densely sampling candidates with overlap are essential to achieve high accuracy, leading to redundant computation and low efficiency. Meanwhile, the pairwise proposal-query matching may also neglect the contextual information. To overcome these drawbacks, alternative solutions like anchor-based and proposal-free methods are developed to address TSGV in an "*end-to-end*" manner. These methods encode the entire video sequence and all video information is maintained in the model, gradually becoming the predominant solution for TSGV. Fig. 9 depicts a chronological overview of the development of supervised learning for TSGV.

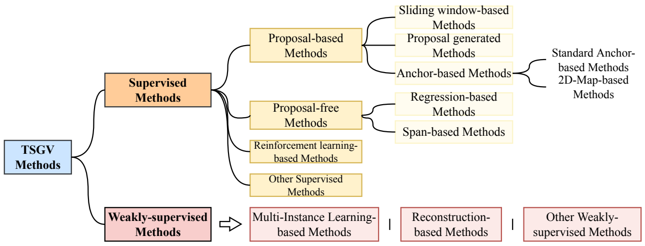

TSGV Methods
Supervised Methods
Proposal-based Methods
Sliding window-based Methods
Proposal generated Methods
Anchor-based Methods
Standard Anchor-based Methods
2D-Map-based Methods
Proposal-free Methods
Regression-based Methods
Span-based Methods
Reinforcement learning-based Methods
Other Supervised Methods
Weakly-supervised Methods
Multi-Instance Learning-based Methods
Reconstruction-based Methods
Other Weakly-supervised Methods

Fig. 10. A taxonomy of methods for TSGV.

Supervised learning requires a large number of annotated samples to train a TSGV method. Considering the difficulty and cost of data annotation, recent studies attempt to solve TSGV with weakly-supervised learning. These methods relieve the annotation burden by learning from video-query pairs without the detailed annotation of the temporal locations of events in videos.

Accordingly, the simple classification of proposal-based and proposal-free methods in Section 2 is incapable of covering all TSGV methods. Based on the method architecture and learning algorithm, we propose a new taxonomy in Fig. 10 to categorize TSGV methods. Next, we review the solutions to TSGV following this taxonomy and discuss the characteristics of each method category. Because the majority are supervised learning solutions, this section is organized mainly based on the categories under supervised learning.

4.1 Proposal-based Method

Depending on the ways to generate proposal candidates, proposal-based methods can be grouped into three categories, *i.e.,* sliding window-based, proposal generated, and anchor-based methods. Sliding window-based and some of the proposal-generated methods follow a two-stage propose-and-rank pipeline, where the generation of proposal candidates is separated from the model computation. Anchor-based methods incorporate proposal generation in model computation to achieve end-to-end learning.

4.1.1 Sliding Window-based Method

The sliding window-based method adopts multi-scale sliding windows (SW) to generate proposal candidates (ref. Fig. 5(a)). Then the multimodal matching module finds the best matching proposal

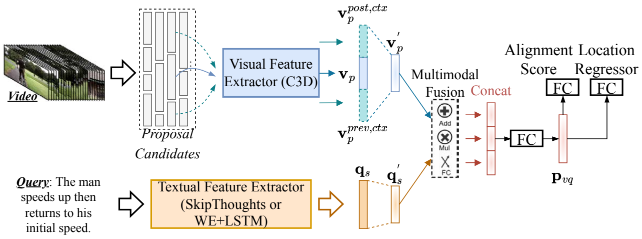

Video
Proposal Candidates
Visual Feature Extractor (C3D)
$\mathbf{v}_p^{post,ctx}$
$\mathbf{v}_p$
$\mathbf{v}_p^{prev,ctx}$
$\mathbf{v}_p^{'}$
Multimodal Fusion
Add
Mul
FC
Concat
FC
Alignment Score
Location Regressor
FC
FC
$\mathbf{p}_{vq}$
Query: The man speeds up then returns to his initial speed.
Textual Feature Extractor (SkipThoughts or WE+LSTM)
$\mathbf{q}_s$
$\mathbf{q}_s^{'}$

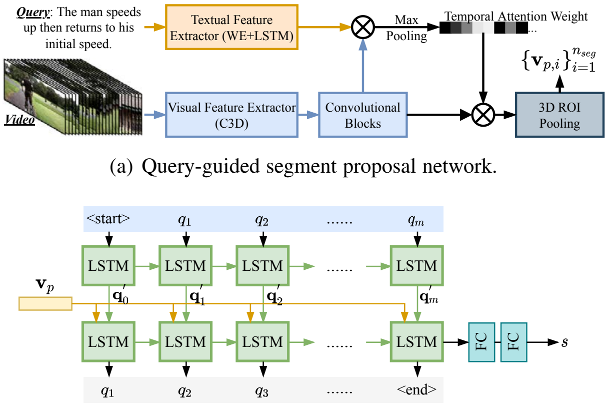
Fig. 11. CTRL architecture, reproduced from Gao *et al.* [9].

for a query. CTRL [9] and MCN [10] are two canonical SW methods, which are also pioneering work in TSGV. They define the task and construct corresponding benchmark datasets.

CTRL first produces proposals of various lengths through sliding windows, then encodes these proposals by a visual encoder, shown in Fig. 11. The query is converted to sentence representation via a textual encoder. For cross-modal reasoning, it builds a relatively simple multimodal processing module with three operators, *i.e.,* add, multiply, and fully connected (FC) layer, to fuse visual and textual features. CTRL designs multi-task objectives by using both an alignment predictor and a regressor. The alignment predictor computes the matching score between the proposal and query (ref. Eqn. 7). However, for an aligned proposal-query pair, the position of the proposal may not match the ground truth moment exactly. The regressor uses the smoothed L_1 loss to compute the corresponding offsets (ref. Eqn. 10) to better align the proposal.

Different from CTRL, MCN aims to project both proposal and query features to a common embedding space. Then, it encourages the distance between the query and the aligned proposal to be smaller than that of negative proposals. Specifically, MCN minimizes the squared distance between the query and proposals to supervise model learning. Negative proposals can be misaligned proposals within the same video (intra-video), or proposals from other videos (inter-video). Thus, MCN builds both intra- and inter-triple-based ranking losses (ref. Eqn. 8) as objectives. The intra-loss differentiates subtle differences within a video, and the inter-loss differentiates broad semantic concepts. Based on MCN, Hendricks *et al.* [45] further proposes MLLC, which treats the video context as a latent variable and unifies MCN and CTRL for moment localization.

The prior methods encode the entire query into one feature vector and apply simple cross-modal reasoning for feature fusion. However, treating queries holistically may obfuscate the keywords that have rich temporal and semantic cues. The simple fusion strategy also leads to inferior cross-modal understanding. Temporal dependencies and reasoning between video events and texts are not fully considered. Spatial-temporal information inside the video or query is also overlooked. A number of methods are proposed to address these issues. Among them, ROLE [11], MCF [46], ACRN [12], TCMN [47], and ASST [49] mainly focus on refining the multimodal interaction/fusion between visual and textual features, through more sophisticated structures or semantic decomposition of video/query. ACL [13], built upon CTRL, explicitly mines activity concepts from video and language as prior knowledge, to calibrate the confidence of the proposal to be the target moment. In addition to multimodal interaction refinement, SLTA [48] and MMRG [50] also exploit to incorporate appearance knowledge, *i.e.,* object-level spatial visual features, to enhance cross-modal reasoning as an additional view of video content. Instead of generating proposals at the initial stage, Ning *et al.* [51] equip SW strategy inside their model enabling end-to-end training. CAMG [127] designs a context-aware moment graph method, which utilizes semantic and temporal moment graphs to refine the proposals with semantic and position information.



(a) Query-guided segment proposal network.

(b) The early fusion retrieval model of QSPN.

Fig. 12. QSPN architecture, reproduced from Xu *et al.* [53].

In general, early SW-based methods have simple architectures. These methods lack both in-depth analyses of semantic knowledge of modalities and fine-grained multimodal fusion mechanisms, leading to inferior performance. The following work attempts to address these weaknesses by devising various techniques to better exploit video content and query, enhancing cross-modal reasoning between them. Despite continuous improvements, the two-stage sliding window-based methods suffer from inevitable drawbacks. Specifically, densely sampling proposals with multi-scale sliding windows result in heavily computational costs, as many overlapped areas are re-computed. These methods are also sensitive to negative samples, where fallacious negative samples may lead to inferior results.

4.1.2 *Proposal Generated Method*

The proposal generated (PG) method alleviates the computation burden of SW-based methods by avoiding the dense sampling process. Instead, PG methods generate proposals conditioned on the query. The number of proposals hence reduces remarkably.

Early proposal-generated methods still follow the two-stage propose-and-rank pipeline. Xu *et al.* [14] employ a pre-trained segment proposal network (SPN) [52] for proposal candidate generation, rather than adopting sliding windows. Based on Xu *et al.* [14], QSPN [53] further ameliorates SPN to produce query-specific proposal candidates. As illustrated in Fig. 12(a), QSPN interacts query embedding with visual features to derive temporal attention weights and re-weights the visual features to refine proposal generation. With the generated proposal feature, QSPN sequentially encodes the proposal with each token in the query and predicts the similarity score, at last, shown in Fig. 12(b). QSPN is optimized by triple-based ranking loss (ref. Eqn. 8), while a captioning loss is adopted to improve performance via query re-generation. Similarly, SAP [54] directly trains a visual concept detector to generate proposal candidates by measuring visual-semantic correlations between query and video frames.

Although the two-stage PG methods mitigate computation complexity to some degree, they still encounter some ineluctable

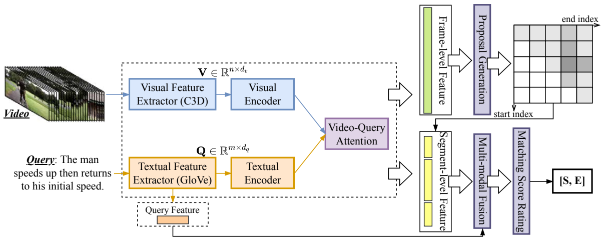

Video
Query: The man speeds up then returns to his initial speed.
$\mathbf{V} \in \mathbb{R}^{n\times d_v}$
Visual Feature Extractor (C3D)
Visual Encoder
$\mathbf{Q} \in \mathbb{R}^{m\times d_q}$
Textual Feature Extractor (GloVe)
Textual Encoder
Video-Query Attention
Query Feature
Frame-level Feature
Proposal Generation
end index
start index
Segment-level Feature
Multi-modal Fusion
Matching Score Rating
[S, E]

Fig. 13. BPNet architecture, reproduced from Xiao *et al.* [55].

drawbacks. To achieve good performance, PG methods still need to sample proposal candidates relatively densely, to increase the chance that at least one proposal can cover or is close to the ground truth moment. Similar to SW-based methods, the two-stage PG methods also rely on ranking-based objectives, making them sensitive to negative samples. Besides, proposal candidates are processed separately; hence, individual pairwise proposal-query matching may neglect the contextual information.

To overcome these defects, recent solutions [55]–[57], [128], [129] reformulate the pipeline of PG methods to a single-pass pattern in an end-to-end manner. Specifically, BPNet [55] (see Fig. 13) and APGN [56] replace the separate proposal generator by a proposal-free module (detailed in Section 4.2) and jointly train it with the main model. In this case, the proposal generation module is supervised by the ground truth moment, and only a few proposals are required to be generated. Besides, since the whole video is encoded as a feature sequence (ref. Section 2.3), visual features are jointly learned and interacted with the query. Thus, the model is able to consider contextual information. LPNet [57] maintains a boundary-aware predictor and learnable proposal module in parallel, where the boundary-aware predictor could refine predictions of the learnable proposal module. CMHN [58] generates proposal candidates with 1D regular convolution and models proposal-query matching in Hamming space through cross-modal hashing. Similar to BPNet, Gao *et al.* [128] also adopt a proposal-free module for candidate generation followed by a candidate refinement. Furthermore, SLP [129] proposes to first select the best-matched frame conditioned on the query. Then it constructs an initial segment based on the frame and dynamically updates it by exploring the adjacent frames with similar semantics.

4.1.3 Anchor-based Method

Sliding window and the early proposal-generated methods follow the two-stage propose-and-rank pipeline which suffers from various drawbacks. Researchers then source for alternative structures without pre-cutting proposal candidates at the input stage. One kind of solution is anchor-based methods, which incorporate proposal generation into answer prediction and maintain the proposals with various learning modules. According to how the anchors are produced and maintained, we further classify them into standard anchor-based and 2D-Map methods.

Standard Anchor-based Method. Methods in this category produce proposal candidates with multi-scale anchors and maintain them sequentially or hierarchically in the model. They aggregate contextual multimodal information and generate the final grounding result in one pass. The first anchor-based method for TSGV is Temporal GroundNet (TGN) by Chen *et al.* [15], shown in Fig. 14. TGN temporally captures the evolving fine-grained frame-by-word interactions between video and query. At each time step, multi-scale proposal candidates ending at the current time are generated using pre-set anchors. Then a sequential LSTM grounder simultaneously scores the group of proposals. TGN adopts weighted binary cross-entropy loss (ref. Eqn. 9) to optimize the model. In contrast, MAN [16] and SCDM [17], [60] adopt temporal convolutional networks to produce proposal candidates hierarchically. That is, proposals with different scales are generated at different levels of the stacked temporal convolution module. SCDM also adopts different multi-scale anchors compared to the standard version. Specifically, it imposes different scale anchors based on a basic span centered at each time step.

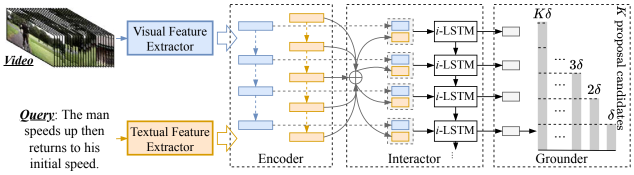


Fig. 14. TGN architecture, reproduced from Chen *et al.* [15].

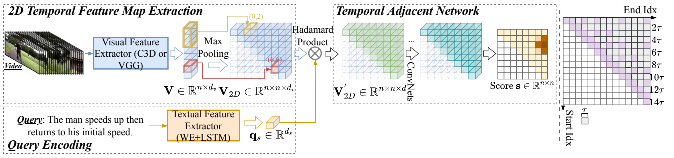

2D Temporal Feature Map Extraction
Video
Visual Feature Extractor (C3D or VGG)
Max Pooling
(0,2)
(6,6)
$\mathbf{V} \in \mathbb{R}^{n\times d_v}$
$\mathbf{V}_{2D} \in \mathbb{R}^{n\times n\times d_v}$
Hadamard Product
Query: The man speeds up then returns to his initial speed.
Textual Feature Extractor (WE+LSTM)
$\mathbf{q}_s \in \mathbb{R}^{d_s}$
Query Encoding
Temporal Adjacent Network
$\mathbf{V}'_{2D} \in \mathbb{R}^{n\times n\times d}$
ConvNets
Score $\mathbf{s} \in \mathbb{R}^{n\times n}$
End Idx
Start Idx
2τ
4τ
6τ
8τ
10τ
12τ
14τ
τ

Fig. 15. 2D-TAN architecture, reproduced from Zhang *et al.* [18].

Subsequent work generally follows the strategies of TGN or SCDM with more sophisticated learning modules and/or auxiliary objectives. To be specific, CMIN [59], [61], CBP [62], FIAN [63], HDRR [66], and MIGCN [67] adopt the strategy of TGN, while CSMGAN [64], RMN [65], IA-Net [68], and DCT-Net [69] apply the strategy of SCDM. These solutions design various cross-modal reasoning strategies to perform a more fine-grained and deeper multimodal interaction between video and query for precise moment localization. In addition, CBP [62] introduces an auxiliary boundary module to compute the confidence of the feature at each time step to be the boundary of the target moment. Some works adopt boundary regression modules to refine the generated moments' start and end time points. MIGCN [67] develops a rank module apart from the boundary regression module to distinguish the optimal proposal from a set of similar proposal candidates. ECCL [130] designs a sliding convolution locator to iteratively predict the best proposal candidates. MA3SRN [131] incorporates optical-flow-guided motion-aware, detection-based appearance-aware, and 3D-aware object features to interact with the query for better grounding.

2D-Map Anchor-based Method. Standard anchor-based method produces proposal candidates with preset multi-scale anchors and maintains them sequentially or hierarchically. These proposals are individually processed and their temporal dependencies are not well considered. Furthermore, the lengths of proposals are restricted by preset anchors. 2D-Map methods use a two-dimensional map to model the temporal relations between proposal candidates, shown in Fig. 5(d). Theoretically, 2D-Map could enumerate all possible proposals at any length, while maintaining their adjacent relations.

Before 2D-Map methods, a prior work TMN [70] first proposes to enumerate all possible consecutive segments as proposals and predict the best-matched proposal as the result through interacting each proposal with the query. However, TMN generates proposals in the answer predictor; its enumeration strategy is more like a variant of a standard anchor-based strategy.

2D-TAN [18] is the first solution modeling proposal with a 2D temporal map, and its overall architecture is shown in Fig. 15 left. 2D-TAN first extracts visual features and converts them into a 2D feature map, while the query is encoded in sentence-level representation. A temporal adjacent network is proposed to fuse the query feature with each proposal feature and embed the video context information with a convolutional network. As shown in Fig. 15 right, 2D-TAN divides the video into evenly spaced video snippets with duration τ, where (i, j) on the 2D map denotes a proposal candidate from time $i\tau$ to $j\tau$[6]. Instead of enumerating all possible consecutive segments as proposals, 2D-MAN proposes a sparse sampling strategy to remove redundant moments which have large overlaps with the selected proposals. The model adopts binary cross-entropy loss for model learning. 2D-TAN is further extended [71] with multi-scale modeling to achieve a larger receptive field and obtain richer contexts. The extended version reduces the complexity of proposal generation from quadratic to linear, making dense video prediction more efficient.

Due to its effectiveness, a series of work follows 2D-TAN's proposal generation[7] or its overall structure. As illustrated in Fig. 15, 2D-TAN directly encodes the query into the sentence-level feature and interacts with proposals via a simple Hadamard product. In this sense, multimodal interaction is overlooked. To remedy, PLN [72], SMIN [73], CLEAR [74], and STCM-Net [82] disentangle video proposals into different temporal granularities [72], [82] or different semantic contents [73], [74], and perform cross-modal reasoning at both coarse- and fine-grained granularities. VLG-Net [75] and RaNet [76] maintain query words and video proposals in a graph and adopt GCN [4], [78] to conduct both intra- and inter-modal interactions. SV-VMR [80] decomposes the query into semantic roles [132] and performs multi-level cross-modal reasoning at the semantic level. MATN [77] further concatenates proposals and query words into a sequence and encodes them through a single-stream transformer network. It also devises a novel multi-stage boundary regression to refine the predicted moments. Instead of using the simple Hadamard product, DMN [81] proposes to project proposals and query features to a common embedding space and leverage metric learning for cross-modal pair discrimination. FVMR [79] and CCA [133] devise joint semantic embedding module for multimodal interaction to facilitate the cross-modal reasoning. Guo *et al.* [134] introduce the Wasserstein distance [135] to match video-text domain. DCLN [136] and TACI [137] decompose 2D-Map into start and end channels for cross-modal reasoning and then fuses to 2D-Map features to facilitate model training. Xu *et al.* [138] propose a contrastive language-action pre-training framework for TSGV. Bao *et al.* [139] address the bias issue via a sample-reweighting-based debiased temporal localizer. Moreover, a series of recent work [140]–[144] focuses on developing multi-stage cross-modal fusion module in hierarchy, sequence, or multi-granularity, for better moment prediction.

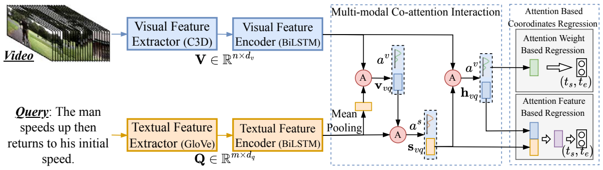

Video
Visual Feature Extractor (C3D)
Visual Feature Encoder (BiLSTM)
$\mathbf{V} \in \mathbb{R}^{n \times d_v}$
Query: The man speeds up then returns to his initial speed.
Textual Feature Extractor (GloVe)
Textual Feature Encoder (BiLSTM)
$\mathbf{Q} \in \mathbb{R}^{m \times d_q}$
Multi-modal Co-attention Interaction
Mean Pooling
A
a^v
$\mathbf{v}_{vq}$
a^s
$\mathbf{s}_{vq}$
$\mathbf{h}_{vq}$
Attention Based Coorodinates Regression
Attention Weight Based Regression
Attention Feature Based Regression
(t_s, t_e)

Fig. 16. ABLR architecture, reproduced from Yuan *et al.* [19].

4.2 Proposal-free Method

Proposal-based methods perform various proposal generations and essentially depend on the ranking of proposal candidates. In contrast, proposal-free methods directly predict the start and end boundaries of the target moment on fine-grained video snippet sequence, without ranking a vast of proposal candidates. Depending on the format of moment boundaries, proposal-free methods are categorized into regression-based and span-based methods.

4.2.1 *Regression-based Method*

Regression-based method computes a time pair (t_s, t_e) and compares the computed pair with ground truth (τ_s, τ_e) for model optimization. Attention-based location regression (ABLR) [19] is one of the first regression-based solutions for TSGV. Depicted in Fig. 16, ABLR extracts visual and textual features and encodes them through BiLSTM networks to aggregate contextual information, respectively. Then, a three-stage multimodal co-attention is developed to perform cross-modal reasoning. The multimodal feature is fed to the regressor for moment prediction. ABLR explores two types of regressors. One is attention weight-based regression, which takes video attention weights as input. Another is attended feature-based regression, which fuses the attended visual and textual features as inputs. The model is optimized by the smoothed L_1 loss. ABLR also devises an attention calibration loss to refine video attention, which encourages higher attention weights to video snippets within the ground truth moment.

Concurrently, ExCL [20] also addresses TSGV by regression and designs three different answer predictors following ideas from reading comprehension in NLP [6]–[8]. Similar to proposal-based methods, subsequent regression work [22], [88]–[92], [95], [96], [145]–[150] dives in designing various feature encoding and cross-modal reasoning strategies for superior multimodal interaction and accurate moment localization. From the perspective of regression, DEBUG [22], GDP [95], and DRN [89] analyze the data imbalance issue in TSGV: the number of video frames is large, but the positive samples are sparse *i.e.,* only two frames for start and end timestamps. They regard all frames within the ground truth moment as positive and densely predict the distances to the boundaries for each frame within the ground truth moment to mitigate the sparsity issue. CMA [88] and DeNet [92] study bias issue in TSGV. Specifically, CMA [88] rectifies the inevitable annotation bias by moment boundary ambiguities via a two-branch cross-modality attention network and a task-specific regression loss. VISA [145] adopts a variational cross-graph correspondence learning with regression head, to study the generalization ability of the model to queries with novel compositions of seen words. HLGT [146] and MGSL-Net [148] deeply mine the transformer variants [151] for TSGV. HiSA [147] introduces contrastive learning to model intra-video entanglement and inter-video connection as auxiliary objectives. PLPNet [149] further decomposes the query into phrases and localizes each phrase jointly as an ancillary.

[6] $i \leq j$, *i.e.,* only the upper triangular area of 2D map is valid

[7] Some methods follow 2D-TAN's proposal generation to produce proposal candidates, but they may not maintain the proposals in a 2D map.

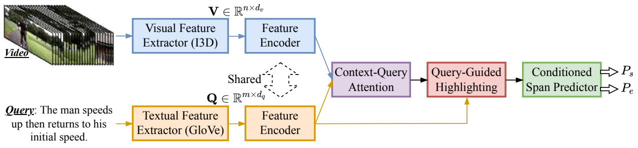

$\mathbf{V} \in \mathbb{R}^{n \times d_v}$
Video
Visual Feature Extractor (I3D)
Feature Encoder
Shared
$\mathbf{Q} \in \mathbb{R}^{m \times d_q}$
Query: The man speeds up then returns to his initial speed.
Textual Feature Extractor (GloVe)
Feature Encoder
Context-Query Attention
Query-Guided Highlighting
Conditioned Span Predictor
P_s
P_e

Fig. 17. VSLNet architecture, reproduced from Zhang *et al.* [23].

DeNet [92] disentangles query into relations and modified features and devises a debias mechanism to alleviate both query uncertainty and annotation bias issues.

There are also regression methods [93], [98], [99], [152] incorporating additional modalities from video to improve the localization performance. For instance, HVTG [98] and MARN [152] extract both appearance and motion features from video. In addition to appearance and motion, PMI [99] further exploits audio features from the video extracted by SoundNet [153]. DRFT [93] leverages the visual, optical flow, and depth flow features of video, and analyzes the retrieval results of different feature view combinations.

4.2.2 *Span-based Method*

Span-based methods aim to predict the probability of each video snippet/frame being the start and end positions of the target moment. Inspired by the reading comprehension (RC) task in NLP [6]–[8], L-Net [21] and ExCL [20] first formulate TSGV as a span prediction task. In addition to the regression-based predictors, ExCL also designs corresponding span prediction heads.

Based on these two works, Zhang *et al.* [23] compare differences between RC and TSGV tasks and propose VSLNet. Specifically, video is continuous and causal relations between video events are usually adjacent, while words in query are discrete and demonstrate syntactic structure. Shown in Fig. 17, VSLNet exploits a context-query attention modified from QANet [7] to perform fine-grained multimodal interaction. Then a conditioned span predictor computes the probabilities of the start/end boundaries of the target moment. To bridge the gap between RC and TSGV, VSLNet introduces a query-guided highlighting module, which effectively narrows down the moment search space to a smaller highlighted region. Existing methods including VSLNet generally perform better on short videos than on long videos. Their follow-up work [100] extends VSLNet to handle long videos by incorporating concepts from multi-paragraph question answering [154]. Long videos are split into multiple short videos and a hierarchical searching strategy is deployed for moment localization.

In general, the overall frameworks of regression- and span-based methods are similar. Thus, the continuous performance improvements of subsequent work [84], [101]–[108], [110]–[113], [155]–[167] are also achieved by modifying the feature encoding and multimodal interaction modules, introducing auxiliary objectives, and/or augmenting additional features. In particular, SeqPAN [101] introduces the concept of named entity recognition [168]–[170] in NLP by splitting the snippet sequence into begin, inside, and end regions of the target moment, and background region. IVG-DCL [102] introduces a dual contrastive learning mechanism to enhance multimodal interaction and leverages a structured causal model [171] to address the selection bias of TSGV. CI-MHA [103] proposes to remedy the start/end prediction noise caused by annotator disagreement via an auxiliary moment segmentation task. ABIN [106] devises an auxiliary adversarial discriminator network to produce coordinate and frame correlation distributions for moment boundary refinement. DORi [113] incorporates appearance features and captures the relations between objects and actions guided by the query. CBLN [84] addresses TSGV from a new perspective. It reformulates TSGV by scoring all pairs of start and end indices simultaneously and predicting moments with a biaffine structure. Hao *et al.* [155] and Liu *et al.* [164] focus on solving the bias issue via video shuffling and contrastive sample generation, respectively. PPT [162] and VPTSL [166] introduce prompts to jointly model video and text with the unified framework. LocFormer [159] designs a multimodal transformer for TSGV in BERT-style. CFSTRI [156], MS2P [158] and STDNet [165] adopt spatio-temporal cues to interact with query for TSGV. Yang *et al.* [157] and Hao [163] decompose the query into multiple semantic phrases to interact with video for boundary prediction. MCA [160] and PEARL [161] further incorporate subtitles of the video to assist the TSGV. EMB [167] solves uncertainties in TSGV with an elastic moment bounding strategy.

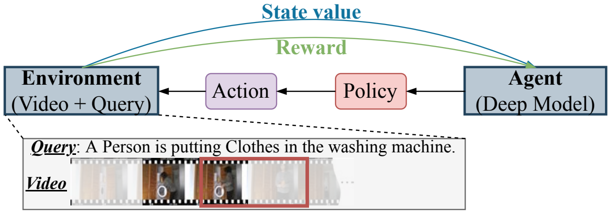


Fig. 18. Illustration of sequence decision making formulation in TSGV.

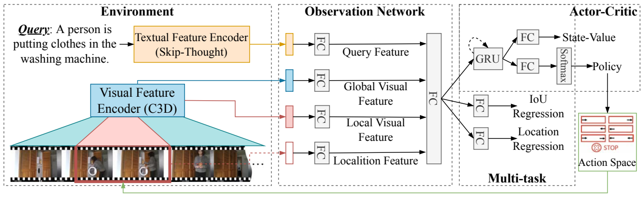


Fig. 19. RWM-RL architecture, reproduced from He *et al.* [24].

4.3 Reinforcement Leaning-based Method

From the perspective of proposal usage, reinforcement learning (RL) based methods are also proposal-free methods. However, their task formulation is fundamentally different from the proposal-free methods reviewed earlier. RL-based method formulates TSGV as a sequence decision-making problem and utilizes deep reinforcement learning techniques to solve it.

Illustrated in Fig. 18, the RL-based method usually maintains a sliding window (the dark red rectangle). The sliding window here is different from that discussed in Section 4.1. The RL-based method only adopts a single window, controlled by an agent. An agent, *i.e.,* a learnable module, controls the window movement based on a set of handcraft-designed temporal transformations *e.g.,* shifting, and scaling. At each learning step, after each movement, a reward is generated to indicate whether the window is closer or farther away from the target moment. The agent will adapt its action for the next step within pre-defined action space.

RWM-RL [24] is one of the first works to define and solve TSGV with an RL framework. Shown in Fig. 19, it consists

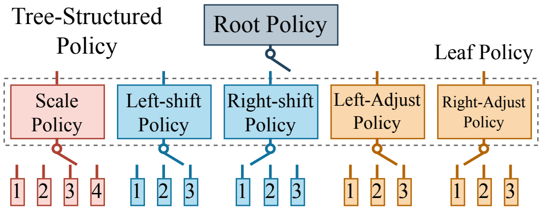


Fig. 20. Tree-structured policy, reproduced from Wu *et al.* [26].

of three modules. The environment module converts the query, global video, and local video segment within the window into corresponding representations. Then the observation network fuses query and video features to output the current state of environment, *i.e.,* multimodal representation, at each learning step. In the decision-making module (*i.e.,* agent), RWM-RL leverages the actor-critic algorithm [172] to compute the state-value and an action policy, *i.e.,* the probabilistic distribution of all pre-designed actions in the action space. The state-value is used for reward computation, and the action policy determines the movement of the sliding window to adjust the temporal boundaries. RWM-RL defines 7 actions: moving start/end point ahead/backward by δ (4 scaling actions), shifting both start and end point backward/forward by δ (2 shifting actions), and a STOP action, where δ denote a basic moving distance. In general, the iterative process ends when encountering the STOP action or reaching the preset maximum number of iteration steps. RWM-RL adopts GRU to model the sequential decision-making process for the actor-critic module. A reward is computed at each step, where the reward is designed to encourage the agent to find a better matching position step by step. All rewards are accumulated for model optimization by utilizing the advantage function [172] as objective and Monte Carlo sampling [173] for policy gradient approximation. To increase the action diversity, RWM-RL further introduces entropy of the policy output as an auxiliary objective following A2-RL [174].

SM-RL [25] presents an RNN-based semantic matching RL model to selectively observe proposal candidates produced by a controllable agent. TSP-PRL [26] designs a hierarchical action space with a tree-structured policy, inspired by human's coarse-to-fine decision-making mechanism. The action selection is controlled by a switch over an interface in a tree-structured policy (see Fig. 20). AVMR [175] treats the RL-based module as a generator and devises a Bayesian ranking module as a discriminator to rank proposals. Based on AVMR, Zeng *et al.* [176] further deploy continual multi-task learning as the discriminator, which jointly optimizes the ranking and localization subtasks, to boost the performance. STRONG [177] considers appearance and motion features and employs parallel spatial-level and temporal-level RL modules for moment localization. TripNet [178] mainly focuses on ameliorating the observation network to boost performance. Instead of using sliding windows, MABAN [179] leverages two individual agents to model start and end points separately. The two agents are conditioned on each other to avoid invalid predictions.

4.4 Other Supervised Method

In addition to the categories mentioned above, researchers also explore other types of formulations to address TSGV, or under different settings. Shao *et al.* [83] design a unified framework based on TAG [180] to perform both video-level retrieval and moment-level localization simultaneously. The two tasks could reinforce each other. Similarly, Jiang *et al.* [181] design a cross-task sample transfer to jointly solve video summarization and moment localization. DPIN [85] devises a dual-path interaction network to integrate the benefits of both proposal-based and proposal-free methods. Inspired by Patrick *et al.* [182], Ding *et al.* [87] propose a support-set based cross-supervision strategy to enhance multimodal interaction, through discriminative contrastive and generative caption objectives. Since multiple moments in a video are semantically correlated and temporally coordinated based on their order, several studies [86], [183]–[185] explore a novel setting of TSGV, named dense events grounding, which allows jointly localizing multiple moments described in a paragraph, *i.e.,* multiple sentences. SNEAK [109] studies the adversarial robustness of TSGV models by examining three facets of vulnerabilities, *i.e.,* vision, language, and cross-modal interaction, from both attack and defense aspects. Yang *et al.* [186] first explore neural architecture search for TSGV. Xu *et al.* [97] and Cao *et al.* [187] further investigate model pre-training for TSGV, and Xu *et al.* [97] construct a large-scale synthesized dataset with annotations and design a boundary-sensitive pretext task. JVTF [188] proposes to solve TSGV by utilizing video question answers as a special variant. Zhang *et al.* [189] design a unified framework for VideoQA, TSGV, and VR with global and segment-level alignments. Cao *et al.* [94] reformulate TSGV as a set prediction task and propose a multimodal transformer model inherited from DETR [190], LVTR [191] and UMT [192] further improve the DETR-based TSGV framework to boost its performance.

4.5 Summary of Supervised Method

We have reviewed different categories of supervised TSGV methods, as well as their advantages and shortcomings. In general, early sliding window-based and proposal-generated methods suffer from low efficiency and flexibility, because of dense and overlapped proposals. These methods also rely on ranking-based loss, making them sensitive to negative samples. Anchor-based methods, another form of the proposal-based solution, learn TSGV in an end-to-end manner. The proposal generation process is incorporated into the model, abnegating the ineffective SW and PG strategies. Anchor-based methods also enable contextualized representation learning and fine-grained multimodal interaction. However, the anchor-based methods still need to maintain a mass of proposals during prediction, which hinders model efficiency.

Proposal-free methods directly learn to predict the boundaries of the target moment, without maintaining any proposals. These methods are more efficient and flexible to adapt to moments with diverse lengths. Nevertheless, compared to proposal-based methods, proposal-free methods overlook the rich information between start and end boundaries and fail to exploit the proposal-level interaction. They also suffer from severe imbalance issues between the positive and negative training samples, *i.e.,* only two (start and end) frames are positive in the whole video. Also belonging to a proposal-free category, the design of RL-based methods is intuitive and effective, kind of simulating human's decision-making strategy. However, their performance is unstable due to the difficulty of optimizing RL-based methods.

Despite a vast number of methods in each category, all methods focus on ameliorating cross-modal reasoning, to achieve fine-grained and precise multimodal interaction. Thus, the high-level pipeline of methods in each category is similar in general.

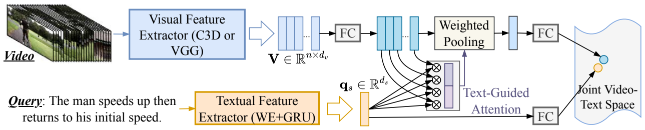

Video
Visual Feature Extractor (C3D or VGG)
$\mathbf{V} \in \mathbb{R}^{n \times d_v}$
FC
Weighted Pooling
FC
$\mathbf{q}_s \in \mathbb{R}^{d_s}$
Text-Guided Attention
FC
Joint Video-Text Space
Query: The man speeds up then returns to his initial speed.
Textual Feature Extractor (WE+GRU)

Fig. 21. TGA architecture, reproduced from Mithun *et al.* [27].

Recall that the feature interactor is responsible for understanding the semantic concepts of both query and video and fusing them to emphasize the video contents that are semantically relevant to the query. In this sense, the quality of the interactor module determines a TSGV model's performance to a great extent.

4.6 Weakly-supervised TSGV Method

Supervised learning usually needs a large number of annotations for model training. Annotating temporal boundaries on video with text description is extremely time-consuming and labor-intensive, often not scalable. Furthermore, annotations also suffer from the inaccurate issue, *i.e.,* action boundaries in videos are usually subjective and inconsistent across different annotators.

Under the weakly-supervised setting, TSGV methods only need video-query pairs but not the annotations of starting/end time. They explore to find results in a shared multimodal feature space or with a reconstruction-based strategy. In general, the existing weakly-supervised TSGV methods can be roughly grouped into multi-instance learning and reconstruction-based models.

4.6.1 *Multi-Instance Learning Method*

Multi-instance learning method generally regards the input video as a bag of instances with bag-level annotations. The prediction of instance, *i.e.,* proposals, is aggregated as the bag-level prediction.

TGA [27] first solves TSGV under the multi-instance learning setting. As shown in Fig. 21, TGA first encodes video and query features and presents text-guided attention to learn text-specific global video representations. Then both visual and textual features are projected to a joint space. TGA regards the video and its corresponding query descriptions as positive pairs, while the video with other queries and the query with other videos as negative pairs. TGA learns visual-text alignment at the video level by maximizing the matching scores of positive samples while minimizing the scores of negative samples.

To achieve good performance, MIL-based methods have to perform precise semantics alignment between video and query. Thus, subsequent solutions [28], [193]–[207] mainly focus on devising a sophisticated cross-modal alignment module, designing a robust proposal selection strategy, and/or building effective learning objectives. WSLLN [28] models alignment and detection modules in parallel to perform proposal selection and video-level alignment simultaneously. VLANet [194] designs a surrogate proposal selection module to prune irrelevant proposal candidates. Chen *et al.* [193] and Teng *et al.* [202] perform video-query alignment at multiple granularities. CCL [196], VCA [198] and MSCL [206] introduce contrastive learning mechanisms to effectively distinguish positive and negative (or counterfactual positive) proposals. BAR [195] involves an additional RL module to progressively refine the retrieved proposals. FSAN [199], WSTAN [203], and LoGAN [204] focus on mining video and query contents and their correlations. Then they design a fine-grained cross-modal alignment module for accurate moment localization. Da *et al.* [197] study the uncertain false-positive frame issue, *i.e.,* an object might appear sparsely across multiple frames and devise an AsyNCE loss to mitigate the issue by disentangling positive pairs from negative ones. CRM [200] uses a cross-sentence relation mining strategy to explicitly model cross-sentence relations in the paragraph and explore cross-moment relations in the video. LCNet [201] further deploys self-supervised cycle consistent loss to guide video-query matching. SAN [207] designs a multi-scale Siamese module to progressively reduce the semantic gap between the visual and textual modalities. Chen *et al.* [205] explore the inter-contrast between videos via composition and design a single-stream framework with multi-task learning.

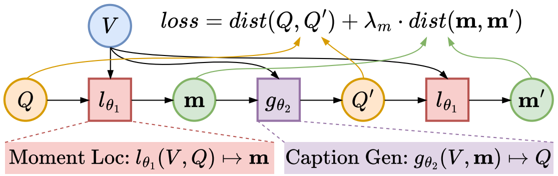

V
$loss = dist(Q, Q') + \lambda_m \cdot dist(\mathbf{m}, \mathbf{m}')$
Q
l_{θ_1}
m
g_{θ_2}
Q′
l_{θ_1}
m′
Moment Loc: $l_{\theta_1}(V, Q) \mapsto \mathbf{m}$
Caption Gen: $g_{\theta_2}(V, \mathbf{m}) \mapsto Q$

Fig. 22. WS-DEC architecture, reproduced from Duan *et al.* [29].

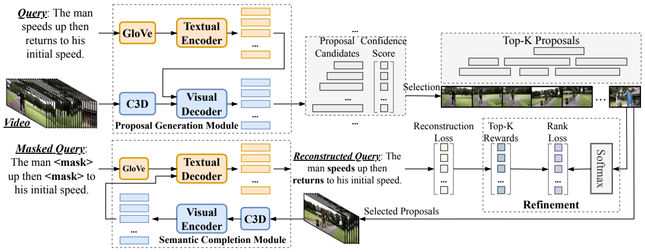

Query: The man speeds up then returns to his initial speed.
GloVe
Textual Encoder
Video
C3D
Visual Decoder
Proposal Generation Module
Proposal Candidates
Confidence Score
Top-K Proposals
Selection
Masked Query: The man <mask> up then <mask> to his initial speed.
GloVe
Textual Decoder
Reconstructed Query: The man speeds up then returns to his initial speed.
Reconstruction Loss
Top-K Rewards
Rank Loss
Softmax
Refinement
Visual Encoder
C3D
Semantic Completion Module
Selected Proposals

Fig. 23. SCN architecture, reproduced from Lin *et al.* [30].

4.6.2 *Reconstruction-based Method*

Reconstruction-based method tackles TSGV in an indirect way. Methods in this category first take video and query as inputs to produce desired proposals matched to the query. Then the proposals are used to reconstruct the query, where the intermediate proposals are treated as localization results.

The idea of reconstruction is first explored by Duan *et al.* [29]. They propose a method to solve weakly supervised dense event captioning (WS-DEC), where moment localization is an auxiliary sub-task to assist model training. The authors indicate that moment localization and event captioning are a pair of dual tasks. Moment localization is to learn a mapping $l_{\theta_1} : (V, Q) \mapsto \mathbf{m}$, *i.e.,* retrieving a moment $\mathbf{m}$ corresponded to the caption C_i from video V. Event captioning inversely generates caption Q for the given $\mathbf{m}$, *i.e.,* $g_{\theta_2} : (V, \mathbf{m}) \mapsto Q$. Since Q and $\mathbf{m}$ are a one-to-one correspondence, the dual problems exist simultaneously, and Q and $\mathbf{m}$ are tired together. By nesting the dual functions, caption-moment pair $(Q, \mathbf{m})$ becomes a fixed-point solution as:

$$Q = g_{\theta_2}(V, l_{\theta_1}(V, Q)), \quad \mathbf{m} = l_{\theta_1}(V, g_{\theta_2}(V, \mathbf{m})), \tag{16}$$

where l_{θ_1} and g_{θ_2} are the localization and captioning modules, respectively. As shown in Fig. 22, WS-DEC first retrieves moment $\mathbf{m}$ by giving video V and caption Q; Then the retrieved $\mathbf{m}$ and V are used to reconstruct the caption, denoted by Q'; Finally, the reconstructed Q' and V are utilized to relocate the moment $\mathbf{m}'$ again. The objective of WS-DEC is to minimize the distances of $\mathbf{m}$-$\mathbf{m}'$ and Q-Q' pairs simultaneously.

SCN [30] adopts a similar idea as WS-DEC. However, SCN is designed for solving weakly supervised TSGV directly; it does not use a specific caption generation module, but switches to reconstruct the masked query. As depicted in Fig. 23, SCN first retrieves a set of proposals from the video. The model then selects top-K proposals as input to reconstruct masked queries, and compute rewards based on reconstruction loss. The rewards further act as feedback to refine proposal generation. CMLNet [208] utilizes a similar structure as SCN and introduces a punishment loss in the candidate generation module. MARN [209] leverages both proposal-level and clip-level video features to produce more accurate proposal candidates. The proposal-level and clip-level features are generated by 2D-Map strategy [18] and BMN [210], respectively. EC-SL [31], [211] improves WS-DEC by introducing a concept learner and an induced set attention block. Both CPL [212] and CNM [213] introduce contrastive learning into their models. CPL devises a Gaussian-based contrastive proposal learning module, and CNM explores the contrastive negative sample mining strategy.

4.6.3 Other Weakly-supervised Method

In addition to MIL and reconstruction methods, Zhang *et al.* [214] consider both inter- and intra-sample confrontments to address the drawbacks of standard MIL-based methods. The latter generally ignores intra-sample confrontment between moments with semantically similar contents. Luo *et al.* [215] solve the TSGV task in a semi-supervised way. They construct a teacher-student network. The teacher module produces instant pseudo labels for unlabeled samples based on predictions. The student module learns from pseudo labels via self-supervised learning. SVPTR [185] explores dense events grounding via contrastive learning under the semi-supervised setting. Nam *et al.* [216] further propose to learn a TSGV model in zero-shot manner to eliminate the annotation cost. In the zero-shot setting, video-query pairs are not provided. They utilize an off-the-shelf object detector and pseudo-query generation module fine-tuned on RoBERTa [39] to produce proposals and queries, and simulate the standard TSGV learning. Gao *et al.* [217] also explores leveraging an off-the-shelf visual concept detector and a pre-trained image-sentence embedding space to perform TSGV without using text annotations on video. Liu *et al.* [218] design a deep semantic clustering network for unsupervised TSGV. Paul *et al.* [219] define the task of localizing novel moments for unseen queries to investigate the ability of TSGV models to novel events. PS-VTG [220] and ViGA [221] further explore to utilize single frame/point annotation for TSGV.

5 Performance Comparison

We now summarize the reported performance of TSGV methods over the years, by category. Due to the page limit, the detailed results are listed in supplementary materials.

Performance Overview. For supervised methods, as summarized in Table 2 and Table 3, in general, anchor-based (ANchor and 2D-map) and proposal-free (ReGression and SpaN) methods are superior to sliding window-based (SW) and proposal-generated (PG) methods. Within the SW category, MMRG [50] introduces a graph structure to model the visual-textual relations and adds a boundary regression auxiliary objective to guide moment retrieval, outperforming early SW methods by a large margin. A similar observation holds in the PG category. Compared to early anchor-based and proposal-free work, recent methods incorporate more sophisticated multimodal interaction strategies to refine the cross-modal reasoning between video and query. They also introduce various auxiliary objectives to enhance the feature representation learning and steer the model for more precise moment localization. In the RL category, recent solutions mainly focus on designing more powerful agents or refined action space (policy) to achieve accurate sequence decisions [26], [179]. Despite the improvements of recent RL-based methods, the performance gap between RL-based methods and anchor-based/proposal-free methods remains distinct. Two possible reasons for the inferior results are: (i) the RL learning process is not very stable, and (ii) the multimodal interaction between the two modalities is not fully exploited in RL methods. Among other methods, GTR [94] and BSP [97] provide a new perspective to solve TSGV. BSP proposes a pre-training paradigm for TSGV by designing a boundary-sensitive pretext task and collecting a synthesized dataset with temporal boundaries. GTR builds an end-to-end framework to learn TSGV from raw videos directly. Although their results are slightly inferior to other solutions, both open up new directions for TSGV.

For weakly-supervised methods (Table 4), in general, MIL-based methods are superior to reconstruction-based methods. Other than cross-modal reasoning, the learning objective also plays a key role in MIL-based methods. Recent solutions adopt more effective strategies or introduce auxiliary objectives, such as contrastive learning [196], pseudo supervision [28], [203], and boundary adjustment [193]. For other methods, PSVL [216] solves TSGV under the zero-shot setting, which assumes that the video-query pairs are inaccessible, *i.e.,* only the text corpora and video collection are given. Although the zero-shot setting is more challenging, arguably the setting is closer to real-world scenarios.

Impact of Features. Improvements in model performance may come from various sources. In particular, various visual (*e.g.,* VGG [43], ResNet [223] C3D [1], I3D [3]) and textual (*e.g.,* GloVe [37], BERT [38]) feature extractors have been utilized in different models. Among visual feature extractors, VGG [43] and ResNet [223] are pre-trained on image datasets, and they are more effective in extracting appearance features like objects and visual concepts from video frames. In contrast, C3D [1] and I3D [3] are pre-trained on video action recognition datasets, for extracting motion features like actions or activities, from video snippets or segments. In general, feature analysis conducted by different methods on the Charades-STA dataset shows that VGG<C3D<I3D, with respect to model performance (able 5). Because TSGV is mainly for activity retrieval, video-based feature extractors are more effective than their image-based counterparts. I3D contains a more sophisticated structure and is trained on larger datasets than C3D, leading to a more powerful representation ability.

Visual features are mostly extracted from RGB frames. Chen *et al.* [93] explore incorporating optical flow and depth map information in frames as complementary visual features. Optical flow focuses on large motion, and depth maps reflect the scene configuration when the action is related to objects recognizable by their shapes. As shown in Table 6, prominent improvements are obtained by adding more visual modality features on Charades-STA and ActivityNet Captions. A query may contain descriptions of both objects and actions. Thus, some methods [99], [113] exploit both appearance and motion features to represent a video. In addition to the motion features extracted by C3D model, Chen *et al.* [99] introduce appearance features by IRV2 [223] and audio features from video by SoundNet [153]. In general, as summarized

TABLE 2
R@1, IoU=m of supervised methods. SW: Sliding Window-based, PG: Proposal Generated, AN: standard Anchor-based, 2D: 2D-Map, RG: Regression-based, SN: Span-based, RL: Reinforcement Learning-based methods.

Category	Method	Venue	Charades-STA			ActivityNet Captions			TACoS$_{\text{org}}$			TACoS$_{\text{2DTAN}}$		
			m=0.3	m=0.5	m=0.7	m=0.3	m=0.5	m=0.7	m=0.3	m=0.5	m=0.7	m=0.3	m=0.5	m=0.7
SW	CTRL [9]	ICCV'17	-	23.63	8.89	-	-	-	18.32	13.30	-	-	-	-
	MCN [10]	ICCV'17	13.57	4.05	-	-	-	-	-	-	-	-	-	-
	MCF [46]	IJCAI'18	-	-	-	-	-	-	18.64	12.53	-	-	-	-
	ROLE [11]	ACM MM'18	25.26	12.12	-	-	-	-	-	-	-	-	-	-
	ACRN [12]	SIGIR'18	-	-	-	-	-	-	19.52	14.62	-	-	-	
	SLTA [48]	ICMR'19	38.96	22.81	8.25	-	-	-	17.07	11.92	-	-	-	-
	ACL-K [13]	WACV'19	30.48	12.20	-	-	-	-	24.17	20.01	-	-	-	-
	ASST [49]	TMM'20	-	37.04	18.04	-	-	-	-	-	-	-	-	-
	MMRG [50]	CVPR'21	71.60	44.25	-	-	-	-	-	-	-	57.83	39.28	-
	I2N [51]	TIP'21	-	56.61	34.14	-	-	-	-	-	-	31.47	29.25	-
	CAMG [127]	ArXiv'22	62.10	48.33	26.53	64.58	46.68	26.64	-	-	-	-	-	-
PG	QSPN [53]	AAAI'19	54.70	35.60	15.80	45.30	27.70	13.60	-	-	-	-	-	-
	SAP [54]	AAAI'19	-	27.42	13.36	-	-	-	-	18.24	-	-	-	-
	BPNet [55]	AAAI'21	65.48	50.75	31.64	58.98	42.07	24.69	25.96	20.96	14.08	-	-	-
	LPNet [57]	EMNLP'21	66.59	54.33	34.03	64.29	45.92	25.39	-	-	-	-	-	-
	APGN [56]	EMNLP'21	-	62.58	38.86	-	48.92	28.64	-	-	-	40.47	27.86	-
	CMHN [58]	TIP'21	-	-	-	62.49	43.47	24.02	30.04	25.58	18.44	-	-	-
	SLP [129]	ACM MM'22	-	64.35	40.43	-	52.89	32.04	-	-	-	42.73	32.58	-
AN	TGN [15]	EMNLP'18	-	-	-	45.51	28.47	-	21.77	18.90	-	-	-	-
	CMIN [59]	SIGIR'19	-	-	-	63.61	43.40	23.88	24.64	18.05	-	-	-	-
	MAN [16]	CVPR'19	-	46.53	22.72	-	-	-	-	-	-	-	-	-
	SCDM [17]	NeurIPS'19	-	54.44	33.43	54.80	36.75	19.86	26.11	21.17	-	-	-	-
	CBP [62]	AAAI'20	-	36.80	18.87	54.30	35.76	17.80	27.31	24.79	19.10	-	-	-
	FIAN [63]	ACM MM'20	-	58.55	37.72	64.10	47.90	29.81	33.87	28.58	-	-	-	-
	CSMGAN [64]	ACM MM'20	-	-	-	68.52	49.11	29.15	33.90	27.09	-	-	-	-
	RMN [65]	COLING'20	-	59.13	36.98	67.01	47.41	27.21	32.21	25.61	-	-	-	-
	IA-Net [68]	EMNLP'21	-	61.29	37.91	67.14	48.57	27.95	-	-	-	37.91	26.27	-
	MIGCN [67]	TIP'21	-	57.10	34.54	60.03	44.94	-	-	-	-	-	-	-
	DCT-Net [69]	TIVC'21	-	-	-	66.00	47.06	27.63	-	-	-	43.25	33.31	24.74
	MA3SRN [131]	ArXiv'22	-	68.98	47.79	-	53.72	32.30	-	-	-	49.41	39.11	-
2D	2D-TAN [18]	AAAI'20	-	39.81	23.31	59.45	44.51	27.38	-	-	-	37.29	25.32	-
	MATN [77]	CVPR'21	-	-	-	-	48.02	31.78	-	-	-	48.79	37.57	-
	SMIN [73]	CVPR'21	-	64.06	40.75	-	48.46	30.34	-	-	-	48.01	35.24	-
	PLN [72]	ArXiv'21	68.60	56.02	35.16	59.65	45.66	29.28	-	-	-	43.89	31.12	-
	RaNet [76]	EMNLP'21	-	60.40	39.65	-	45.59	28.67	-	-	-	43.34	33.54	-
	FVMR [79]	ICCV'21	-	55.01	33.74	60.63	45.00	26.85	-	-	-	41.48	29.12	-
	VLG-Net [75]	ICCV'21	-	-	-	-	46.32	29.82	-	-	-	45.46	34.19	-
	CLEAR [74]	TIP'21	-	-	-	59.96	45.33	28.05	-	-	-	42.18	30.27	15.54
	Sun *et al.* [144]	SIGIR'22	-	60.82	41.16	-	47.92	30.47	-	-	-	48.81	36.74	-
RG	ABLR [19]	AAAI'19	-	-	-	55.67	36.79	-	19.50	9.40	-	-	-	-
	ExCL [20]	NAACL'19	61.50	44.10	22.40	63.00	43.60	24.10	45.50	28.00	13.80	-	-	-
	DEBUG [22]	EMNLP'19	54.95	37.39	17.92	55.91	39.72	-	23.45	-	-	-	-	-
	GDP [95]	AAAI'20	54.54	39.47	18.49	56.17	39.27	-	24.14	-	-			
	DRN [89]	CVPR'20	-	53.09	31.75	-	45.45	24.36	-	23.17	-	-	-	-
	LGI [90]	CVPR'20	72.96	59.46	35.48	58.52	41.51	23.07	-	-	-	-	-	-
	CPNet [91]	AAAI'21	-	60.27	38.74	-	40.56	21.63	-	-	-	42.61	28.29	-
	HiSA [147]	TIP'22	74.84	61.10	39.70	64.58	45.36	27.68	-	-	-	53.31	42.14	29.32
SN	VSLNet [23]	ACL'20	70.46	54.19	35.22	63.16	43.22	26.16	29.61	24.27	20.03	-	-	-
	CPN [111]	CVPR'21	75.53	59.77	36.67	62.81	45.10	28.10	-	-	-	48.29	36.58	21.58
	CI-MHA [103]	SIGIR'21	69.87	54.68	35.27	61.49	43.97	25.13	-	-	-	-	-	-
	IVG-DCL [102]	CVPR'21	67.63	50.24	32.88	63.22	43.84	27.10	38.84	29.07	19.05	-	-	-
	SeqPAN [101]	ACL'21	73.84	60.86	41.34	61.65	45.50	28.37	31.72	27.19	21.65	48.64	39.64	28.07
	ACRM [105]	TMM'21	73.47	57.93	38.33	-	-	-	-	-	-	51.26	39.34	26.94
	ABDIN [106]	TMM'21	-	-	-	63.19	44.02	24.23	23.63	20.16	-	-	-	-
	VSLNet-L [100]	TPAMI'21	70.46	54.19	35.22	62.35	43.86	27.51	32.04	27.92	23.28	47.66	36.34	26.42
	PEARL [108]	WACV'22	71.90	53.50	35.40	-	-	-	-	-	-	42.94	32.07	18.37
RL	RWM-RL [24]	AAAI'19	-	36.70	-	-	36.90	-	-	-	-	-	-	-
	SM-RL [25]	CVPR'19	-	24.36	11.17	-	-	-	20.25	15.95	-	-	-	-
	TSP-PRL [26]	AAAI'20	-	45.45	24.75	56.02	38.82	-	-	-	-	-	-	-
	TripNet [178]	BMVC'20	54.64	38.29	16.07	48.42	32.19	13.93	23.95	19.17	9.52	-	-	-
	MBAN [179]	TIP'21	-	56.29	32.26	-	42.42	24.34	-	-	-	-	-	-
	URL [176]	TMCCA'22	77.88	55.69	-	-	76.88	50.11	-	-	-	73.26	50.53	-
Other	DPIN [85]	ACM MM'20	-	47.98	26.96	62.40	47.27	28.31	-	-	-	46.74	32.92	-
	DepNet [86]	AAAI'21	-	-	-	72.81	55.91	33.46	-	-	-	41.34	27.16	-
	GTR [94]	EMNLP'21	-	62.58	39.68	-	50.57	29.11	-	-	-	40.39	30.22	-
	BSP [97]	ICCV'21	68.76	53.63	29.27	-	-	-	-	-	-	-	-	-
	CMAS [186]	TIP'22	-	48.37	29.44	-	46.23	29.48	-	-	-	31.37	16.85	-

TABLE 3
Results of supervised methods on DiDeMo dataset.

Category	Method	Venue	R@1, IoU=m			mIoU
			0.5	0.7	1.0	
SW	MCN [10]	ICCV'17	-	-	28.10	41.08
	MLLC [45]	EMNLP'18	-	-	27.46	41.20
	ROLE [11]	ACM MM'18	29.40	15.68	-	-
	ACRN [12]	SIGIR'18	27.44	16.65	-	-
	SLTA [48]	ICMR'19	30.92	17.16	-	-
	TCMN [47]	ACM MM'19	-	-	28.90	41.03
	ASST [49]	TMM'20	-	-	32.38	47.49
	I2N [51]	TIP'21	-	-	29.00	44.32
PG	EFRC [14]	ArXiv'18	11.9	5.5	13.23	27.57
AN	TGN [15]	EMNLP'18	-	-	28.23	42.97
	MAN [16]	CVPR'19	-	-	27.02	41.16
2D	TMN [70]	ECCV'18	-	-	22.92	35.17
	VLG-Net [75]	ICCV'21	33.35	25.57	25.57	-
SN	L-Net [21]	AAAI'19	-	-	-	41.43
RL	SM-RL [25]	CVPR'19	-	-	31.06	43.94

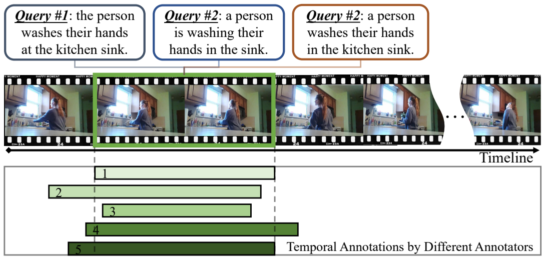

Query #1: the person washes their hands at the kitchen sink.
Query #2: a person is washing their hands in the sink.
Query #2: a person washes their hands in the kitchen sink.
Timeline
1
2
3
4
5
Temporal Annotations by Different Annotators

Fig. 24. Illustration of data uncertainty in TSGV benchmarks. Annotation uncertainty means disagreements of annotated ground-truth moment across different annotators. Query uncertainty means various query expressions for one ground-truth moment.

in Table 7, improving the feature extractor or exploiting more diverse features leads to better accuracy.

Moreover, we also conduct the efficiency comparison among different method categories by selecting one representative model from each category. The results and discussion are presented in Section A.1 in the Appendix.

6 Challenges and Future Directions

6.1 Critical Analysis

Data Uncertainty. Recent studies [92], [224] observe that data samples among current benchmark datasets are ambiguous and inconsistent. First, the annotated moments of a query may be discrepant across annotators, *i.e., annotation uncertainty* [224]. As shown in Fig. 24, for the same query #1, temporal boundaries annotated on the same video by different annotators are different. Such an issue is inevitable due to the subjectivity of annotators. Second, multiple queries may be used to describe the same event/moment, *i.e., query uncertainty*. Fig. 24 also shows that the three queries are attached to the same moment.

For annotation uncertainty, existing methods usually apply single-style annotations, *i.e.,* each data sample is labeled by one annotator, because of the potentially expensive cost of multiple labeling. The inherent uncertainty in moment localization is ignored. Consequently, models may capture single-style prediction bias during training, leading to inferior generalization performance. For query uncertainty, similarly, methods only take one query as input for a moment and encode the query as a deterministic vector. In this case, the variety of query expressions cannot be learned by the model. The model may not well handle queries in different expressions for the same event.

To mitigate annotation uncertainty, Otani *et al.* [224] propose to re-annotate the Charades-STA and ActivityNet Captions datasets on Amazon Mechanical Turk. They also present two alternative evaluation metrics by considering the issues of multiple ground truths and potential miss-labeled samples. Specifically, the first metric evaluates the predicted moments with respect to the nearest-neighbor reference, which is based on the fact that a video may have multiple positive moments for a single query sentence. When a predicted moment is close to at least one of the reference moments, *i.e.,* its IoU with reference moment is larger than a threshold, it is counted as positive. The second metric considers the reliability of human annotations. When a reference moment largely overlaps with the majority of other reference moments, the reference moment is more reliable. A reference moment that is different from others is likely miss-labeled. Zhou *et al.* [92] address both annotation and query uncertainties from a model design perspective. For query uncertainty, they introduce a decoupling method to disentangle each query into a relation feature and a modified feature. The relation feature encodes the discriminative and consistent information; the modified feature encodes the personalized information. Then the modified feature is encoded as Gaussian distribution and a sampling operation is adopted in the latent space to obtain multiple query representations. For annotation uncertainty, they propose a debias mechanism modified from multiple choice learning to generate diverse predictions. Recently, Zhou *et al.* [225] devise a framework to achieve diverse moment localization with only single-label annotations, by constructing soft multi-labels through semantic similarity of multiple video-query pairs. Huang *et al.* [167] introduce elastic moment bounding to accommodate flexible and adaptive moment boundaries. The goal is to model a universally interpretable video-text correlation with tolerance to underlying uncertainties in pre-fixed annotations. Despite many efforts made, the uncertainty issue remains far from being solved.

Data Bias. Otani *et al.* [224] and Yuan *et al.* [126] conduct analysis on Charades-STA and ActivityNet Captions. They count frequent actions in queries and visualize joint distributions of the start and end timestamps of the ground-truth moment. As shown in Fig. 25, a few frequent action verbs cover most of the actions in the dataset, *i.e.,* a long-tail distribution exists in both datasets. A large number of queries describe some common events, while only a few queries cover the remaining actions. Fig. 26 shows that moment distributions are identical in train and test sets with a distinct distributional bias for both datasets. Because of the distributional bias, a TSGV model could make a good guess of the target moment, even without taking into consideration the input video and query [224]. For instance, Otani *et al.* [224] modify 2D-TAN [18] to build a Blind-TAN model by removing the video feature extractor and replacing the map of visual features with a learnable map in the same shape. By training Blind-TAN solely with query sentences, the learnable map may acquire some ideas on when certain actions are likely to happen. Experiments on benchmark datasets show that, without accessing the video content, Blind-TAN achieves comparable performance with state-of-the-art methods. This result demonstrates the severe distributional bias in existing TSGV benchmarks.

TABLE 4
Results of weakly-supervised methods, where MIL is Multi-Instance Learning-based method, REC denotes Reconstruction-based method, and * denotes the zero-shot setting.

Category	Method	Venue	Charades-STA			ActivityNet Captions				DiDeMo		
			R@1, IoU=m			R@1, IoU=m				R@n, IoU=1.0		mIoU
			m=0.3	m=0.5	m=0.7	m=0.1	m=0.3	m=0.5	m=0.7	n=1	n=5	
MIL	TGA [27]	CVPR'19	32.14	19.94	8.84	-	-	-	-	12.19	39.74	24.92
	WSLLN [28]	EMNLP'19	-	-	-	75.40	42.80	22.70	-	19.40	54.40	27.40
	Chen *et al.* [193]	ArXiv'20	39.80	27.30	12.90	74.20	44.30	23.60	-	-	-	-
	VLANet [194]	ECCV'20	45.24	31.83	14.17	-	-	-	-	19.32	65.68	25.33
	CCL [196]	NeurIPS'20	-	33.21	15.68	-	50.12	31.07	-	-	-	-
	BAR [195]	ACM MM'20	51.64	33.98	15.97	-	53.41	33.12	-	-	-	-
	MS-2DTN [222]	ICPR'21	-	30.38	17.31	-	49.79	29.68	-	-	-	-
	LoGAN [204]	WACV'21	51.67	34.68	14.54	-	-	-	-	39.20	64.04	38.28
	VCA [198]	ACM MM'21	58.58	38.13	19.57	67.96	50.45	31.00	-	-	-	-
	FSAN [199]	EMNLP'21	-	-	-	78.45	55.11	29.43	-	19.40	57.85	31.92
	CRM [200]	ICCV'21	53.66	34.76	16.37	81.61	55.26	32.19	-	-	-	-
	LCNet [201]	TIP'21	59.60	39.19	18.87	78.58	48.49	26.33	-	-	-	-
	WSTAN [203]	TMM'21	43.39	29.35	12.28	79.78	52.45	30.01	-	19.40	54.64	31.94
	Teng *et al.* [202]	TMM'21	-	-	-	65.99	44.49	24.33	-	17.00	64.80	29.59
	Chen *et al.* [205]	AAAI'22	43.31	31.02	16.53	71.86	46.62	29.52	-	-	-	-
	MSCL [206]	ArXiv'22	58.92	43.15	23.49	75.61	55.05	38.23	-	-	-	-
	SAN [207]	TMM'22	51.02	31.02	13.12	-	48.44	30.54	13.85	-	-	-
REC	WS-DEC [29]	NeurIPS'18	-	-	-	62.71	41.98	23.34	-	-	-	-
	SCN [30]	AAAI'20	42.96	23.58	9.97	71.48	47.23	29.22	-	-	-	-
	MARN [209]	ArXiv'20	48.55	31.94	14.81	-	47.01	29.95	-	-	-	-
	EC-SL [31]	CVPR'21	-	-	-	68.48	44.29	24.16	-	-	-	-
	CMLNet [208]	TIVC'22	48.99	11.24	-	84.08	49.39	22.58	-	-	-	-
	CNM [213]	AAAI'22	60.39	35.43	15.45	78.13	55.68	33.33	-	-	-	-
	CPL [212]	CVPR'22	65.99	49.05	22.61	71.23	50.07	30.14	-	-	-	-
Other	RTBPN [214]	ACM MM'20	60.04	32.36	13.24	73.73	49.77	29.63	-	20.79	60.26	29.81
	U-VMR [217]	TCSVT'21	46.69	20.14	8.27	69.63	46.15	26.38	11.64	-	-	-
	PSVL [216]*	ICCV'21	46.47	31.29	14.17	-	44.74	30.08	14.74	-	-	-
	PS-VTG [220]	TMM'22	60.40	39.22	20.17	-	59.71	39.59	21.98	-	-	-
	SVPTR [185]	CVPR'22	55.14	32.44	15.53	-	78.07	61.70	38.36	-	-	-
	DSCNet [218]	AAAI'22	44.15	28.73	14.67	-	47.29	28.16	-	-	-	-
	ViGA [221]	SIGIR'22	71.21	45.05	20.27	-	59.61	35.79	16.96	-	-	-

TABLE 5
Result of different visual features on Charades-STA.

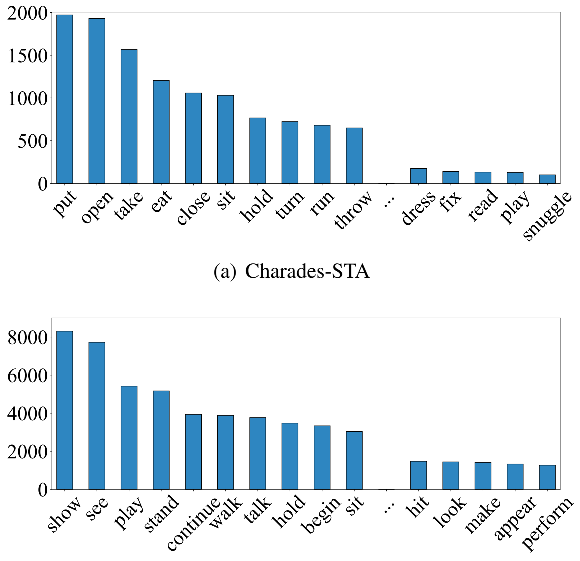
Method	Category	Venue	Feature	R@1, IoU=m		
				m=0.3	m=0.5	m=0.7
DRN [89]	RG	CVPR'20	VGG	-	42.90	23.68
			C3D	-	45.40	26.40
			I3D	-	53.09	31.75
CPNet [91]	RG	AAAI'21	C3D	-	40.32	22.47
			I3D	-	60.27	38.74
BPNet [55]	PG	AAAI'21	C3D	55.46	38.25	20.51
			I3D	65.48	50.75	31.64
LPNet [57]	PG	EMNLP'21	C3D	59.14	40.94	21.13
			I3D	66.59	54.33	34.03
RaNet [76]	2D	EMNLP'21	VGG	-	43.87	26.83
			I3D	-	60.40	39.65
FVMR [79]	2D	ICCV'21	VGG	-	42.36	24.14
			C3D	-	38.16	18.22
			I3D	-	55.01	33.74
HDRR [66]	AN	ACMMM'21	C3D	62.37	43.04	21.32
			TS	68.33	54.06	27.31
			I3D	73.44	59.46	34.11
MIGCN [67]	AN	TIP'21	C3D	-	42.26	22.04
			TS	-	51.80	29.33
			I3D	-	57.10	34.54

TABLE 6
Results of DRFT with different visual modality features on Charades-STA and ActivityNet Captions datasets, where R, F and D denote RGB, flow, and depth modalities, respectively.

Method	Venue	Feature	R@1, IoU=m			mIoU
			m=0.3	m=0.5	m=0.7	
DRFT [93]	NeurIPS'21	Charades-STA				
		R	73.85	60.79	36.72	52.64
		R+F	74.26	61.93	38.69	53.92
		R+F+D	76.68	63.03	40.15	54.89
		ActivityNet Captions				
		R	60.25	42.37	25.23	43.18
		R+F	61.80	43.71	26.43	44.82
		R+F+D	62.91	45.72	27.79	45.86

TABLE 7
Result of PMI with different modality features on the ActivityNet Captions. IRV2 is Inception-ResNet v2 [223] visual feature and A is SoundNet [153] audio feature.

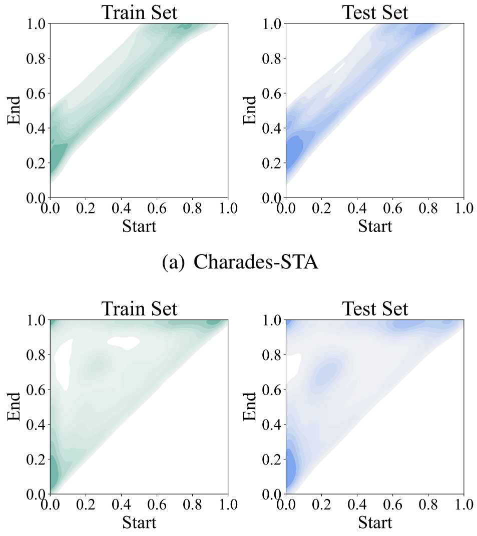
Method	Venue	Feature	R@1, IoU=m		
			m=0.3	m=0.5	m=0.7
PMI [99]	ECCV'20	C3D	59.69	38.28	17.83
		C3D+IRV2	60.16	39.16	18.02
		C3D+IRV2+A	61.22	40.07	18.29



(a) Charades-STA

(b) ActivityNet Captions

Fig. 25. The top-30 frequent actions in Charades-STA and ActivityNet Captions datasets.



(a) Charades-STA

(b) ActivityNet Captions

Fig. 26. An illustration of moment distributions for Charades-STA and ActivityNet Captions, where "Start" and "End" axes represent the normalized start and end timestamps, respectively. The deeper the color, the larger density (*i.e.,* more annotations) in the dataset.

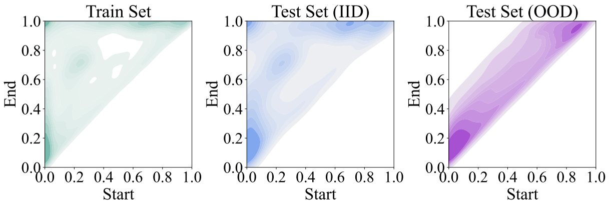


Fig. 27. Illustration of moment distributions of ActivityNet-CD dataset.

To investigate the effects of distributional bias among existing TSGV methods, Yuan *et al.* [126] further re-organize the two benchmark datasets and develop Charades-CD and ActivityNet-CD datasets. Each dataset contains two test sets, *i.e.,* the independent-and-identical distribution (iid) test set, and the out-of-distribution (ood) test set (see Fig. 27). Then, Yuan *et al.* [126] collects a set of SOTA TSGV baselines and evaluates them on the reorganized benchmark datasets. Results show that baselines generally achieve impressive performance on the iid test set, but fail to generalize to the ood test set. It is worth noting that weakly-supervised methods are naturally immune to distributional bias since they do not require annotated samples for training.

Recently, several solutions are proposed to alleviate the distributional bias. Yang *et al.* [226] develop a deconfounded cross-modal matching method to remove distributional bias by leveraging the structured casual mechanism [171]. Luo *et al.* [215] devise a self-supervised method to solve TSGV with pseudo label generation. Zhang *et al.* [227] disentangles bias from the TSGV model by adjusting the losses to compensate for biases dynamically. Liu *et al.* [164] propose a Debiasing-TSG model to filter and remove the negative biases in both vision and language modalities via feature distillation and contrastive sample generation. Hao *et al.* [155] propose to use shuffled videos to address distributional bias without losing grounding accuracy. Specifically, they introduce two auxiliary tasks, *i.e.,* cross-modal matching and temporal order discrimination, to promote the grounding model training. The cross-modal matching task leverages the content consistency between shuffled and original videos, to force the grounding model to mine visual contents to semantically match queries. The temporal order discrimination task leverages the difference in temporal order to strengthen the understanding of long-term temporal contexts. Although solutions are developed to address moment distributional bias *e.g.,* debias strategies and dataset reorganization, it remains unclear if the current benchmarks provide the right setup to evaluate TSGV methods. Meanwhile, the long tail distribution of action verbs in queries has not been well explored.

Recently, because of the inevitable limitations of current benchmark datasets, Soldan *et al.* [121] present the MAD dataset. MAD comprises long-form videos, highly descriptive sentences, and a large diversity in vocabulary. Most importantly, the timestamps of moments are uniformly distributed in the video. Lei *et al.* [228] also develop a new benchmark dataset termed QVHighlights to avert data bias of existing TSGV datasets.

6.2 Future Directions

6.2.1 *Effective Feature Extractor(s)*

Feature quality directly affects TSGV performance. Illustrated in the common pipeline (Fig. 4), the existing solutions mainly extract visual and textual features independently using the corresponding pre-trained visual (*e.g.,* C3D [1] and I3D [3]) and textual extractors (*e.g.,* GloVe [37], BERT [38] and RoBERTa [39]). Thus, there is a large gap between the extracted visual and textual features in different feature spaces. Although TSGV methods attempt to project them into the same feature space, the natural gap between them is hard to be eliminated. There may also be differences between TSGV datasets and the datasets used for pre-training the feature extractors, which leads to information loss or inaccurate representations.

Recently, Zhang *et al.* [77] develop a single-stream feature extraction framework for TSGV, following BERT [38]. Visual and textual features are concatenated and jointly encoded with stacked transformer blocks. Similarly, LocFormer [159] also concatenates

BERT-based query features and I3D-based video features, and feed them into a transformer-based localization module for video grounding. However, the visual and textual features remain separately generated by different pre-trained extractors. Xu *et al.* [97] propose a pre-training strategy for TSGV by constructing a large-scale synthesized dataset with TSGV annotations. Inspired by ViT [229], Cao *et al.* [94] develop a video cubic embedding module to extract 3D visual tokens and learn video content from scratch without reliance on pre-trained visual feature extractors. Although they adopt GloVe [37] embeddings for queries, the issues of feature gap are not well alleviated. Xu *et al.* [138] indicate that, in TSGV, the video encoder is usually fixed during fine-tuning. Thus, it cannot learn temporal boundaries and unseen classes, causing a domain gap with respect to the downstream task. To eliminate this issue, they propose a post-pre-training approach without freezing the video encoder and introduce a masked contrastive learning loss to capture visio-linguistic relations between activities, background video clips, and language queries. Inspired by DETR [190], Liu *et al.* [192] further design a unified multi-modal transformer for jointly optimizing moment retrieval and highlight detection, as well as mitigating the gap between visual and textual features. In general, despite many attempts like utilizing more powerful feature extractors (ViT [229], BERT [38], etc.) or designing unified frameworks to better align video and text features, the semantic discrepancy between visual and textual features remains a key challenge.

With the recent success of pre-training techniques for natural language processing [38], [39] and image-linguistic [230], [231] tasks, video-linguistic pre-training (VLP) is being developed to improve video-text related downstream tasks. For instance, ActBERT [232] proposes to encode joint video-text representations from unlabeled data via self-supervised learning, which leverages global action information to catalyze mutual interactions between text and local regional objects. By designing a set of self-supervised objectives, *e.g.,* masked language modeling, masked action/object classification, and cross-modal matching, ActBERT not only learns the fine-grained video-text representations but also enforces the video and text representations being encoded in the same feature space. Similarly, other video-based vision-language pre-training methods (*e.g.,* BVET [233], ClipBERT [234], VideoCLIP [235], VLM [236], MERIOT [237], etc.) also aim to learn better joint video-text representations by designing more sophisticated objectives or training strategies. After pre-training, these VLP models could be applied to various downstream video-and-language tasks, including text-video clip retrieval, video captioning, video question answering, moment localization, etc.

From the perspective of TSGV, VLP models are good choices for feature extractors. Compared to the traditional visual and textual feature extractors, VLP models well mitigate the gaps between visual and textual features. More importantly, the features extracted by VLP models contain more cross-modal knowledge, which may further boost TSGV performance. However, applying off-the-shelf VLP models for TSGV tasks is yet well explored. On the other hand, the success of VLP also encourages specific pre-training strategies for the TSGV task. Cao *et al.* [187] indicate that almost all existing video-text pre-training methods are limited to retrieval-based downstream tasks. Their transfer potentials to localization-based tasks are underexplored. Based on the observation that current VLP methods are incompatible with localization tasks, they propose a localization-oriented video-text pre-training framework. Zeng *et al.* [162] further design a point prompt tuning paradigm for the TSGV task. In general, localization-oriented VTP is a promising direction for TSGV.

6.2.2 *TSGV with Multiple Answers*

Existing TSGV benchmark datasets generally hold an implicit assumption that, for a query, there is only one ground truth moment exists in the input video. In reality, a query may describe multiple disjoint moments in a video. Based on the observation, Lei *et al.* [228] present a unified benchmark dataset named QVHighlights for both TSGV and highlight detection tasks. In the dataset, each video is annotated with a human-written free-form language query, relevant moments in the video regarding the query, and five-point scale saliency scores for all query-relevant clips. This comprehensive annotation enables researchers to develop and evaluate systems that detect relevant moments as well as salient highlights for diverse and flexible user queries. Specifically, for the TSGV setting, for each language query, QVHighlights provides one or more disjoint moments in the video, enabling a more realistic evaluation of TSGV methods.

To solve TSGV with multiple answers, a method is expected to deeply understand video contents and their semantic relationship to the language query. Inspired by DETR [190], Lei *et al.* [228] further propose Moment-DETR, an end-to-end transformer encoder-decoder architecture that views TSGV as a direct set prediction problem. The model takes the extracted video and query representations as inputs and predicts moment coordinates and saliency scores end-to-end, without any human prior, such as proposal generation or non-maximum suppression. From the query perspective, Bao *et al.* [86] convert TSGV to dense events grounding task, which aims to jointly localize the multiple moments described in a paragraph *i.e.,* multiple queries. Then they devise a DepNet to adaptively aggregate the temporal and semantic information of dense events into a compact set, and selectively propagate the aggregated information to every single event with soft attention. Based on DepNet, Shi *et al.* [183] further present an end-to-end parallel decoding paradigm by re-purposing a transformer-alike architecture from the perspective of TSGV, as language-conditioned regression. Jiang *et al.* [184] build a graph-based transformer with language reconstruction to jointly extract temporal moments and reconstruct queries with extracted moments for explainability. Liu *et al.* [192] also apply a DETR-style framework and introduce pre-training with ASR captions for this task.

In general, jointly localizing multiple moments helps to alleviate the bias in TSGV. Joint localization may also help to improve overall accuracy as moments are semantically correlated and temporally coordinated by their order in a video. Multiple answers for a query is a novel task extended from the standard TSGV task, and this setting is more realistic and less biased.

6.2.3 *Spatio-Temporal Sentence Grounding in Videos*

Spatio-temporal sentence grounding in videos (STSGV) is another extension of TSGV. The goal of TSGV is to extract a temporal moment, *i.e.,* detecting the start and end timestamps in a video for a language query. One step further, given a query, STSGV aims to sequentially localize the referring instances in a sequence of continuous frames in the video *i.e.,* a spatio-temporal tube. Compared to TSGV, STSGV is more complicated since the task requires localizing not only the event's temporal boundaries but also the bounding boxes among frames in the video segment. Recently, a series of work [238]–[255] has been proposed for this problem. A number of datasets are made available, including VID-sentence [240] which is based on ImageNet video object detection, ActivityNet-SRL [242] from existing caption and grounding datasets, VidSTG [243], and HC-STVG [247]. For instance, Lin *et al.* [254] propose a STVGFormer with static-dynamic cross-modal understanding. In STVGFormer, a static branch learns to predict the spatial location according to static cues like human appearance; a dynamic branch learns to predict temporal boundaries according to dynamic cues like human action. Then a static-dynamic interaction block is designed to enable the two branches to query useful and complementary information from the opposite branch. Yang *et al.* [255] devise TubeDETR, inspired by the success of DETR-based architecture for text-conditioned object detection. TubeDETR jointly encodes text, appearance, and motion information, with the aim to predict the moment's temporal and spatial boundaries simultaneously.

Despite the availability of multiple datasets and methods, annotating spatio-temporal tubes in the video is more difficult and labor-intensive, compared to TSGV annotation. Thus, many methods [238]–[241], [248] seek to solve STSGV under the weakly-supervised setting, which does not require fully annotated datasets. For instance, Chen *et al.* [240] utilize a pre-trained instance generator to produce spatio-temporal instances from video. They then adopt an attentive interactor to exploit the complicated relationships between instances and the sentence. The overall model is optimized through a multiple-instance learning strategy. Tan *et al.* [248] further design a self-supervised grounding mechanism with a contrastive multi-layer multi-modal attention module to locate spatial-temporal tubes in the video. Although some promising results are obtained, STSGV remains in its early stage.

6.2.4 Multi-modal Temporal Grounding in Video

TSGV is a form of temporal video grounding using text as query, *i.e.,* language modality. Other modalities, such as audio, image, and short video clip, may also serve as queries for temporal video grounding. In fact, temporal video grounding with other modalities has also been studied in recent years, such as audio-visual event localization, image-to-video retrieval, and video re-localization. To be specific, audio-visual event localization (AVEL) [256]–[263] is to retrieve the synchronized video segment for a given audio from an untrimmed video. The task of image-to-video retrieval (IVR) [264]–[267] is to localize video segments that contain similar activity as in the query image. Similarly, given a query video and a reference video, video re-localization (VRL) [268]–[271] aims to retrieve a segment in the reference video that semantically corresponds to the query video. Conceptually, the query is in the form of audio in AVEL, appearance vision (image) in IVR, and motion vision (video clip) in VRL, respectively. Despite the different query modalities used in AVEL, IVR, and VRL, their overall modeling process is similar to TSGV from the perspective of feature space. For instance, for AVEL, Xuan *et al.* [259] adopt a VGG network pre-trained on AudioSet [272] to extract feature sequence for audio, and use ResNet to extract feature sequence for video. Then an attentive-based cross-modal network is applied to learn the multimodal interactions between video and audio to perform event localization. For IVR, Liu *et al.* [267] extract the image query feature through a pre-trained VGG network, and the video feature sequence via R-C3D [52] model. Similarly, for VRL, Feng *et al.* [268] utilize the pre-trained C3D [1] to extract feature sequences of both query and reference videos. To sum up, after feature extraction for query modality, the input formats of TSGV, AVEL, IVR, and VRL are almost the same. In principle, solutions to these tasks should be similar to each other, except for some subtle differences. However, compared to TSGV, temporal video grounding with such modalities is not widely studied.

Different query modalities could provide extra guidance to boost the performance for moment localization in videos. For instance, audio signals (*e.g.,* dog bark, noise in kitchen) offer auxiliary clues [99], [273] for precise localization. Audio transcription from video (if exists) using ASR [274], [275] could provide relevant information for cross-modal alignment between the video and query. For instance, Chen *et al.* [99] introduce audio as an additional feature to TSGV and achieve better performance on several benchmark datasets. From the perspective of the query, different modalities of the query (*e.g.,* audio, sentence, and image) that describe the same event can be used for cross-validation of the retrieved results. Although TSGV, AVEL, IVR, and VRL accept different query modalities as a whole, there is a lack of a unified framework, which is suitable for all settings.

6.2.5 Video Corpus Moment Retrieval

Video corpus moment retrieval (VCMR) extends video sources from a single video in TSGV to a large collection of videos. That is, VCMR aims to retrieve a matching moment to a query from a collection of untrimmed and unsegmented videos, *i.e.,* a video corpus. VCMR poses challenges to efficiently identify the relevant videos and to localize the relevant moments in the identified videos. Escorcia *et al.* [276] first extend TSGV to VCMR by modifying the existing TSGV benchmark datasets (*i.e.,* DiDeMo, Charades-STA, and ActivityNet Captions) to fit the VCMR setting. Then, they devise a clip-query alignment model, which learns to align the features of a natural language query to a sequence of short video clips that compose a candidate moment in a video. Lei *et al.* [277] construct the TVR dataset, where the videos come with associated textual subtitles and each query is associated with a tight temporal window in the corresponding video. This dataset is specifically designed for both TSGV and VCMR tasks, where it contains more than $100k$ queries collected on $21.8k$ videos from 6 TV shows of diverse genres. Based on TVR, Lei *et al.* [278] further extend it to a multilingual version named mTVR, which contains both English and Chinese queries. One of the purposes of mTVR is to investigate the generalization ability of VCMR models from one language to another language.

A number of methods [279]–[289] has been developed for VCMR. Li *et al.* [279] design a hierarchical transformer-based model for video-language omni-representation learning and fine-tuning on TVR dataset. Zhang *et al.* [280] develop a hierarchical multi-modal encoder to learn multimodal interactions at both coarse- and fine-grained granularities. Zhang *et al.* [281] introduce contrastive learning to replace the time-consuming multimodal interaction strategy in VCMR to achieve a balance between efficiency and retrieval accuracy. Hou *et al.* [284] develop a two-step multimodal fusion for precise and efficient moment retrieval. Paul *et al.* [283] propose a hierarchical moment alignment network to effectively learn a joint embedding space to align the corresponding video moments and sentences. Liu *et al.* [287] further explore to solve the multilingual VCMR problem by devising a cross-lingual cross-modal consolidation strategy.

In general, VCMR contains two sub-tasks, *i.e.,* video retrieval, and moment localization. If a TSGV model is directly adapted, the query needs to interact with every video in the corpus, which is infeasible. However, VCMR is closer to practical scenarios as videos are ubiquitous. Although existing solutions to VCMR achieve consecutive improvements, the performance of VCMR is still inferior for real-world applications.

7 Conclusion

Many techniques are available to learn dense representations of various types of data *e.g.,* text, video, and audio. Through multimodal interaction, cross-modal applications like TSGV become feasible. In this survey, we start with how to extract features from text and video, then focus on the interaction between the two types of features for TSGV. Although TSGV has a short history, we have seen the trend of development from sliding window methods to proposal-based and proposal-free methods, then different views of the task with solutions from reinforcement learning and weakly-supervised learning. At the same time, we also see the challenges in this field; hence the results obtained on benchmark datasets may not necessarily reflect a model's performance in reality. Addressing these challenges would certainly bring improvements to current solutions. Furthermore, as a fundamental task, solutions to TSGV directly benefit many more related applications like spatio-temporal sentence grounding in videos and video corpus moment retrieval. We hope this survey could serve as a good reference for researchers working on these interesting problems.

References

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[2] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2016.

[3] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[4] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-tad: Sub-graph localization for temporal action detection," in *CVPR*, 2020.

[5] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *ACL*, 2017.

[6] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," in *ICLR*, 2017.

[7] A. W. Yu, D. Dohan, Q. Le, T. Luong, R. Zhao, and K. Chen, "Fast and accurate reading comprehension by combining self-attention and convolution," in *ICLR*, 2018.

[8] H. Huang, C. Zhu, Y. Shen, and W. Chen, "Fusionnet: Fusing via fully-aware attention with application to machine comprehension," in *ICLR*, 2018.

[9] J. Gao, C. Sun, Z. Yang, and R. Nevatia, "Tall: Temporal activity localization via language query," in *ICCV*, 2017.

[10] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with natural language," in *ICCV*, 2017.

[11] M. Liu, X. Wang, L. Nie, Q. Tian, B. Chen, and T.-S. Chua, "Cross-modal moment localization in videos," in *ACM MM*, 2018.

[12] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua, "Attentive moment retrieval in videos," in *SIGIR*, 2018.

[13] R. Ge, J. Gao, K. Chen, and R. Nevatia, "Mac: Mining activity concepts for language-based temporal localization," in *WACV*, 2019.

[14] H. Xu, K. He, L. Sigal, S. Sclaroff, and K. Saenko, "Text-to-clip video retrieval with early fusion and re-captioning," *ArXiv*, vol. abs/1804.05113, 2018.

[15] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, "Temporally grounding natural sentence in video," in *EMNLP*, 2018.

[16] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, and L. S. Davis, "Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment," in *CVPR*, 2019.

[17] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, "Semantic conditioned dynamic modulation for temporal sentence grounding in videos," in *NeurIPS*, 2019.

[18] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2d temporal adjacent networks formoment localization with natural language," in *AAAI*, vol. 34, 2020.

[19] Y. Yuan, T. Mei, and W. Zhu, "To find where you talk: Temporal sentence localization in video with attention based location regression," in *AAAI*, vol. 33, 2019.

[20] S. Ghosh, A. Agarwal, Z. Parekh, and A. Hauptmann, "ExCL: Extractive Clip Localization Using Natural Language Descriptions," in *NAACL*, 2019.

[21] J. Chen, L. Ma, X. Chen, Z. Jie, and J. Luo, "Localizing natural language in videos," in *AAAI*, vol. 33, 2019.

[22] C. Lu, L. Chen, C. Tan, X. Li, and J. Xiao, "DEBUG: A dense bottom-up grounding approach for natural language video localization," in *EMNLP*, 2019.

[23] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, "Span-based localizing network for natural language video localization," in *ACL*, 2020.

[24] D. He, X. Zhao, J. Huang, F. Li, X. Liu, and S. Wen, "Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos," in *AAAI*, vol. 33, 2019.

[25] W. Wang, Y. Huang, and L. Wang, "Language-driven temporal activity localization: A semantic matching reinforcement learning model," in *CVPR*, 2019.

[26] J. Wu, G. Li, S. Liu, and L. Lin, "Tree-structured policy based progressive reinforcement learning for temporally language grounding in video," in *AAAI*, vol. 34, 2020.

[27] N. C. Mithun, S. Paul, and A. K. Roy-Chowdhury, "Weakly supervised video moment retrieval from text queries," in *CVPR*, 2019.

[28] M. Gao, L. Davis, R. Socher, and C. Xiong, "WSLLN:weakly supervised natural language localization networks," in *EMNLP*, 2019.

[29] X. Duan, W. Huang, C. Gan, J. Wang, W. Zhu, and J. Huang, "Weakly supervised dense event captioning in videos," in *NeurIPS*, vol. 31, 2018.

[30] Z. Lin, Z. Zhao, Z. Zhang, Q. Wang, and H. Liu, "Weakly-supervised video moment retrieval via semantic completion network," in *AAAI*, vol. 34, 2020.

[31] S. Chen and Y.-G. Jiang, "Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning," in *CVPR*, 2021.

[32] Y. Yang, Z. Li, and G. Zeng, "A survey of temporal activity localization via language in untrimmed videos," in *ICCST*, 2020.

[33] X. Liu, X. Nie, Z. Tan, J. Guo, and Y. Yin, "A survey on natural language video localization," *ArXiv*, vol. abs/2104.00234, 2021.

[34] X. Lan, Y. Yuan, X. Wang, Z. Wang, and W. Zhu, "A survey on temporal sentence grounding in videos," *ArXiv*, vol. abs/2109.08039, 2021.

[35] M. Liu, L. Nie, Y. Wang, M. Wang, and Y. Rui, "A survey on video moment localization," *ACM Comput. Surv.*, 2022.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ArXiv*, vol. abs/1301.3781, 2013.

[37] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *EMNLP*, 2014.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[39] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[40] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *NeurIPS*, vol. 28, 2015.

[41] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *EMNLP*, 2017.

[42] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *EMNLP*, 2019.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv*, vol. abs/1409.1556, 2014.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[45] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with temporal language," in *EMNLP*, 2018.

[46] A. Wu and Y. Han, "Multi-modal circulant fusion for video-to-language and backward," in *IJCAI*, 2018.

[47] S. Zhang, J. Su, and J. Luo, "Exploiting temporal relationships in video moment localization with natural language," in *ACM MM*, 2019.

[48] B. Jiang, X. Huang, C. Yang, and J. Yuan, "Cross-modal video moment retrieval with spatial and language-temporal attention," in *ACM ICMR*, 2019.

[49] K. Ning, M. Cai, D. Xie, and F. Wu, "An attentive sequence to sequence translator for localizing video clips by natural language," *IEEE TMM*, vol. 22, 2020.

[50] Y. Zeng, D. Cao, X. Wei, M. Liu, Z. Zhao, and Z. Qin, "Multi-modal relational graph for cross-modal video moment retrieval," in *CVPR*, 2021.

[51] K. Ning, L. Xie, J. Liu, F. Wu, and Q. Tian, "Interaction-integrated network for natural language moment localization," *IEEE TIP*, vol. 30, 2021.

[52] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *ICCV*, 2017.

[53] H. Xu, K. He, B. A. Plummer, L. Sigal, S. Sclaroff, and K. Saenko, "Multilevel language and vision integration for text-to-clip retrieval," in *AAAI*, vol. 33, 2019.

[54] S. Chen and Y.-G. Jiang, "Semantic proposal for activity localization in videos via sentence query," in *AAAI*, vol. 33, 2019.

[55] S. Xiao, L. Chen, S. Zhang, W. Ji, J. Shao, L. Ye, and J. Xiao, "Boundary proposal network for two-stage natural language video localization," in *AAAI*, vol. 35, 2021.

[56] D. Liu, X. Qu, J. Dong, and P. Zhou, "Adaptive proposal generation network for temporal sentence localization in videos," in *EMNLP*, 2021.

[57] S. Xiao, L. Chen, J. Shao, Y. Zhuang, and J. Xiao, "Natural language video localization with learnable moment proposals," in *EMNLP*, 2021.

[58] Y. Hu, M. Liu, X. Su, Z. Gao, and L. Nie, "Video moment localization via deep cross-modal hashing," *IEEE TIP*, vol. 30, 2021.

[59] Z. Zhang, Z. Lin, Z. Zhao, and Z. Xiao, "Cross-modal interaction networks for query-based moment retrieval in videos," in *SIGIR*, 2019.

[60] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, "Semantic conditioned dynamic modulation for temporal sentence grounding in videos," *IEEE TPAMI*, vol. 1, 2020.

[61] Z. Lin, Z. Zhao, Z. Zhang, Z. Zhang, and D. Cai, "Moment retrieval via cross-modal interaction networks with query reconstruction," *IEEE TIP*, vol. 29, 2020.

[62] J. Wang, L. Ma, and W. Jiang, "Temporally grounding language queries in videos by contextual boundary-aware prediction," in *AAAI*, 2020.

[63] X. Qu, P. Tang, Z. Zou, Y. Cheng, J. Dong, P. Zhou, and Z. Xu, "Fine-grained iterative attention network for temporal language localization in videos," in *ACM MM*, 2020.

[64] D. Liu, X. Qu, X.-Y. Liu, J. Dong, P. Zhou, and Z. Xu, "Jointly cross- and self-modal graph attention network for query-based moment localization," in *ACM MM*, 2020.

[65] D. Liu, X. Qu, J. Dong, and P. Zhou, "Reasoning step-by-step: Temporal sentence localization in videos via deep rectification-modulation network," in *COLING*, 2020.

[66] Z. Ma, X. Han, X. Song, Y. Cui, and L. Nie, "Hierarchical deep residual reasoning for temporal moment localization," in *ACM MM Asia*, 2021.

[67] Z. Zhang, X. Han, X. Song, Y. Yan, and L. Nie, "Multi-modal interaction graph convolutional network for temporal language localization in videos," *IEEE TIP*, vol. 30, 2021.

[68] D. Liu, X. Qu, and P. Zhou, "Progressively guide to attend: An iterative alignment framework for temporal sentence grounding," in *EMNLP*, 2021.

[69] W. Wang, J. Cheng, and S. Liu, "Dct-net: A deep co-interactive transformer network for video temporal grounding," *Image and Vision Computing*, vol. 110, 2021.

[70] B. Liu, S. Yeung, E. Chou, D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Temporal modular networks for retrieving complex compositional activities in videos," in *ECCV*, 2018.

[71] S. Zhang, H. Peng, J. Fu, Y. Lu, and J. Luo, "Multi-scale 2d temporal adjacency networks for moment localization with natural language," *IEEE TPAMI*, 2021.

[72] Q. Zheng, J. Dong, X. Qu, X. Yang, S. Ji, and X. Wang, "Progressive localization networks for language-based moment localization," *ArXiv*, vol. abs/2102.01282, 2021.

[73] H. Wang, Z.-J. Zha, L. Li, D. Liu, and J. Luo, "Structured multi-level interaction network for video moment localization via language query," in *CVPR*, 2021.

[74] Y. Hu, L. Nie, M. Liu, K. Wang, Y. Wang, and X.-S. Hua, "Coarse-to-fine semantic alignment for cross-modal moment localization," *IEEE TIP*, vol. 30, 2021.

[75] M. Soldan, M. Xu, S. Qu, J. Tegner, and B. Ghanem, "Vlg-net: Video-language graph matching network for video grounding," in *ICCV*, 2021.

[76] J. Gao, X. Sun, M. Xu, X. Zhou, and B. Ghanem, "Relation-aware video reading comprehension for temporal language grounding," in *EMNLP*, 2021.

[77] M. Zhang, Y. Yang, X. Chen, Y. Ji, X. Xu, J. Li, and H. T. Shen, "Multi-stage aggregated transformer network for temporal language localization in videos," in *CVPR*, 2021.

[78] Q. Huang, J. Wei, Y. Cai, C. Zheng, J. Chen, H.-f. Leung, and Q. Li, "Aligned dual channel graph convolutional network for visual question answering," in *ACL*, 2020.

[79] J. Gao and C. Xu, "Fast video moment retrieval," in *ICCV*, 2021.

[80] Z. Wu, J. Gao, S. Huang, and C. Xu, "Diving into the relations: Leveraging semantic and visual structures for video moment retrieval," in *ICME*, 2021.

[81] Z. Wang, L. Wang, T. Wu, T. Li, and G. Wu, "Negative sample matters: A renaissance of metric learning for temporal grounding," *ArXiv*, vol. abs/2109.04872, 2021.

[82] Z. Jia, M. Dong, J. Ru, L. Xue, S. Yang, and C. Li, "Stcm-net: A symmetrical one-stage network for temporal language localization in videos," *Neurocomputing*, vol. 471, 2022.

[83] D. Shao, Y. Xiong, Y. Zhao, Q. Huang, Y. Qiao, and D. Lin, "Find and focus: Retrieve and localize video events with natural language queries," in *ECCV*, 2018.

[84] D. Liu, X. Qu, J. Dong, P. Zhou, Y. Cheng, W. Wei, Z. Xu, and Y. Xie, "Context-aware biaffine localizing network for temporal sentence grounding," in *CVPR*, 2021.

[85] H. Wang, Z.-J. Zha, X. Chen, Z. Xiong, and J. Luo, "Dual path interaction network for video moment localization," in *ACM MM*, 2020.

[86] P. Bao, Q. Zheng, and Y. Mu, "Dense events grounding in video," in *AAAI*, vol. 35, 2021.

[87] X. Ding, N. Wang, S. Zhang, D. Cheng, X. Li, Z. Huang, M. Tang, and X. Gao, "Support-set based cross-supervision for video grounding," in *ICCV*, 2021.

[88] B. Zhang, Y. Li, C. Yuan, D. Xu, P. Jiang, and Y. Shan, "A simple yet effective method for video temporal grounding with cross-modality attention," *ArXiv*, vol. abs/2009.11232, 2020.

[89] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, "Dense regression network for video grounding," in *CVPR*, 2020.

[90] J. Mun, M. Cho, and B. Han, "Local-global video-text interactions for temporal grounding," in *CVPR*, 2020.

[91] K. Li, D. Guo, and M. Wang, "Proposal-free video grounding with contextual pyramid network," in *AAAI*, vol. 35, 2021.

[92] H. Zhou, C. Zhang, Y. Luo, Y. Chen, and C. Hu, "Embracing uncertainty: Decoupling and de-bias for robust temporal grounding," in *CVPR*, 2021.

[93] Y.-W. Chen, Y.-H. Tsai, and M.-H. Yang, "End-to-end multi-modal video temporal grounding," in *NeurIPS*, vol. 34, 2021.

[94] M. Cao, L. Chen, M. Z. Shou, C. Zhang, and Y. Zou, "On pursuit of designing multi-modal transformer for video grounding," in *EMNLP*, 2021.

[95] L. Chen, C. Lu, S. Tang, J. Xiao, D. Zhang, C. Tan, and X. Li, "Rethinking the bottom-up framework for query-based video localization," in *AAAI*, vol. 34, 2020.

[96] X. Liu, X. Nie, J. Teng, L. Lian, and Y. Yin, "Single-shot semantic matching network for moment localization in videos," *ACM TOMCCAP*, vol. 17, 2021.

[97] M. Xu, J.-M. Pérez-Rúa, V. Escorcia, B. Martinez, X. Zhu, L. Zhang, B. Ghanem, and T. Xiang, "Boundary-sensitive pre-training for temporal localization in videos," in *ICCV*, 2021.

[98] S. Chen and Y.-G. Jiang, "Hierarchical visual-textual graph for temporal activity localization via language," in *ECCV*, 2020.

[99] S. Chen, W. Jiang, W. Liu, and Y.-G. Jiang, "Learning modality interaction for temporal sentence localization and event captioning in videos," in *ECCV*, 2020.

[100] H. Zhang, A. Sun, W. Jing, L. Zhen, J. T. Zhou, and R. S. M. Goh, "Natural language video localization: A revisit in span-based question answering framework," *IEEE TPAMI*, vol. 1, 2021.

[101] H. Zhang, A. Sun, W. Jing, L. Zhen, J. T. Zhou, and S. M. R. Goh, "Parallel attention network with sequence matching for video grounding," in *Findings of ACL*, 2021.

[102] G. Nan, R. Qiao, Y. Xiao, J. Liu, S. Leng, H. Zhang, and W. Lu, "Interventional video grounding with dual contrastive learning," in *CVPR*, 2021.

[103] X. Yu, M. Malmir, X. He, J. Chen, T. Wang, Y. Wu, Y. Liu, and Y. Liu, "Cross interaction network for natural language guided video moment retrieval," in *SIGIR*, 2021.

[104] H. Tang, J. Zhu, L. Wang, Q. Zheng, and T. Zhang, "Multi-level query interaction for temporal language grounding," *IEEE TITS*, 2021.

[105] H. Tang, J. Zhu, M. Liu, Z. Gao, and Z. Cheng, "Frame-wise cross-modal matching for video moment retrieval," *IEEE TMM*, 2021.

[106] Z. Zhang, Z. Zhao, Z. Zhang, Z. Lin, Q. Wang, and R. Hong, "Temporal textual localization in video via adversarial bi-directional interaction networks," *IEEE TMM*, vol. 23, 2021.

[107] S. Qi, L. Yang, C. Li, and Y. Huang, "Collaborative spatial-temporal interaction for language-based moment retrieval," in *WCSP*, 2021.

[108] L. Zhang and R. J. Radke, "Natural language video moment localization through query-controlled temporal convolution," in *WACV*, 2022.

[109] W. Gou, W. Shi, J. Lou, L. Huang, P. Zhou, and R. Li, "Sneak: Synonymous sentences-aware adversarial attack on natural language video localization," *ArXiv*, vol. abs/2112.04154, 2021.

[110] C. Rodriguez, E. Marrese-Taylor, F. S. Saleh, H. LI, and S. Gould, "Proposal-free temporal moment localization of a natural-language query in video using guided attention," in *WACV*, 2020.

[111] Y. Zhao, Z. Zhao, Z. Zhang, and Z. Lin, "Cascaded prediction network via segment tree for temporal video grounding," in *CVPR*, 2021.

[112] G. Liang, S. Ji, and Y. Zhang, "Local-enhanced interaction for temporal moment localization," in *ACM ICMR*, 2021.

[113] C. Rodriguez-Opazo, E. Marrese-Taylor, B. Fernando, H. Li, and S. Gould, "Dori: Discovering object relationships for moment localization of a natural language query in a video," in *WACV*, 2021.

[114] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, 2016.

[115] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *ECCV*, 2016.

[116] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *ACL: System Demonstrations*, 2014.

[117] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," in *ICCV*, 2017.

[118] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, 2015.

[119] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal, "Grounding action descriptions in videos," *TACL*, vol. 1, 2013.

[120] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, "Script data for attribute-based recognition of composite activities," in *ECCV*, 2012.

[121] M. Soldan, A. Pardo, J. L. Alcázar, F. C. Heilbron, C. Zhao, S. Giancola, and B. Ghanem, "Mad: A scalable dataset for language grounding in videos from movie audio descriptions," in *CVPR*, 2022.

[122] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[123] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE TPAMI*, vol. 39, 2017.

[124] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE TPAMI*, vol. 42, 2020.

[125] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, "Natural language object retrieval," in *CVPR*, 2016.

[126] Y. Yuan, X. Lan, X. Wang, L. Chen, Z. Wang, and W. Zhu, "A closer look at temporal sentence grounding in videos: Dataset and metric," in *ACM HUMA*, 2021.

[127] Y. Hu, Y. Xu, Y. Zhang, R. Feng, T. Zhang, X. Lu, and S. Gao, "Camg: Context-aware moment graph network for multimodal temporal activity localization via language," *SSRN*, 2022.

[128] J. Gao, X. Sun, B. Ghanem, X. Zhou, and S. Ge, "Efficient video grounding with which-where reading comprehension," *IEEE TCSVT*, vol. 32, 2022.

[129] D. Liu and W. Hu, "Skimming, locating, then perusing: A human-like framework for natural language video localization," in *ACM MM*, 2022.

[130] X. Liu, X. Nie, J. Teng, F. Hao, and Y. Yin, "Eccl: Explicit correlation-based convolution boundary locator for moment localization," in *ICASSP*, 2021.

[131] D. Liu, X. Fang, W. Hu, and P. Zhou, "Exploring optical-flow-guided motion and detection-based appearance for temporal sentence grounding," *ArXiv*, vol. abs/2203.02966, 2022.

[132] P. Shi and J. Lin, "Simple bert models for relation extraction and semantic role labeling," *ArXiv*, vol. abs/1904.05255, 2019.

[133] Z. Wu, J. Gao, S. Huang, and C. Xu, "Learning commonsense-aware moment-text alignment for fast video temporal grounding," *ArXiv*, vol. abs/2204.01450, 2022.

[134] C. Guo, D. Liu, and P. Zhou, "A hybird alignment loss for temporal moment localization with natural language," in *ICME*, 2022.

[135] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *ICCV*, 2009.

[136] B. Zhang, B. Jiang, C. Yang, and L. Pang, "Dual-channel localization networks for moment retrieval with natural language," in *ACM ICMR*, 2022.

[137] J. Shin and J. Moon, "Learning to combine the modalities of language and video for temporal moment localization," *Computer Vision and Image Understanding*, vol. 217, 2022.

[138] M. Xu, E. Gundogdu, M. Lapin, B. Ghanem, M. Donoser, and L. Bazzani, "Contrastive language-action pre-training for temporal localization," *ArXiv*, vol. abs/2204.12293, 2022.

[139] P. Bao and Y. Mu, "Learning sample importance for cross-scenario video temporal grounding," in *ACM ICMR*, 2022.

[140] M. Zheng, D.-Q. Yang, Z. Ye, T. Lei, Y. Peng, and Y. Liu, "Team pku-wict-mipl pic makeup temporal video grounding challenge 2022 technical report," *ArXiv*, vol. abs/2207.02687, 2022.

[141] X. Ding, N. Wang, S. Zhang, Z. Huang, X. Li, M. Tang, T. Liu, and X. Gao, "Exploring language hierarchy for video grounding," *IEEE TIP*, vol. 31, 2022.

[142] G. Wang, X. Xu, F. Shen, H. Lu, Y. Ji, and H. T. Shen, "Cross-modal dynamic networks for video moment retrieval with text query," *IEEE TMM*, vol. 24, 2022.

[143] G. Wang, X. Jiang, N. Liu, and X. Xu, "Language-enhanced object reasoning networks for video moment retrieval with text query," *Computers and Electrical Engineering*, vol. 102, 2022.

[144] X. Sun, X. Wang, J. Gao, Q. Liu, and X. Zhou, "You need to read again: Multi-granularity perception network for moment retrieval in videos," in *SIGIR*, 2022.

[145] J. Li, J. Xie, L. Qian, L. Zhu, S. Tang, F. Wu, Y. Yang, Y. Zhuang, and X. Wang, "Compositional temporal grounding with structured variational cross-graph correspondence learning," in *CVPR*, 2022.

[146] X. Fang, D. Liu, P. Zhou, Z. Xu, and R. Li, "Hierarchical local-global transformer for temporal sentence grounding," *ArXiv*, vol. abs/2208.14882, 2022.

[147] Z. Xu, D. Chen, K. Wei, C. Deng, and H. Xue, "Hisa: Hierarchically semantic associating for video temporal grounding," *IEEE TIP*, vol. 31, 2022.

[148] D. Liu, X. Qu, X. Di, Y. Cheng, Z. Xu, and P. Zhou, "Memory-guided semantic learning network for temporal sentence grounding," in *AAAI*, 2022.

[149] S. Li, C. Li, M. Zheng, and Y. Liu, "Phrase-level prediction for video temporal localization," in *ACM ICMR*, 2022.

[150] Z. Guo, Z. Zhao, W. Jin, D. Wang, R. Liu, and J. Yu, "Taohighlight: Commodity-aware multi-modal video highlight detection in e-commerce," *IEEE TMM*, vol. 24, 2022.

[151] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, vol. 30, 2017.

[152] D. Liu, X. Qu, P. Zhou, and Y. Liu, "Exploring motion and appearance information for temporal sentence grounding," in *AAAI*, 2022.

[153] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *NeurIPS*, vol. 29, 2016.

[154] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," in *ACL*, 2018.

[155] J. Hao, H. Sun, P. Ren, J. Wang, Q. Qi, and J. Liao, "Can shuffling video benefit temporal bias problem: A novel training framework for temporal grounding," *ArXiv*, vol. abs/2207.14698, 2022.

[156] S. Qi, L. Yang, C. Li, and Y. Huang, "Coarse-to-fine spatial-temporal relationship inference for temporal sentence grounding," *IEEE Access*, vol. 9, 2021.

[157] S. Yang and X. Wu, "Entity-aware and motion-aware transformers for language-driven action localization in videos," in *IJCAI*, 2022.

[158] X. Shen, L. Lan, H. Tan, X. Zhang, X. Ma, and Z. Luo, "Joint modality synergy and spatio-temporal cue purification for moment localization," in *ACM ICMR*, 2022.

[159] C. Rodriguez-Opazo, E. Marrese-Taylor, B. Fernando, H. Takamura, and Q. Wu, "Locformer: Enabling transformers to perform temporal moment localization on long untrimmed videos with a feature sampling approach," *ArXiv*, vol. abs/2112.10066, 2021.

[160] H. Fu and H. Wang, "Multiple cross-attention for video-subtitle moment retrieval," *Pattern Recognition Letters*, vol. 156, 2022.

[161] L. Zhang and R. J. Radke, "Natural language video moment localization through query-controlled temporal convolution," in *WACV*, 2022.

[162] Y. Zeng, "Point prompt tuning for temporally language grounding," in *SIGIR*, 2022.

[163] J. Hao, H. Sun, P. Ren, J. Wang, Q. Qi, and J. Liao, "Query-aware video encoder for video moment retrieval," *Neurocomputing*, vol. 483, 2022.

[164] D. Liu, X. Qu, and W. Hu, "Reducing the vision and language bias for temporal sentence grounding," in *ACM MM*, 2022.

[165] Y. Xu, Y. Zhang, R. Feng, R.-W. Zhao, T. Zhang, X. Lu, and S. Gao, "Stdnet: Spatio-temporal decomposed network for video grounding," in *ICME*, 2022.

[166] B. Li, Y. Weng, B. Sun, and S. Li, "Towards visual-prompt temporal answering grounding in medical instructional video," in *ACM MM*, 2022.

[167] J. Huang, H. Jin, S. Gong, and Y. Liu, "Video activity localisation with uncertainties in temporal boundary," *ArXiv*, vol. abs/2206.12923, 2022.

[168] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *ACL*, 2016.

[169] J. T. Zhou, H. Zhang, D. Jin, H. Zhu, M. Fang, R. S. M. Goh, and K. Kwok, "Dual adversarial neural transfer for low-resource named entity recognition," in *ACL*, 2019.

[170] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," in *ACL*, 2020.

[171] J. Pearl, M. Glymour, and N. P. Jewell, *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[172] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[173] A. Shapiro, "Monte carlo sampling methods," *Handbooks in operations research and management science*, vol. 10, 2003.

[174] D. Li, H. Wu, J. Zhang, and K. Huang, "A2-rl: Aesthetics aware reinforcement learning for image cropping," in *CVPR*, 2018.

[175] D. Cao, Y. Zeng, X. Wei, L. Nie, R. Hong, and Z. Qin, "Adversarial video moment retrieval by jointly modeling ranking and localization," in *ACM MM*, 2020.

[176] Y. Zeng, D. Cao, S. Lu, H. Zhang, J. Xu, and Z. Qin, "Moment is important: Language-based video moment retrieval via adversarial learning," *ACM TMCCA*, vol. 18, 2022.

[177] D. Cao, Y. Zeng, M. Liu, X. He, M. Wang, and Z. Qin, "Strong: Spatio-temporal reinforcement learning for cross-modal video moment localization," in *ACM MM*, 2020.

[178] M. Hahn, A. Kadav, J. M. Rehg, and H. P. Graf, "Tripping through time: Efficient localization of activities in videos," in *BMVC*, 2020.

[179] X. Sun, H. Wang, and B. He, "Maban: Multi-agent boundary-aware network for natural language moment retrieval," *IEEE TIP*, vol. 30, 2021.

[180] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *ICCV*, 2017.

[181] H. Jiang and Y. Mu, "Joint video summarization and moment localization by cross-task sample transfer," in *CVPR*, 2022.

[182] M. Patrick, P.-Y. Huang, Y. Asano, F. Metze, A. G. Hauptmann, J. F. Henriques, and A. Vedaldi, "Support-set bottlenecks for video-text representation learning," in *ICLR*, 2021.

[183] F. Shi, L. Wang, and W. Huang, "End-to-end dense video grounding via parallel regression," *ArXiv*, vol. abs/2109.11265, 2021.

[184] X. Jiang, X. Xu, J. Zhang, F. Shen, Z. Cao, and X. Cai, "Gtlr: Graph-based transformer with language reconstruction for video paragraph grounding," in *ICME*, 2022.

[185] X. Jiang, X. Xu, J. Zhang, F. Shen, Z. Cao, and H. T. Shen, "Semi-supervised video paragraph grounding with contrastive encoder," in *CVPR*, 2022.

[186] X. Yang, S. Wang, J. Dong, J. Dong, M. Wang, and T.-S. Chua, "Video moment retrieval with cross-modal neural architecture search," *IEEE TIP*, vol. 31, 2022.

[187] M. Cao, T. Yang, J. Weng, C. Zhang, J. Wang, and Y. Zou, "Locvtp: Video-text pre-training for temporal localization," in *ECCV*, 2022.

[188] H. S. Nawaz, Z. Shi, Y. Gan, A. Hirpa, J. Dong, and H. Zheng, "Temporal moment localization via natural language by utilizing video question answers as a special variant and bypassing nlp for corpora," *IEEE TCSVT*, vol. 32, 2022.

[189] Y. Zhang, F. Niu, Q. Ping, and G. Thattai, "A multi-level alignment training scheme for video-and-language grounding," *ArXiv*, vol. abs/2204.10938, 2022.

[190] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[191] S. Woo, J. Park, I. Koo, S. Lee, M. Jeong, and C. Kim, "Explore-and-match: Bridging proposal-based and proposal-free with transformer for sentence grounding in videos," *ArXiv*, vol. abs/2201.10168, 2022.

[192] Y. Liu, S. Li, Y. Wu, C. W. Chen, Y. Shan, and X. Qie, "Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection," in *CVPR*, 2022.

[193] Z. Chen, L. Ma, W. Luo, P. Tang, and K.-Y. K. Wong, "Look closer to ground better: Weakly-supervised temporal grounding of sentence in video," *ArXiv*, vol. abs/2001.09308, 2020.

[194] M. Ma, S. Yoon, J. Kim, Y. Lee, S. Kang, and C. D. Yoo, "Vlanet: Video-language alignment network for weakly-supervised video moment retrieval," in *ECCV*, 2020.

[195] J. Wu, G. Li, X. Han, and L. Lin, "Reinforcement learning for weakly supervised temporal grounding of natural language in untrimmed videos," in *ACM MM*, 2020.

[196] Z. Zhang, Z. Zhao, Z. Lin, j. zhu, and X. He, "Counterfactual contrastive learning for weakly-supervised vision-language grounding," in *NeurIPS*, vol. 33, 2020.

[197] C. Da, Y. Zhang, Y. Zheng, P. Pan, Y. Xu, and C. Pan, "Asynce: Disentangling false-positives for weakly-supervised video grounding," in *ACM MM*, 2021.

[198] Z. Wang, J. Chen, and Y.-G. Jiang, "Visual co-occurrence alignment learning for weakly-supervised video moment retrieval," in *ACM MM*, 2021.

[199] Y. Wang, W. Zhou, and H. Li, "Fine-grained semantic alignment network for weakly supervised temporal language grounding," in *Findings of EMNLP*, 2021.

[200] J. Huang, Y. Liu, S. Gong, and H. Jin, "Cross-sentence temporal and semantic relations in video activity localisation," in *ICCV*, 2021.

[201] W. Yang, T. Zhang, Y. Zhang, and F. Wu, "Local correspondence network for weakly supervised temporal sentence grounding," *IEEE TIP*, vol. 30, 2021.

[202] J. Teng, X. Lu, Y. Gong, X. Liu, X. Nie, and Y. Yin, "Regularized two granularity loss function for weakly supervised video moment retrieval," *IEEE TMM*, 2021.

[203] Y. Wang, J. Deng, W. Zhou, and H. Li, "Weakly supervised temporal adjacent network for language grounding," *IEEE TMM*, 2021.

[204] R. Tan, H. Xu, K. Saenko, and B. A. Plummer, "Logan: Latent graph co-attention network for weakly-supervised video moment retrieval," in *WACV*, 2021.

[205] J. Chen, W. Luo, W. Zhang, and L. Ma, "Explore inter-contrast between videos via composition for weakly supervised temporal sentence grounding," in *AAAI*, 2022.

[206] S. Mo, D. Liu, and W. Hu, "Multi-scale self-contrastive learning with hard negative mining for weakly-supervised query-based video grounding," *ArXiv*, vol. abs/2203.03838, 2022.

[207] Y. Wang, M. Liu, Y. Wei, Z. Cheng, Y. Wang, and L. Nie, "Siamese alignment network for weakly supervised video moment retrieval," *IEEE TMM*, 2022.

[208] T. Han, K. Wang, J. Yu, and J. Fan, "Weakly supervised moment localization with natural language based on semantic reconstruction," *Image and Vision Computing*, vol. 126, 2022.

[209] Y. Song, J. Wang, L. Ma, Z. Yu, and J. Yu, "Weakly-supervised multi-level attentional reconstruction network for grounding textual queries in videos," *ArXiv*, vol. abs/2003.07048, 2020.

[210] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *ICCV*, 2019.

[211] S. Chen, "Towards bridging video and language by caption generation and sentence localization," in *ACM MM*, 2021.

[212] M. Zheng, Y. Huang, Q. Chen, Y. Peng, and Y. Liu, "Weakly supervised temporal sentence grounding with gaussian-based contrastive proposal learning," in *CVPR*, 2022.

[213] M. Zheng, Y. Huang, Q. Chen, and Y. Liu, "Weakly supervised video moment localization with contrastive negative sample mining," in *AAAI*, 2022.

[214] Z. Zhang, Z. Lin, Z. Zhao, J. Zhu, and X. He, "Regularized two-branch proposal networks for weakly-supervised moment retrieval in videos," in *ACM MM*, 2020.

[215] F. Luo, S. Chen, J. Chen, Z. Wu, and Y.-G. Jiang, "Self-supervised learning for semi-supervised temporal language grounding," *ArXiv*, vol. abs/2109.11475, 2021.

[216] J. Nam, D. Ahn, D. Kang, S. J. Ha, and J. Choi, "Zero-shot natural language video localization," in *ICCV*, 2021.

[217] J. Gao and C. Xu, "Learning video moment retrieval without a single annotated video," *IEEE TCSVT*, 2021.

[218] D. Liu, X. Qu, Y. Wang, X. Di, K. Zou, Y. Cheng, Z. Xu, and P. Zhou, "Unsupervised temporal video grounding with deep semantic clustering," in *AAAI*, 2022.

[219] S. Paul, N. C. Mithun, and A. K. Roy-Chowdhury, "Text-based temporal localization of novel events," *ArXiv*, 2022.

[220] Z. Xu, K. Wei, X. Yang, and C. Deng, "Point-supervised video temporal grounding," *IEEE TMM*, 2022.

[221] R. Cui, T. Qian, P. Peng, E. Daskalaki, J. Chen, X.-W. Guo, H. Sun, and Y.-G. Jiang, "Video moment retrieval from text queries via single frame annotation," in *SIGIR*, 2022.

[222] D. Li, R. Wu, Y. Tang, Z. Zhang, and W. Zhang, "Multi-scale 2d representation learning for weakly-supervised moment retrieval," in *ICPR*, 2021.

[223] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI*, 2017.

[224] M. Otani, Y. Nakahima, E. Rahtu, and J. Heikkilä, "Uncovering hidden challenges in query-based video moment retrieval," in *BMVC*, 2020.

[225] H. Zhou, C. Zhang, Y. Luo, C. Hu, and W. Zhang, "Thinking inside uncertainty: Interest moment perception for diverse temporal grounding," *IEEE TCSVT*, vol. 32, 2022.

[226] X. Yang, F. Feng, W. Ji, M. Wang, and T.-S. Chua, "Deconfounded video moment retrieval with causal intervention," in *SIGIR*, 2021.

[227] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, "Towards debiasing temporal sentence grounding in video," *ArXiv preprint arXiv:2111.04321*, vol. abs/2111.04321, 2021.

[228] J. Lei, T. L. Berg, and M. Bansal, "Qvhighlights: Detecting moments and highlights in videos via natural language queries," in *NeurIPS*, 2021.

[229] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[230] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," in *ICLR*, 2020.

[231] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[232] L. Zhu and Y. Yang, "Actbert: Learning global-local video-text representations," in *CVPR*, 2020.

[233] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, Y.-G. Jiang, L. Zhou, and L. Yuan, "Bevt: Bert pretraining of video transformers," *ArXiv*, vol. abs/2112.01529, 2021.

[234] J. Lei, L. Li, L. Zhou, Z. Gan, T. L. Berg, M. Bansal, and J. Liu, "Less is more: Clipbert for video-and-language learningvia sparse sampling," in *CVPR*, 2021.

[235] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer, "Videoclip: Contrastive pre-training for zero-shot video-text understanding," *ArXiv*, vol. abs/2109.14084, 2021.

[236] H. Xu, G. Ghosh, P.-Y. Huang, P. Arora, M. Aminzadeh, C. Feichtenhofer, F. Metze, and L. Zettlemoyer, "Vlm: Task-agnostic video-language model pre-training for video understanding," in *ACL Findings*, 2021.

[237] R. Zellers, X. Lu, J. Hessel, Y. Yu, J. S. Park, J. Cao, A. Farhadi, and Y. Choi, "Merlot: Multimodal neural script knowledge models," in *NeurIPS*, 2021.

[238] D.-A. Huang, S. Buch, L. Dery, A. Garg, L. Fei-Fei, and J. C. Niebles, "Finding "it": Weakly-supervised reference-aware visual grounding in instructional videos," in *CVPR*, 2018.

[239] J. Shi, J. Xu, B. Gong, and C. Xu, "Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses," in *CVPR*, 2019.

[240] Z. Chen, L. Ma, W. Luo, and K.-Y. K. Wong, "Weakly-supervised spatio-temporally grounding natural sentence in video," in *ACL*, 2019.

[241] J. Chen, W. Bao, and Y. Kong, "Activity-driven weakly-supervised spatio-temporal grounding from untrimmed videos," in *ACM MM*, 2020.

[242] A. Sadhu, K. Chen, and R. Nevatia, "Video object grounding using semantic roles in language description," in *CVPR*, 2020.

[243] Z. Zhang, Z. Zhao, Y. Zhao, Q. Wang, H. Liu, and L. Gao, "Where does it exist: Spatio-temporal video grounding for multi-form sentences," in *CVPR*, 2020.

[244] Z. Zhang, Z. Zhao, Z. Lin, B. Huai, and J. Yuan, "Object-aware multi-branch relation networks for spatio-temporal video grounding," in *IJCAI*, 2020.

[245] K. Shen, L. Wu, F. Xu, S. Tang, J. Xiao, and Y. Zhuang, "Hierarchical attention based spatial-temporal graph-to-sequence learning for grounded video description," in *IJCAI*, 2020.

[246] Q. Feng, Y. Wei, M. Cheng, and Y. Yang, "Decoupled spatial temporal graphs for generic visual grounding," *ArXiv*, vol. abs/2103.10191, 2021.

[247] Z. Tang, Y. Liao, S. Liu, G. Li, X. Jin, H. Jiang, Q. Yu, and D. Xu, "Human-centric spatio-temporal video grounding with visual transformers," *IEEE TCSVT*, 2021.

[248] R. Tan, B. Plummer, K. Saenko, H. Jin, and B. Russell, "Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos," in *NeurIPS*, vol. 34, 2021.

[249] R. Su, Q. Yu, and D. Xu, "Stvgbert: A visual-linguistic transformer based framework for spatio-temporal video grounding," in *ICCV*, 2021.

[250] M. Cao, J. Jiang, L. Chen, and Y. Zou, "Correspondence matters for video referring expression comprehension," in *ACM MM*, 2022.

[251] Y. Li, J. Yu, Z. Cai, and Y. Pan, "Cross-modal target retrieval for tracking by natural language," in *CVPR Workshops*, 2022.

[252] M. Li, T. Wang, H. Zhang, S. Zhang, Z. Zhao, J. Miao, W. Zhang, W. Tan, J. Wang, P. Wang, S. Pu, and F. Wu, "End-to-end modeling via information tree for one-shot natural language spatial video grounding," in *ACL*, 2022.

[253] Z. Xiong, D. Liu, and P. Zhou, "Gaussian kernel-based cross modal network for spatio-temporal video grounding," *ArXiv*, vol. abs/2207.00744, 2022.

[254] Z. Lin, C. Tan, J. Hu, Z. Jin, T. Ye, and W. Zheng, "Stvgformer: Spatio-temporal video grounding with static-dynamic cross-modal understanding," in *ACM MM Workshop*, 2022.

[255] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "Tubedetr: Spatio-temporal video grounding with transformers," in *CVPR*, 2022.

[256] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu, "Audio-visual event localization in unconstrained videos," in *ECCV*, 2018.

[257] Y. Wu, L. Zhu, Y. Yan, and Y. Yang, "Dual attention matching for audio-visual event localization," in *ICCV*, 2019.

[258] H. Xu, R. Zeng, Q. Wu, M. Tan, and C. Gan, "Cross-modal relation-aware networks for audio-visual event localization," in *ACM MM*, 2020.

[259] H. Xuan, Z. Zhang, S. Chen, J. Yang, and Y. Yan, "Cross-modal attention network for temporal inconsistent audio-visual event localization," in *AAAI*, vol. 34, 2020.

[260] B. Duan, H. Tang, W. Wang, Z. Zong, G. Yang, and Y. Yan, "Audio-visual event localization via recursive fusion by joint co-attention," in *WACV*, 2021.

[261] H. Xuan, L. Luo, Z. Zhang, J. Yang, and Y. Yan, "Discriminative cross-modality attention network for temporal inconsistent audio-visual event localization," *IEEE TIP*, vol. 30, 2021.

[262] C. Xue, X. Zhong, M. Cai, H. Chen, and W. Wang, "Audio-visual event localization by learning spatial and semantic co-attention," *IEEE TMM*, 2021.

[263] Y. Xia, Z. Zhao, S. Ye, Y. Zhao, H. Li, and Y. Ren, "Video-guided curriculum learning for spoken video grounding," in *ACM MM*, 2022.

[264] N. Garcia and G. Vogiatzis, "Asymmetric spatio-temporal embeddings for large-scale image-to-video retrieval," in *BMVC*, 2018.

[265] Z. Zhang, Z. Zhao, Z. Lin, J. Song, and D. Cai, "Localizing unseen activities in video via image query," in *IJCAI*, 2019.

[266] R. Xu, L. Niu, J. Zhang, and L. Zhang, "A proposal-based approach for activity image-to-video retrieval," in *AAAI*, vol. 34, 2020.

[267] L. Liu, J. Li, L. Niu, R. Xu, and L. Zhang, "Activity image-to-video retrieval by disentangling appearance and motion," in *AAAI*, vol. 35, 2021.

[268] Y. Feng, L. Ma, W. Liu, T. Zhang, and J. Luo, "Video re-localization," in *ECCV*, 2018.

[269] Y. Feng, L. Ma, W. Liu, and J. Luo, "Spatio-temporal video re-localization by warp lstm," in *CVPR*, 2019.

[270] Y.-H. Huang, K.-J. Hsu, S.-K. Jeng, and Y.-Y. Lin, "Weakly-supervised video re-localization with multiscale attention model," in *AAAI*, vol. 34, 2020.

[271] C. Jiang, K. Huang, S. He, X. Yang, W. Zhang, X. Zhang, Y. Cheng, L. Yang, Q. Wang, F. Xu, T. Pan, and W. Chu, "Learning segment similarity and alignment in large-scale content based video retrieval," in *ACM MM*, 2021.

[272] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[273] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," *IEEE TPAMI*, 2018.

[274] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel, "Racial disparities in automated speech recognition," *National Academy of Sciences*, vol. 117, 2020.

[275] H. Huang, F. Xue, H. Wang, and Y. Wang, "Deep graph random process for relational-thinking-based speech recognition," in *ICML*, vol. 119, 2020.

[276] V. Escorcia, M. Soldan, J. Sivic, B. Ghanem, and B. Russell, "Temporal localization of moments in video collections with natural language," *ArXiv*, vol. abs/1907.12763, 2019.

[277] J. Lei, L. Yu, T. L. Berg, and M. Bansal, "Tvr: A large-scale dataset for video-subtitle moment retrieval," in *ECCV*, 2020.

[278] J. Lei, T. Berg, and M. Bansal, "mTVR: Multilingual moment retrieval in videos," in *ACL*, 2021.

[279] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "HERO: Hierarchical encoder for Video+Language omni-representation pre-training," in *EMNLP*, 2020.

[280] B. Zhang, H. Hu, J. Lee, M. Zhao, S. Chammas, V. Jain, E. Ie, and F. Sha, "A hierarchical multi-modal encoder for moment localization in video corpus," *ArXiv*, vol. abs/2011.09046, 2020.

[281] H. Zhang, A. Sun, W. Jing, G. Nan, L. Zhen, J. T. Zhou, and R. S. M. Goh, "Video corpus moment retrieval with contrastive learning," in *SIGIR*, 2021.

[282] S. Maeoki, Y. Mukuta, and T. Harada, "Video moment retrieval with text query considering many-to-many correspondence using potentially relevant pair," *ArXiv*, vol. abs/2106.13566, 2021.

[283] S. Paul, N. C. Mithun, and A. K. Roy-Chowdhury, "Text-based localization of moments in a video corpus," *IEEE TIP*, vol. 30, 2021.

[284] Z. Hou, C.-W. Ngo, and W. K. Chan, "Conquer: Contextual query-aware ranking for video corpus moment retrieval," in *ACM MM*, 2021.

[285] Z. Gao, H. Liu, and J. Liu, "Coarse to fine: Video retrieval before moment localization," *ArXiv*, vol. abs/2110.07201, 2021.

[286] S. Yoon, D. Kim, J. Kim, and C. D. Yoo, "Cascaded mpn: Cascaded moment proposal network for video corpus moment retrieval," *IEEE Access*, vol. 10, 2022.

[287] J. Liu, T. Yu, H. Peng, M. Sun, and P. Li, "Cross-lingual cross-modal consolidation for effective multilingual video corpus moment retrieval," in *Findings of NAACL*, 2022.

[288] D. Kim, S. Yoon, J. W. Hong, and C. D. Yoo, "Semantic association network for video corpus moment retrieval," in *ICASSP*, 2022.

[289] X. Sun, X. Long, D. He, S. Wen, and Z. Lian, "Vsrnet: End-to-end video segment retrieval with text query," *Pattern Recognition*, vol. 119, 2021.

[290] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *ACM MM*, 2014.

[291] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan, "Visual grounding via accumulated attention," in *CVPR*, 2018.

[292] S. Yang, G. Li, and Y. Yu, "Dynamic graph attention for referring expression comprehension," in *ICCV*, 2019.

[293] J. Deng, Z. Yang, T. Chen, W. Zhou, and H. Li, "Transvg: End-to-end visual grounding with transformers," in *ICCV*, 2021.

[294] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *CVPR*, 2016.

[295] X. Li, F. Zhou, C. Xu, J. Ji, and G. Yang, "Sea: Sentence encoder assembly for video retrieval by textual queries," *IEEE TMM*, vol. 23, 2021.

[296] Z. Wang, Y. Wu, K. Narasimhan, and O. Russakovsky, "Multi-query video retrieval," in *ECCV*, 2022.

[297] X. Wang, L. Zhu, Z. Zheng, M. Xu, and Y. Yang, "Align and tell: Boosting text-video retrieval with local alignment and fine-grained supervision," *IEEE TMM*, 2022.

[298] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim, "TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering," in *CVPR*, 2017.

[299] J. Lei, L. Yu, M. Bansal, and T. Berg, "TVQA: Localized, compositional video question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[300] J. Gao, R. Ge, K. Chen, and R. Nevatia, "Motion-appearance co-memory networks for video question answering," in *CVPR*, 2018.

[301] J. Liang, l. Jiang, L. Cao, Y. Kalantidis, L. Li, and A. G. Hauptmann, "Focal visual-text attention for memex question answering," *IEEE TPAMI*, 2019.

[302] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao, "Activitynet-qa: A dataset for understanding complex web videos via question answering," in *AAAI*, 2019.

[303] H. Le, D. Sahoo, N. Chen, and S. Hoi, "Multimodal transformer networks for end-to-end video-grounded dialogue systems," in *ACL*, 2019.

[304] J. Lei, L. Yu, T. Berg, and M. Bansal, "TVQA+: Spatio-temporal grounding for video question answering," in *ACL*, 2020.

[305] J. Kim, M. Ma, T. Pham, K. Kim, and C. D. Yoo, "Modality shifting attention network for multi-modal video question answering," in *CVPR*, 2020.

[306] S. Kim, S. Jeong, E. Kim, I. Kang, and N. Kwak, "Self-supervised pre-training and contrastive representation learning for multiple-choice video qa," in *AAAI*, 2021.

[307] R. Pasunuru and M. Bansal, "Game-based video-context dialogue," *ArXiv*, vol. abs/1809.04560, 2018.

[308] ——, "Dstc7-avsd: Scene-aware video-dialogue systems with dual attention," in *AAAI workshop*, 2019.

[309] H. Le, N. Chen, and S. Hoi, "Vgnmn: Video-grounded neural module networks for video-grounded dialogue systems," in *NAACL*, 2022.

APPENDIX A
SUPPLEMENTARY MATERIALS

We present the following content as supplementary material: (i) the efficiency comparison among different method categories, and (ii) the comparison of TSGV and other video-language tasks.

A.1 Efficiency Comparison

In addition to the reported performance overview, we also provide an empirical efficiency comparison among different categories of methods, such as training and test time. It is infeasible to compile and conduct efficiency evaluation for all TSGV models due to the computation resource and time constraints. Meanwhile, some methods do not make their codes publicly available. Hence, we select one or two representative models from each category for efficiency comparison. The purpose is to provide a glimpse of the efficiency of the different *categories* of methods, rather than a detailed comparison among all methods.

To be specific, we choose CTRL [9] to represent sliding window-based method (SW), LPNet [57] for proposal-generated method (PG), SCDM [17] for the standard anchor-based method (AN), 2D-TAN [18] for 2D-Map anchor-based method (2D), ABLR [19] for regression-based method (RG), VSLNet [23] for span-based method (SN), and RWM-RL [24] to represent reinforcement learning-based method (RL). Note that for the PG category, we choose LPNet [57] instead of the two early PG methods QSPN [53] and SAP [54]. QSPN is implemented in Caffe [290], which is difficult for us to compile under our preset environment. SAP does not release its source codes.

The ideal setting for model efficiency evaluation is to run all selected models on a benchmark dataset. However, we observe that the models process data in very different ways, such as different video/text features or different feature sequence lengths. In fact, there is no one common benchmark dataset on which all the selected models have been evaluated. For efficiency comparison, we choose to mock feature inputs to the models via random initialization *i.e.,* not relying on an existing benchmark dataset. Specifically, we set the batch size to be 16, video sequence length 256, video feature dimension $1,024$, word sequence length 30, and word feature dimension 300. Then, the sizes of virtual video and text feature inputs for each batch are $16 \times 256 \times 4096$ and $16 \times 30 \times 300$, respectively. We follow the hyperparameters listed in the corresponding paper or code repository for other model settings. It is worth noting that the feature pre-load time is the same for all these methods due to the same mock features being utilized, so we do not count the data processing time in the evaluation. We run 200 steps for model training and testing to calculate the execution time. The hyperparameters of the mocked inputs and train/test steps are summarized in Table 8. For a fair comparison, all experiments are conducted on a single NVIDIA Tesla V100 GPU with 32GB memory.

The results of the efficiency comparison are summarized in Table 9. Under the same environment and feature input settings, the SN category achieves the highest efficiency, SW and RL

TABLE 8
The hyperparameters of feature input simulation for TSGV models, where L_{video} is video sequence length, d_{video} is video feature dimension, L_{text} is word sequence length, and d_{text} is word feature dimension.

Batch Size	L_{video}	d_{video}	L_{text}	d_{text}	Train/Test Steps
16	256	$1,024$	30	300	200

TABLE 9
Efficiency comparison among the select TSGV models from different method categories, where T_{train} and T_{test} represent the total training and testing time in seconds.

Category	Model	Backend	T_{train}	T_{test}
SW	CTRL [9]	TensorFlow	1732.51	3013.42
PG	LPNet [57]	TensorFlow	113.45	42.98
AN	SCDM [17]	TensorFlow	61.99	279.01
2D	2D-TAN [18]	PyTorch	356.10	148.31
RG	ABLR [19]	TensorFlow	142.63	81.00
SN	VSLNet [23]	TensorFlow	35.45	32.38
RL	RWM-RL [24]	PyTorch	2214.92	3296.86

categories are the least efficient. PG, RG, 2D, and AN categories are in between these two extremes. For the PG category, LPNet utilizes VSLNet as the backbone to generate proposals and then selects a few top-ranked proposals for refinement to get the final prediction. Thus, LPNet is much faster than the SW-based method as well as the early PG-based solutions. For CTRL, SCDM, and RWM-RL, we observe their testing time is longer than the training time because these models only accept a single video as input each time by architecture design. In summary, the efficiency results of different method categories generally support our discussion on model efficiency in the main paper.

A.2 Comparison between TSGV and Other VL Tasks

In this section, we briefly discuss the relationships between TSGV and other vision-language (VL) tasks. Our goal is to better understand TSGV from different perspectives and the similarities and dissimilarities between TSGV and other VL tasks. We begin with the definition of TSGV: given an untrimmed video, TSGV is to retrieve a video segment, also known as a temporal moment, that semantically corresponds to a query in natural language *i.e.,* sentence. Fig. 28 provides an illustration.

A.2.1 TSGV versus Visual Grounding

Visual grounding (VG) [291]–[293] aims to locate the most relevant object or region in an image, based on a natural language query. The query can be a phrase, a sentence, or even a multi-round dialogue. Illustrated in Fig. 29, for a language query "a woman in black playing a game with her friends", VG needs to return the bounding box of the corresponding object from the image as the answer.

Based on the definitions of TSGV and VG, the only difference between the two tasks is the reference, *i.e.,* a video in TSGV and an image in VG. Solutions to VG [291]–[293] usually apply a pre-trained CNN model to extract features from the whole image and cropped regions, resulting in a sequence of visual features. The language query is encoded by pre-trained word embeddings or language models. After that, various cross-modal learning frameworks are designed to encode the multimodal interactions between image and query and to predict the target bounding boxes. In general, the overall processes of TSGV and VG are the same. However, TSGV focuses on learning the relationships between language query and the visual contents in a dynamic temporal sequence *i.e.,* detecting the temporal boundaries. In contrast, VG focuses on learning the relationships between the query and the visual contents in a static spatial region *i.e.,* detecting spatial bounding boxes. Moreover, as discussed in Section 6.2.3, STSGV is a task to sequentially localize the referring instances from a sequence of continuous frames in a video. In this case, STSGV could be regarded as a combined task of TSGV and VG. In summary, the relationships among TSGV, VG, and STSGV suggest that some VL tasks are related, and solutions could be shared among them to some extent.

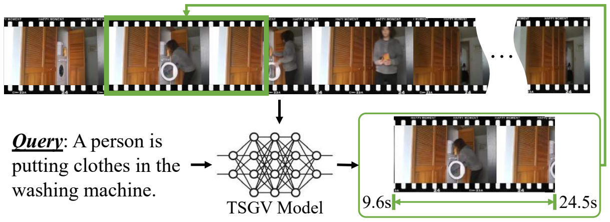


Fig. 28. An illustration of temporal sentence grounding in videos (TSGV).

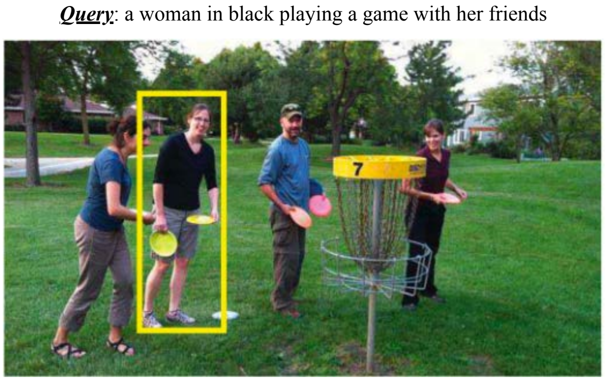


Fig. 29. An illustration of visual grounding (VG). Given a language query and an image, VG aims to localize the referential object (yellow box) from the image, where the answer bounding box contains the object described by the query.

A.2.2 TSGV versus Video Retrieval

Given a query and a set of candidate videos, video retrieval (VR) [294]–[297] is a task to retrieve and rank candidate videos by their relevance to the query. In general, queries for VR are not limited to text. Here we only consider the text-video retrieval scenario for its relevance to TSGV. As depicted in Fig. 30, given a language query "The man continues to pour more ingredients in and then puts it on a table.", VR retrieves the videos whose content matches the query description. The general procedure of VR is to conduct cross-modal reasoning between text query and video candidates and project them into a joint embedding space. Within the joint space, VR aims to reduce the distance of the matching video-query pairs and increase the distance of non-matching pairs.

In our comparison, VR is to retrieve a video from a set of videos, and TSGV is to localize a temporal segment within a single video. To some extent, VR is coarse-grained retrieval while TSGV is fine-grained retrieval. Thus, VR focuses more on the

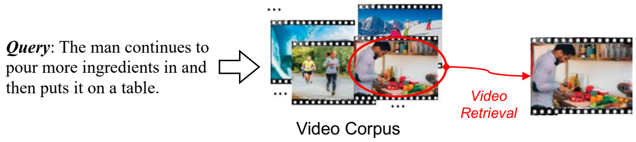

Query: The man continues to pour more ingredients in and then puts it on a table.
Video Corpus
Video Retrieval

Fig. 30. An illustration of text-based video retrieval (VR). Given a text query and a set of candidate videos, video retrieval (VR) retrieves and ranks candidate videos by their relevance to the query.

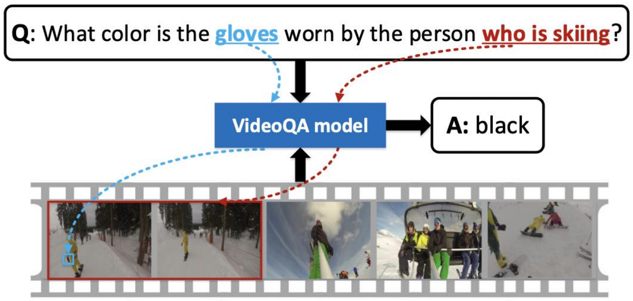

Q: What color is the gloves worn by the person who is skiing?
VideoQA model
A: black

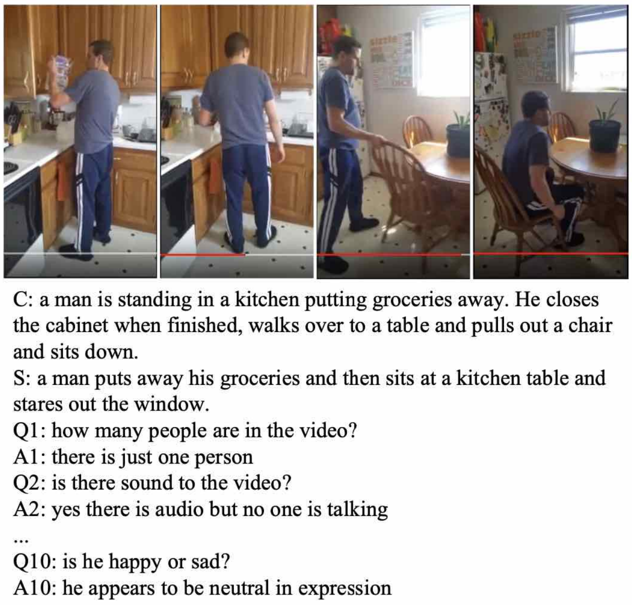
Fig. 31. An example of video question answering (VideoQA) from Yu *et al.* [302]. Given a text question, VideoQA needs to fully understand the fine-grained semantics of the question (*e.g.,* keywords) and to perform cross-modal reasoning on the visual contents (frames in the red border and objects in the blue box) to answer the question.

overall semantic knowledge of the query as well as the overall information of candidate videos. TSGV is expected to understand the fine-grained query information and the representations and relationships of different events in a video. For a query, VR usually retrieves target videos from thousands of video candidates, while TSGV only considers the interaction between the query and a single video. Thus, the fine-grained cross-modal reasoning between language query and video in TSGV could be inefficient or even infeasible for VR. However, if TSGV adopts sliding window-based or proposal generation methods, then the video is first decomposed into a set of proposal candidates. Treating the proposal candidates as a set of short videos, TSGV and VR become similar since both tasks aim to rank the best matching candidates among multiple proposals/videos, for a given language query. In this case, solutions to VR may be applicable to TSGV to some extent. Besides, VCMR (discussed in Section 6.2.5) is to retrieve a matching moment to a query from a collection of untrimmed and unsegmented videos; this task can be regarded as a combination of TSGV and VR.

A.2.3 TSGV versus Video Question Answering

Video question answering (VideoQA) [298]–[302] is to answer a question in text form, based on the events/objects contained in an input video. As shown in Fig. 31, given a question "What color are the gloves worn by the person who is skiing?", the VideoQA model needs to understand the key components of the question (*e.g.,* "gloves" and "skiing"), and to interact with the video to ground the events and/or objects mentioned in the question. Then, the model predicts the answer based on the retrieved events/objects.

VideoQA contains two reasoning steps. The first is to localize the contents relevant to the given question from the video. The second is to infer the answer based on the grounded contents. Since temporal grounding in the video is an indispensable component, TSGV could serve as an intermediate step in VideoQA.



C: a man is standing in a kitchen putting groceries away. He closes the cabinet when finished, walks over to a table and pulls out a chair and sits down.
S: a man puts away his groceries and then sits at a kitchen table and stares out the window.
Q1: how many people are in the video?
A1: there is just one person
Q2: is there sound to the video?
A2: yes there is audio but no one is talking
...
Q10: is he happy or sad?
A10: he appears to be neutral in expression

Fig. 32. An example of video grounded dialogue (VideoDial) from Le *et al.* [303], where C denotes video caption, S represents video summary, Q_i denotes the i-th turn question and A_i represents the i-th tun answer. The definition of VideoDial is that, given a video, a dialogue is conducted based on the visual and audio aspects of the given video.

As object grounding is also required for VideoQA in answer prediction, both visual grounding (VG) and STSGV can be applied here. In fact, several work [304]–[306] applies TSGV as an auxiliary component in VideoQA. For instance, Lei *et al.* [304] propose a spatio-temporal answerer with grounded evidence. They design a video-text fusion module followed by a span predictor to localize the temporal boundaries of moments relevant to the answer. Similarly, Kim *et al.* [305] deploy a moment proposal network to localize the required temporal moment of interest for question answering.

A.2.4 TSGV versus Video Grounded Dialogue

Video grounded dialogue (VideoDial) [303], [307]–[309] is to conduct a multi-turn conversation, based on the visual and audio aspects of a given video. Similar to VideoQA, VideoDial also requires moment localization in the video as an intermediate step to support answer generation. However, there are several differences between VideoQA and VideoDial. First, VideoQA is usually formulated as a multiple-choice problem, while VideoDial is a generation task. Second, VideoQA is a single-turn task while VideoDial consists of a multi-turn conversation. Thus, VideoDial is considered more challenging and it needs an in-depth understanding of the visual and/or audio contents. Meanwhile, VideoDial also requires continuous moment localization based on both the current utterance and conversation history. In general, TSGV is an indispensable component in VideoDial.