

Vid-Group: Temporal Video Grounding Pretraining from Unlabeled Videos in the Wild

Peijun Bao¹, Chenqi Kong^{1†}, Siyuan Yang¹, Zihao Shao²,
Xinghao Jiang³, Boon Poh Ng¹, Meng Hwa Er¹, Alex Kot^{1,4},
¹Nanyang Technological University ²Peking University
³Shanghai Jiaotong University ⁴Shenzhen MSU-BIT University
peijun001@e.ntu.edu.sg chenqi.kong@ntu.edu.sg

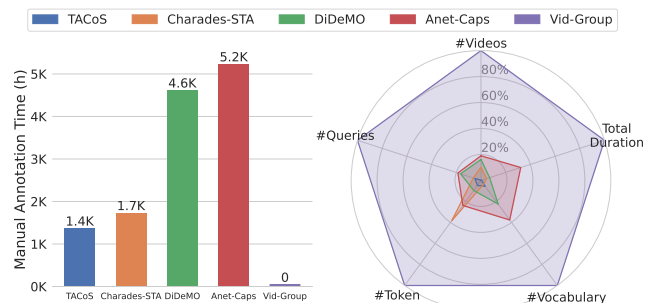
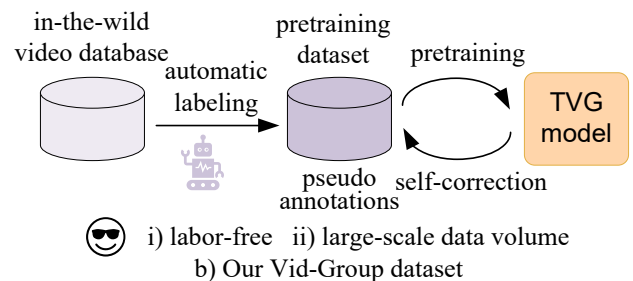
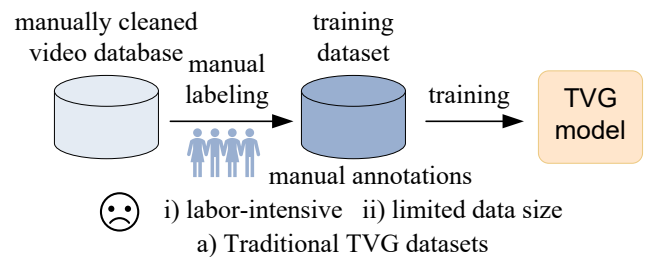
Abstract

Given a natural language query, temporal video grounding aims to localize the described temporal moment in an untrimmed video. A major challenge of this task is its heavy dependence on labor-intensive annotations for training. Unlike existing works that directly train models on manually curated data, we propose a novel paradigm to reduce annotation costs: pretraining the model on unlabeled, real-world videos. To support this, we introduce Temporal Video Grounding Pretraining (Vid-Group), a large-scale dataset collected in a scalable manner with minimal human intervention, consisting of over 50K videos captured in the wild and 200K pseudo annotations. Direct pretraining on these imperfect pseudo annotations, however, presents significant challenges, including mismatched sentence-video pairs and imprecise temporal boundaries. To address these issues, we propose the ReCorrect algorithm, which comprises two main phases: semantics-guided refinement and memory-consensus correction. The semantics-guided refinement enhances the pseudo labels by leveraging semantic similarity with video frames to clean out unpaired data and make initial adjustments to temporal boundaries. In the following memory-consensus correction phase, a memory bank tracks the model predictions, progressively correcting the temporal boundaries based on consensus within the memory. Comprehensive experiments demonstrate ReCorrect’s strong generalization abilities across multiple downstream settings. The code, dataset, and pretrained models are available at <https://github.com/baopj/Vid-Group>.

1. Introduction

Given a natural language query and an untrimmed video, the task of Temporal Video Grounding (TVG) [8, 15] aims to temporally localize the video moment described by the language query. TVG is one of the most fundamental tasks in video understanding and has a wide range of real-world

†: Corresponding author.



c) Comparison of manual annotation time and dataset size (dataset size metrics normalized by Vid-Group)

Figure 1. a) A crucial challenge in temporal video grounding is its reliance on massive datasets with labor-intensive annotations. b) To overcome this, we introduce a large-scale dataset for Temporal Video Grounding Pretraining (Vid-Group), collected in a scalable way with minimal human involvement. c) Compared to previous TVG datasets, Vid-Group achieves a substantially larger dataset size while maintaining zero manual annotation cost.

applications [28, 31, 49], such as video retrieval, video summarization, and video surveillance analysis. In recent years, the performance of TVG has been improved by deep learning techniques [3, 4, 20, 25, 33, 42, 44, 46] and the availability of manually annotated data [8, 10, 15]. However, as shown in Fig 1, collecting these manual annotations, which include sentence queries and temporal boundaries, remains both expensive and time-consuming. For instance, according to [40], manually annotating the ActivityNet Captions dataset required at least 5,200 hours, which is both impractical and unscalable in real-world settings. Additionally, manual annotations often exhibit language and temporal biases [16, 41], such as in query style and temporal boundary distribution, which limit practical applicability.

To this end, we introduce Temporal **Video Grounding Pretraining** (Vid-Group), a large-scale dataset collected in a scalable manner with minimal human intervention. Vid-Group leverages multimodal large language models, such as GPT-4o, to generate pseudo-annotations for unlabeled videos sourced from the wild. Compared to previous TVG datasets [8, 10, 15, 30], Vid-Group achieves a substantially larger dataset size, including the number of videos and annotations, while maintaining zero manual annotation cost. However, due to the minimal human involvement in creating these samples, directly pretraining on them poses significant challenges. Common issues include videos lacking meaningful activity, mismatched video-query pairs, and imprecise temporal boundaries (examples shown in Fig 4).

To address these issues, we propose the **Refinement and Correction** (ReCorrect) algorithm with a self-correction mechanism of pseudo annotations during pretraining. ReCorrect consists of two main phases: semantics-guided refinement and memory-consensus correction. In the semantics-guided refinement phase, we enhance the pseudo labels by leveraging semantic similarity between video frames and pseudo labels to clean error-prone training samples, such as idle videos and mismatched video-query pairs, while initially adjusting the temporal boundaries. In the subsequent memory-consensus correction phase, a memory bank continuously tracks the model’s predictions during pretraining. This memory bank then serves as a reference to progressively calibrate the temporal boundaries of the pseudo labels based on consensus within the memory. The pretrained ReCorrect model can then be seamlessly adapted to downstream tasks with limited manual labels, such as zero-shot inference and unsupervised finetuning.

Experimental results indicate that unsupervised ReCorrect achieves approximately **85%** of the state-of-the-art performance obtained by fully supervised methods on both the Charades-STA [8] and ActivityNet Captions [15] benchmarks. Moreover, the zero-shot variant of ReCorrect attains **75%** and **80%** of the fully supervised performance on these two datasets, respectively. This highlights the Vid-Group’s

potential to address the critical challenge of heavy reliance on manual annotations in TVG.

Our main contributions can be summarised as follows:

1. We introduce Vid-Group, a large-scale and diverse dataset containing over 50K videos captured in the wild and 200K pseudo training samples designed for pretraining in temporal video grounding.
2. To tackle the issues of error-prone pseudo training samples, we propose the ReCorrect algorithm. ReCorrect incorporates semantics guided refinement to clean and adjust pseudo labels and exploit memory consensus correction to calibrate temporal boundaries based on consensus within a memory bank.
3. Comprehensive experiments demonstrate ReCorrect’s state-of-the-art performance under both zero-shot and unsupervised settings.

2. Related Works

Fully-Supervised Temporal Video Grounding. The performance of fully supervised Temporal Video Grounding (TVG) has been improved by the advancement of deep learning techniques [2, 3, 6, 20, 25, 33, 42, 44, 46] and the availability of manually annotated data [8, 10, 15]. For instance, Liu *et al.* [20] propose applying attention mechanism to highlight the crucial part of visual features. Ding *et al.* [6] introduce support-set supervision as an additional regularization term to enhance fully supervised TVG. While achieving promising performance, these fully-supervised methods rely on the manual annotations, which are labor-intensive and subjective to label.

Unsupervised Temporal Video Grounding. To eliminate annotation costs, some recent works [7, 11, 13, 27, 32] investigate unsupervised approaches using only unlabeled videos from the existing manually-cleaned datasets such as Charades-STA and ActivityNet Captions. For example, Kim *et al.* [13] propose a language-free training algorithm to train the TVG model without language data. However, these models depend on manually cleaned, unscalable video data. In contrast, the proposed Vid-Group dataset is collected in a scalable manner without human intervention, and our ReCorrect framework is specifically designed to rely solely on unlabeled videos captured in the wild.

Zero-Shot Temporal Video Grounding. Luo *et al.* [22] explores the first zero-shot TVG approach by designing training-free modules to adapt off-the-shelf vision language model CLIP [29] to TVG tasks. A list of further works [38, 39, 48] further investigates adapting other vision language models, such as BLIP-2 [17] to zero-shot TVG in a training-free manner. Recently, the task of TVG has also drawn increasing attention from video large language models. Several works [9, 12, 36], such as VTimeLLM, propose time-aware instruction tuning specifically to enhance the zero-shot TVG capability of video large language model.

Table 1. Statistics of temporal video grounding datasets (training split), where the annotation time for each dataset is estimated by [40].

Dataset	Annotation Time (h)	Videos		Language Queries					
		Video Numbers	Total Duration	Query Numbers	Total Tokens	Vocabulary			
						Adj.	Nouns	Verbs	Total
TACoS [30]	1.4K	0.1K	4.7h	9.8K	0.1M	0.3K	0.8K	0.7K	1.7K
Charades-STA [8]	1.7K	5.3K	45.8h	12.4K	0.8M	0.1K	0.6K	0.4K	1.1K
DiDeMO [10]	4.6K	8.5K	70.9h	33.0K	0.2M	1.3K	4.8K	2.4K	7.0K
Anet-Captions [15]	5.2K	10.0K	326.1h	37.4K	0.5M	2.1K	7.9K	4.0K	11.7K
Vid-Group (Ours)	Labor-Free	52.7K	1013.2h	200.3K	2.1M	6.5K	21.0K	6.6K	31.1K

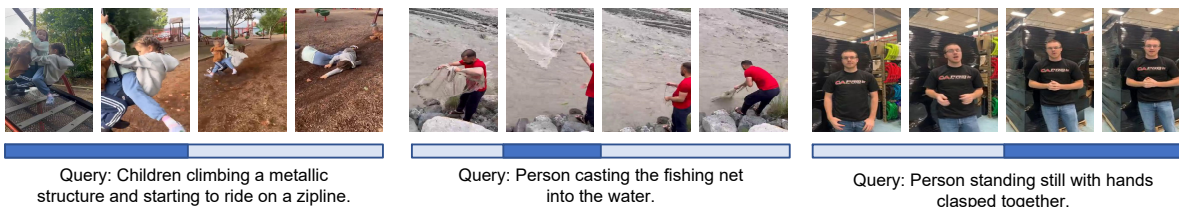


Figure 2. Illustration of video samples and pseudo-annotations, including sentence queries and temporal boundaries, from the Temporal Video Grounding Pretraining (Vid-Group) dataset. The dark blue box represents the temporal boundary of the described video moment.

Pseudo Boundary Refinement in TVG. Several zero-shot and unsupervised TVG explore enhance the pseudo temporal boundaries by computing the similarity between text and video proposals using frozen vision-language models. For instance, Luo *et al.* [22] and Zheng *et al.* [48] propose to generate more accurate temporal boundaries through similarity computation via CLIP [29] and BLIP-2 [17], respectively. Similarly, VTG-GPT introduces a proposal generator and scorer to refine pseudo boundaries based on visual-semantic similarities in zero-shot TVG. Unlike these methods which statically refine pseudo labels, our Reccorrect leverages a self-correction mechanism to adaptively improve pseudo temporal boundaries based on the memory consensus.

Temporal Video Grounding Pretraining. Most existing works for temporal video grounding pretraining [5, 34, 37] focus on pretraining the feature extraction backbone for TVG. Our work is *orthogonal* to these previous approaches: while their approach focuses on *pretraining feature extraction backbone*, ours targets pretraining the TVG model itself with a *fixed feature backbone*. This distinction is further emphasized by the fact that our model supports zero-shot setting without additional finetuning, whereas theirs cannot. However, these methods directly use pseudo annotations as ground truth without label correction during pretraining. As a result, their TVG performance remains significantly lower than most zero-shot methods, including off-the-shelf vision-language models such as Luo *et al.* [22] (see Table 2).

3. Vid-Group Dataset

Overview. Temporal video grounding (TVG) [8, 15] aims to temporally identify the video moment in an untrimmed

video as described by a language query. Although fully-supervised methods achieve promising performance, the high cost of annotation still limits its practical application.

To this end, as presented in Fig 2, we introduce Temporal Video Grounding Pretraining (Vid-Group), a large-scale dataset containing over 50K videos captured in the wild and 200K training annotations, collected efficiently through an automated process with minimal human involvement. As summarized in Table 1, Vid-Group contains five times the number of videos and queries compared to the previous largest dataset, ActivityNet Captions, while maintaining zero manual annotation cost. In contrast, ActivityNet Captions and DiDeMo require 5.2K and 4.6K hours, respectively, for manual labeling, which is unscalable and impractical for real-world applications. Moreover, Vid-Group encompasses rich semantic content, including diverse activities across various visual domains (samples can be found in the supplementary material).

Dataset Construction. To collect the videos, we define a list of target activities and use web crawling on YouTube to gather videos up to a maximum duration of t_{max} . Each video is uniformly sampled into n_{v2f} frames, and these frames are concatenated to form a composite image. We employ a multimodal language model (MLLM), such as GPT-4o, to generate pseudo labels, guided by a carefully designed prompt that instructs the MLLM to produce descriptive sentences with associated frame indices matching the image content. These frame indices are then mapped to start and end timestamps within each video. To this end, this process is designed to be highly scalable, with minimal manual intervention primarily limited to defining search keywords and designing the prompt. More details of dataset construction

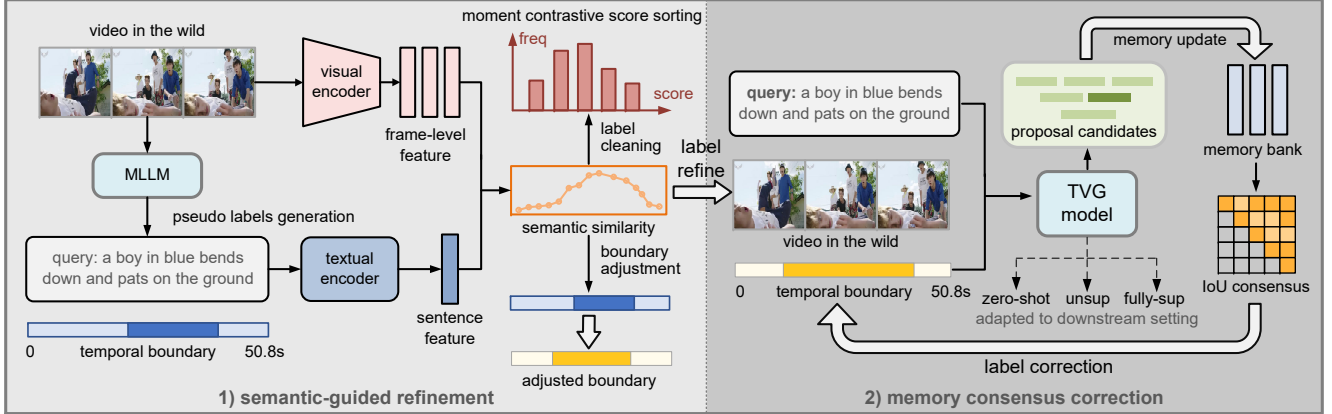


Figure 3. Overview of the Refinement and Correction (ReCorrect) algorithm for temporal video grounding pretraining from in-the-wild videos. ReCorrect consists of two key phases: 1) semantics-guided refinement, which leverages semantic similarity to clean noisy pseudo training samples, such as idle videos and unmatched video-query pairs, while initially adjusting temporal boundaries, and 2) memory-consensus correction, where a memory bank tracks model predictions, progressively correcting temporal boundaries based on consensus within the memory.

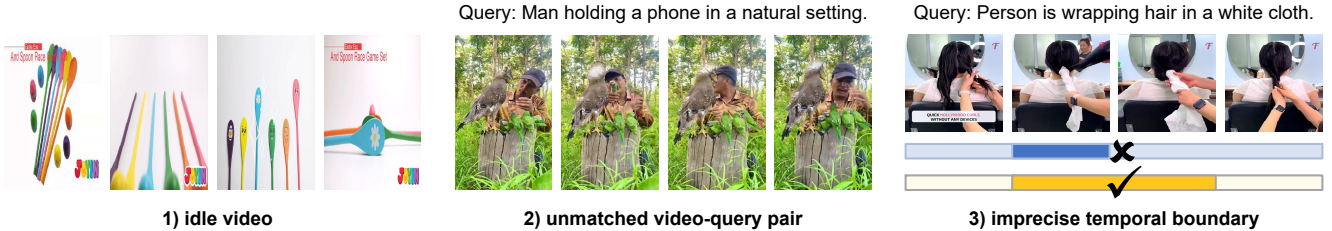


Figure 4. Collected in a scalable, labor-free manner, the Vid-Group dataset exhibits three common errors in pseudo-labeled training samples: 1) idle videos lacking meaningful activity, 2) unmatched video-query pairs where the query event does not appear in the video, and 3) imprecise temporal boundaries where video-query matches are correct but temporal boundaries are inaccurate.

can be found in the supplement. The complete dataset is released at <https://github.com/baopj/Vid-Group>.

4. ReCorrect Algorithm

Since videos captured in the wild are inherently unclean, and MLLM’s limitations in labeling accuracy introduce additional noise, there are widespread errors in the pseudo annotations of the Vid-Group Dataset, as shown in Fig. 4. These errors fall into three main categories: 1) Idle videos that lack any meaningful events. 2) Unmatched video-query pairs where the pseudo queries do not correspond to any video frames. 3) Imprecise temporal boundaries where the query matches the video, but the temporal alignment is inaccurate. These errors present significant challenges for direct pretraining on the pseudo training samples. To deal with these issues, as illustrated in Fig. 3, we propose the Refinement and Correction (ReCorrect) algorithm. It consists of 1) semantics-guided refinement, which removes erroneous training samples and initially adjusts temporal boundaries, and 2) memory-consensus correction, where a memory bank tracks predictions to correct boundaries based

on consensus.

4.1. Pretraining on Vid-Group

Semantics-Guided Refinement. To ensure that the pseudo query Q are aligned with the video moment in the untrimmed video, we propose a semantics guided refinement to clean out the unmatched pair of video and sentence, as well as adjust the pseudo temporal boundary. We first extract the query feature q and the visual feature v_t for t -th frame with pretrained CLIP model [29], and compute the semantic similarities s_t between them, formulated as:

$$s_t = \frac{q^\top v_t}{\|q\| \cdot \|v_t\|}, \quad t = 1 \dots T \quad (1)$$

where T is the total number of frames in the video.

Let $b = (\tau_s, \tau_e)$ denote the pseudo-temporal boundaries provided by the MLLM for the query, where τ_s and τ_e represent the start and end timepoints, respectively. Then, we compute the moment contrastive scores, which indicate the contrastive semantic relevance of the video content to the sentence, comparing the content inside the pseudo-temporal

moment versus outside it, formulated as:

$$\gamma(\tau_s, \tau_e) = \frac{\sum_{t=\tau_s}^{\tau_e} s_t}{\sum_{t=1}^{\tau_s-1} s_t + \sum_{t=\tau_t+1}^T s_t} \quad (2)$$

A high value of $\gamma(\tau_s, \tau_e)$ indicates strong relevance between the pseudo query and the video moment defined by the temporal boundaries. We sort the moment contrastive score γ for each data sample in descending order, cleaning out the bottom R percent and selecting only the remaining as training samples. Subsequently, we adjust the pseudo temporal boundary by either shrinking or expanding the start time τ_s based on the semantic similarity s_t . Specifically, if $\gamma(\tau_s, \tau_e) < \alpha_1 \cdot \gamma(\tau'_s, \tau_s)$, we shrink τ_s by δ , updating τ'_s as $\tau'_s = \tau_s - \delta$. Otherwise, if $\gamma(\tau_s, \tau_e) < \alpha_2 \cdot \gamma(\tau'_s, \tau_s)$, we expand τ_s by δ , assigning τ'_s as $\tau'_s = \tau_s + \delta$, where α_1 and α_2 are predefined hyperparameters. This process can be formulated as:

$$\tau'_s = \begin{cases} \tau_s - \delta, & \text{if } \gamma(\tau_s, \tau_e) < \alpha_1 \cdot \gamma(\tau'_s, \tau_s) \\ \tau_s + \delta, & \text{elif } \gamma(\tau_s, \tau_e) < \alpha_2 \cdot \gamma(\tau'_s, \tau_s) \end{cases} \quad (3)$$

We repeat this process until no further adjustments are made to τ_s . The same approach is also applied to τ_e to refine the end time point. The final adjusted pseudo-temporal boundary is denoted as \hat{b} .

Memory Consensus Correction. Although the pseudo temporal boundaries are initially improved through semantics guided refinement, they remain inaccurate and may not fully align with the sentence queries. To address this, we introduce a memory consensus correction method that calibrates the boundaries in a coarse-to-fine manner. We maintain a memory bank \mathcal{M} to store potential candidates for pseudo temporal boundaries. For the i -th data sample, its memory bank \mathcal{M}_i is initialized as $\{\hat{b}_i\}$, where \hat{b}_i denotes the temporal boundaries adjusted by the semantics-guided refinement.

We use the same model architecture as the fully supervised TVG model SimBase [2] for pretraining. Let the model predict U temporal boundaries p_{ij}^u for the sentence query in the i -th data sample at the j -th epoch. If the memory bank \mathcal{M}_i contains N_j instances at the j -th epoch, we compute the consensus score c_r for the r -th memory instance m_r by summing its Intersection over Union (IoU) with the other $N_j - 1$ instances in the memory bank as:

$$c_r = \sum_{k=1, k \neq r}^{N_j} \sigma(m_r, m_k) \quad (4)$$

where σ denotes the IoU operator. Rather than directly using the temporal boundary \hat{b}_i as pseudo ground truth, which is still prone to errors, we use the consensus scores c_r to determine the most reliable pseudo ground truth from the

memory bank. The instance m_{r^*} with the highest consensus is selected as the pseudo ground truth to correct \hat{b}_i as:

$$r^* = \operatorname{argmax}_r(c_r) \quad (5)$$

Next, we determine which prediction p_{ij}^u to insert into the memory bank \mathcal{M}_i . In detail, we use the confidence scores predicted by the model and select u^* , the one with the highest confidence score to insert into the memory bank \mathcal{M}_i :

$$u^* = \operatorname{argmax}_u(f_u), \quad (6)$$

where f_u is the confidence score for the u -th prediction.

Finally, using the memory instance m_{r^*} with consensus, the pretraining loss function is defined as:

$$\mathcal{L}_{\text{pretrain}} = \lambda \sum_u \mathcal{L}_{\text{SimBase}}(p_{ij}^u, m_{r^*}) + (1-\lambda) \sum_u \mathcal{L}_{\text{SimBase}}(p_{ij}^u, \hat{b}_i), \quad (7)$$

where λ is a hyperparameter to balance the loss terms and $\mathcal{L}_{\text{SimBase}}$ is the loss function as defined in SimBase [2].

4.2. Adaptation to Downstream Settings

The pretrained ReCorrect model can be seamlessly adapted to various settings on the target dataset for temporal video grounding, demonstrating robust generalization, especially when manual annotations of the target dataset are limited. Here, we illustrate adaptation to zero-shot and unsupervised settings as representative examples. **1) Zero-shot setting.** The pretrained models are applied directly to the target datasets without fine-tuning, which means that the model operates without access to any videos or annotations from the target dataset. **2) Unsupervised setting.** Only unlabeled videos from the target dataset are used to finetune the pretrained models. First, we generate pseudo annotations for these unlabeled videos following Vid-Group, and then finetune the pretrained model using ReCorrect algorithm. The loss function for unsupervised finetuning is defined as:

$$\mathcal{L}_{\text{unsup}} = \mathcal{L}_{\text{pretrain}}(p^{\text{unsup}}, \hat{b}^{\text{unsup}}) \quad (8)$$

where p^{unsup} denotes the model's prediction, and \hat{b}^{unsup} represents the pseudo temporal boundary, which is progressively refined by ReCorrect. **3) Adaptation to other settings** such as fully-supervised scenarios, along with the corresponding results, is further illustrated in the supplementary materials.

5. Experiment

5.1. Datasets

As most previous works in zero-shot TVG [12, 22, 36, 48] and unsupervised TVG [11, 13, 27, 32, 47] report results on Charades-STA [8] and ActivityNet Captions [15] datasets, we evaluate the proposed methods on these two datasets.

Table 2. Performance comparison of state-of-the-art methods in **zero-shot** settings, which is divided into four parts. The first, second, and third parts (separated by dashed lines) represent three major types of state-of-the-art methods in zero-shot learning categorized by their pretraining strategies. The final part presents several competitive baseline models pretrained on the proposed Vid-Group dataset.

Method	Charades STA				ActivityNet Captions			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
UniVTG [19] ICCV23	44.09	25.22	10.03	27.12	—	—	—	—
AutoTVG [45] arxiv24	—	30.68	17.42	29.23	—	43.03	25.46	—
<hr style="border-top: 1px dashed black;"/>								
VideoChat [18] arxiv23	9.00	3.30	1.30	6.50	8.80	3.70	1.50	7.20
VideoLLaMA [43] EMNLP23	10.40	3.80	0.90	7.10	6.90	2.10	0.80	6.50
VideoChatGPT [23] ACL24	20.00	7.70	1.70	13.70	26.40	13.60	6.10	18.90
VTimeLLM-7B [12] CVPR24	51.00	27.50	11.40	31.20	44.00	27.80	14.30	30.40
VTimeLLM-13B [12] CVPR24	55.30	34.30	14.70	34.60	44.80	29.50	14.20	31.40
HawkEye [36] arxiv24	50.60	31.40	14.50	33.70	49.10	29.30	10.70	32.70
TRACE [9] ICLR25	—	40.30	19.40	—	—	—	—	—
<hr style="border-top: 1px dashed black;"/>								
Luo et al. [22] WACV24	56.77	42.93	20.13	37.92	48.28	27.90	11.57	32.37
TFVTG [48] ECCV24	67.04	49.97	24.32	44.51	49.34	27.02	13.39	34.10
VTG-GPT [38] AppliedScience24	59.48	43.68	25.94	39.81	47.13	28.25	12.84	30.49
Moment-GPT [39] AAAI25	58.20	38.40	21.60	36.50	48.10	31.10	14.90	30.80
<hr style="border-top: 1px solid black;"/>								
GPT-4o Pretraining	61.77	45.46	23.10	41.43	49.15	28.28	13.52	33.21
GPT-4o Pretraining + SPL [47]	63.35	46.48	24.28	42.47	51.70	31.04	14.16	34.55
GPT-4o Pretraining + VTG-GPT [38]	63.15	46.85	23.35	42.31	52.57	31.31	13.74	34.47
ReCorrect (Ours)	66.54	51.15	28.54	45.63	54.68	33.35	15.15	35.96

5.2. Evaluation Metrics

We adopt the evaluation metric ‘R@m’ for temporal video grounding to evaluate performance. Specifically, we calculate the Intersection over Union (IoU) between the predicted temporal moment and the ground truth. Then ‘R@m’ is defined as the percentage of language queries that have correct temporal grounding results, where a grounding result is correct if its IoU is greater than m .

5.3. Implementation Details

We utilize SimBase [2] as the network architecture for TVG model. The pretrained CLIP [29] model is used to extract visual and textual features. The hyperparameter cleaning ratio R is set to 40%. We train our model using the Adam optimizer [14] with a batch size of 256 and a learning rate of 0.0004. The pretraining epoch number is set to 15. More details on implementation can be found in the supplementary materials.

5.4. Comparisons on Zero-Shot Inference

5.4.1. Compared Methods

Table 2 compares our ReCorrect algorithm against the state-of-the-art TVG methods, which are categorized into the following classes:

- **UniVTG** [19] and **AutoTVG** [45] use existing video datasets for pre-training, namely VideoCC [26] and a

30K-video subset of HowTo100M [24], respectively. They leverage the videos’ subtitles and timestamps during pre-training, and their performance is reported in the first part of Table 2. Other TVG pretraining methods [5, 34, 37] are excluded as they cannot perform zero-shot inference.

- **VideoChat** [18], **VideoLLaMA** [43], **VideoChatGPT** [23], **VTimeLLM** [12], **HawkEye** [36], and **TRACE** [9] are video large language models for video understanding, as shown in the second part of Table 2. Among them, VTimeLLM, HawkEye, and TRACE are specifically optimized to enhance TVG capability, whereas VideoChat, VideoLLaMA, and VideoChatGPT are intended for general video understanding, with their TVG results reported in the VTimeLLM paper [12]. Their pretraining datasets vary depending on the model and include datasets such as WebVid2M [1] and InternVid [35]. Further details can be found in the original papers.
- **Luo et al.** [22], **TFVTG** [48], **Moment-GPT** [39], and **VTG-GPT** [38] in the third part of Table 2 adapt off-the-shelf vision-language foundation models (*e.g.*, CLIP [29], BLIP-2 [17], VideoChatGPT [23]) to TVG tasks by designing additional training-free modules.

Moreover, as shown in the final part of Table 2 (separated by a solid line), we carefully devise a list of competitive baselines, all of which are pretrained on the proposed Vid-Group dataset with SimBase as the TVG backbone: 1)

Table 3. Performance comparison of state-of-the-art methods under **unsupervised** settings. The gray row represents our unsupervised ReCorrect’s performance as a percentage of the state-of-the-art fully-supervised method SimBase.

Method	Charades STA				ActivityNet Captions			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
Gao et al [7] <small>TCSVT21</small>	46.69	20.14	8.27	–	46.15	26.38	11.64	–
PSVL [27] <small>ICCV21</small>	46.47	31.29	14.17	31.24	44.74	30.08	14.74	29.62
PZTVG [32] <small>MM22</small>	46.83	33.21	18.51	32.62	45.73	31.26	17.84	30.35
Kim et al. [13] <small>WACV22</small>	52.95	37.24	19.33	36.05	47.61	32.59	15.42	31.85
CoroNet [11] <small>AAAI24</small>	49.21	34.60	17.93	32.73	46.05	28.19	12.84	31.11
Lu et al. [21] <small>TIP24</small>	47.74	34.62	20.16	32.97	49.26	31.45	15.27	33.25
SPL [47] <small>ACL23</small>	60.73	40.70	19.62	40.47	50.24	27.24	15.03	35.44
GPT-4o Finetuning	61.24	44.51	22.11	40.91	49.33	28.94	13.20	33.10
ReCorrect Finetuning	65.75	47.32	25.83	44.48	55.30	35.64	17.38	37.89
GPT-4o Pretraining + Finetuning	65.72	49.10	25.21	44.22	50.58	30.56	14.13	34.09
GPT-4o Pretraining + Finetuning + SPL [47]	68.14	50.59	25.80	45.68	53.29	32.55	15.47	35.96
GPT-4o Pretraining + Finetuning + VTG-GPT [38]	67.52	50.39	25.63	45.22	53.26	32.22	14.76	35.37
ReCorrect Pretraining + Finetuning (Ours)	70.96	54.42	31.10	48.66	58.31	37.83	18.57	39.74
Fully-Supervised SimBase [2]	77.77	66.48	44.01	56.15	63.98	49.35	30.48	47.07
Relative to SimBase	91.2%	81.9%	70.7%	86.7%	91.1%	76.7%	60.9%	84.4%

GPT-4o Pretraining: Direct pretraining using pseudo labels from GPT-4o. 2) **GPT-4o Pretraining + SPL:** We reimplement the state-of-the-art unsupervised SPL [47] to refine the pseudo labels generated by GPT-4o, strictly following its official source code. 3) **GPT-4o Pretraining + VTG-GPT:** We use the boundary generation module from the leading zero-shot method VTG-GPT [38] to adjust the pseudo temporal boundaries when pretraining on the Vid-Group dataset.

5.4.2. Comparison Results

Table 2 shows that our ReCorrect method significantly outperforms all previous zero-shot approaches. For instance, it surpasses the state-of-the-art zero-shot model Moment-GPT by 7 points in R@0.7 and over 5 points in mIoU on both datasets. GPT-4o Pretraining achieves similar results to VTG-GPT on both datasets. Incorporating SPL and VTG-GPT into GPT-4o Pretraining to refine pseudo labels consistently boosts its performance across both datasets. However, the improvements delivered by our ReCorrect method are even more substantial, highlighting its superiority in progressively correcting pseudo labels through memory consensus.

5.5. Comparisons on Unsupervised Finetuning

5.5.1. Compared Methods

Under the unsupervised setting, our full ReCorrect model is first pretrained on the Vid-Group dataset and then finetuned on unlabeled videos from either the Charades-STA or ActivityNet Captions dataset. We compare the proposed method ReCorrect against a list of state-of-the-art unsupervised methods [11, 13, 21, 32, 47]. Additionally, we

consider the following baselines for comparison: 1) **GPT-4o Finetuning:** Training from scratch on the downstream dataset (*i.e.*, either Charades-STA or ActivityNet Captions) using the original pseudo annotations provided by GPT-4o, without pretraining. 2) **ReCorrect Finetuning:** Training from scratch on the downstream dataset using ReCorrect algorithm. 3) **GPT-4o Finetuning + Pretraining:** Pretraining on the Vid-Group dataset and then finetuning on the downstream dataset with the unaltered pseudo annotations from GPT-4o. 4) **GPT-4o Pretraining + Finetuning + SPL:** Pretraining and fine-tuning while employing SPL [47] to refine the GPT-pseudo labels, which is carefully implemented by us using SPL’s official source code. 5) **GPT-4o Pretraining + Finetuning + VTG-GPT:** Pretraining and fine-tuning by using the boundary generation module from VTG-GPT [38] to adjust the pseudo temporal boundaries, according to its official code.

5.5.2. Comparison Results

Table 3 summarizes the performance comparisons in unsupervised learning settings. The pretrained versions of GPT-4o Finetuning and ReCorrect Finetuning demonstrate approximately 3 and 5-point improvements over the non-pretrained versions. It is notable that the unsupervised ReCorrect method achieves approximately 85% of the overall performance of the fully-supervised SimBase on both datasets in terms of metrics mIoU. Such a close margin between unsupervised and fully supervised methods underscores the potential of our scalable and nearly labor-free Vid-Group dataset to reduce the need for manual annotations in the TVG task.

Table 4. Module ablation studies on zero-shot inference.

Clean	Adjust	Correct	R@0.3	R@0.5	R@0.7	mIoU
✗	✗	✗	61.77	45.46	23.10	41.43
✓	✗	✗	64.96	48.00	23.86	42.94
✗	✓	✗	65.27	48.28	25.63	44.17
✓	✓	✗	65.83	49.46	26.82	44.45
✓	✓	✓	66.54	51.15	28.54	45.63

Table 5. Module ablation studies on unsupervised learning.

Pretrain	Finetune			R@0.3	R@0.7	R@0.7	mIoU
	Clean	Adjust	Correct				
✗	✗	✗	✗	61.24	44.51	22.11	40.91
✓	✗	✗	✗	67.52	50.70	26.28	45.37
✓	✓	✗	✗	68.62	51.89	27.27	46.35
✓	✓	✓	✗	69.52	53.21	30.17	47.61
✓	✓	✓	✓	70.96	54.42	31.10	48.66

5.6. Ablation Studies

To assess the effectiveness of the proposed ReCorrect algorithm, we conduct ablation studies on Charades-STA.

Effectiveness of the proposed modules. Our ReCorrect algorithm comprises three major modules: 1) label cleaning, 2) boundary adjustment in the semantics-guided refinement phase, and 3) memory consensus correction. These modules are utilized for both pretraining and unsupervised learning. To evaluate their effectiveness, we investigate their impact on both zero-shot inference and unsupervised learning. Table 4 demonstrates the effectiveness of each module for zero-shot inference performance, where the three major modules are denoted as “Clean,” “Adjust,” and “Correct” respectively. Table 5 further evaluates the effectiveness of these modules on unsupervised fine-tuning. These results consistently show that each module contributes positively to performance in both pretraining and unsupervised learning, and removing any one of them leads to a notable decline in performance.

Scability of pretraining dataset size. Fig. 5 illustrates zero-shot performance against the number of data samples for pretraining on a semi-logarithmic scale. We evaluate the overall performance as the average of R@m values, where $m \in \{0.3, 0.5, 0.7\}$. A linear increase in performance is observed as the pretraining data size doubles, starting from 6.3K to 12.7K. This trend continues as the data size grows from 25.3K to 200.3K, although the slope of the increase becomes less steep at higher scales. These results demonstrate that our Vid-Group dataset exhibits scalable performance improvements.

Pilot experiments on visual prompt strategies. There are two common strategies for inputting image sequences into multimodal large language models when constructing visual prompts: feeding the image sequence separately versus

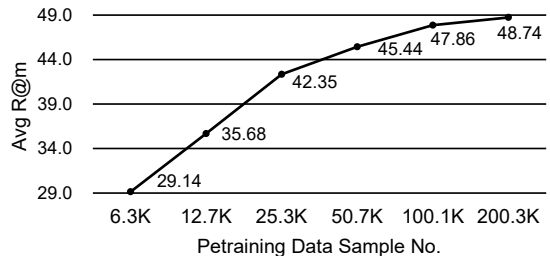


Figure 5. Scability of pretraining dataset size.

Table 6. Preliminary ablation studies on visual prompt strategies.

Method	R@0.3	R@0.5	R@0.7	mIoU
Separation	42.62	25.77	10.73	27.53
Concatenation	42.70	25.66	11.61	28.11

concatenating them into a single composite image. Prior to generating pseudo labels for the full dataset, we conduct a preliminary ablation study to evaluate the impact of these methods. Specifically, we randomly sample 1,000 unlabeled videos from the Vid-Group dataset and generate pseudo labels with GPT-4o via each method. Table 6 compares the zero-shot accuracy on the Charades-STA dataset of TVG models trained on these pseudo labels. The results show that both methods yield similar performance, with the concatenation approach performing slightly better. Consequently, we adopt the concatenation method for constructing pseudo labels for the Vid-Group dataset.

6. Conclusion

This paper introduces Vid-Group, a large-scale dataset for Temporal Video Grounding Pretraining, collected with minimal manual intervention. To address inherent errors in Vid-Group’s pseudo labels, we propose the Refinement and Correction (ReCorrect) algorithm, which features a self-correction mechanism consisting of semantics-guided refinement and memory-consensus correction. Our experiments demonstrate the effectiveness of ReCorrect in zero-shot and unsupervised learning, showing its strong generalizability.

Acknowledgements

This work was carried out at the Rapid-Rich Object Search (ROSE) Lab, School of Electrical & Electronic Engineering, Nanyang Technological University (NTU), Singapore. The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (NSCC Project ID: 12003782).

References

- [1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 6
- [2] Peijun Bao and Alex Kot. Simbase: A simple baseline for temporal video grounding. *ArXiv*, 2024. 2, 5, 6, 7
- [3] Peijun Bao, Qian Zheng, and Yadong Mu. Dense events grounding in video. In *AAAI*, 2021. 2
- [4] Peijun Bao, Yong Xia, Wenhan Yang, Boon Poh Ng, Meng Hwa Er, and Alex C Kot. Local-global multi-modal distillation for weakly-supervised temporal video grounding. In *AAAI*, 2024. 2
- [5] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. Locvtp: Video-text pre-training for temporal localization. In *ECCV*, 2022. 3, 6
- [6] Xinpeng Ding, N. Wang, Shiwei Zhang, De Cheng, Xiaomeng Li, Ziyuan Huang, Mingqian Tang, and Xinbo Gao. Support-set based cross-supervision for video grounding. In *ICCV*, 2021. 2
- [7] Junyuan Gao and Changsheng Xu. Learning video moment retrieval without a single annotated video. *TCSVT*, 2021. 2, 7
- [8] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 1, 2, 3, 5
- [9] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Qingbin Liu, and Xi Chen. Trace: Temporal grounding video llm via causal event modeling. In *ICLR*, 2025. 2, 6
- [10] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2, 3
- [11] Meghana Holla and Ismini Lourentzou. Commonsense for zero-shot natural language video localization. In *AAAI*, 2024. 2, 5, 7
- [12] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, 2024. 2, 5, 6
- [13] Dahye Kim, Jungin Park, Jiyoung Lee, Seong Hyeon Park, and Kwanghoon Sohn. Language-free training for zero-shot video grounding. In *WACV*, 2022. 2, 5, 7
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6
- [15] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 1, 2, 3, 5
- [16] Juncheng Li, Junlin Xie, Long Qian, Linchao Zhu, Siliang Tang, Fei Wu, Yi Yang, Yueting Zhuang, and Xin Eric Wang. Compositional temporal grounding with structured variational cross-graph correspondence learning. In *CVPR*, 2022. 2
- [17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 2, 3, 6
- [18] Kunchang Li, Yanan He, Yi Wang, Yizhuo Li, Wen Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *ArXiv*, 2023. 6
- [19] Kevin Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *ICCV*, 2023. 6
- [20] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. In *ACM MM*, 2018. 2
- [21] Yu Lu, Ruijie Quan, Linchao Zhu, and Yi Yang. Zero-shot video grounding with pseudo query lookup and verification. *TIP*, 2024. 7
- [22] Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. Zero-shot video moment retrieval from frozen vision-language models. In *WACV*, 2024. 2, 3, 5, 6
- [23] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahaad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*, 2024. 6
- [24] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 6
- [25] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *CVPR*, 2020. 2
- [26] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Hauth Anja, Manen Santiago, Sun Chen, and Cordelia Schmid. Learning audio video modalities from image captions. In *ECCV*, 2022. 6
- [27] Jinwoo Nam, Daechul Ahn, Dongyeop Kang, Seong Jong Ha, and Jonghyun Choi. Zero-shot natural language video localization. In *ICCV*, 2021. 2, 5, 7
- [28] Mengshi Qi, Jie Qin, Yi Yang, Yunhong Wang, and Jiebo Luo. Semantics-aware spatial-temporal binaries for cross-modal video retrieval. *TIP*, 2021. 2
- [29] Alec Radford, Jong Wook Kim, Chris Hallacy, and et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 3, 4, 6
- [30] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzels, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *TACL*, 2013. 2, 3
- [31] G. Sreenu and M. A. Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 2019. 2
- [32] Guolong Wang, Xun Wu, Zhaoyuan Liu, and Junchi Yan. Prompt-based zero-shot video moment retrieval. In *ACM MM*, 2022. 2, 5, 7
- [33] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *AAAI*, 2020. 2
- [34] Lang Wang, Gaurav Mittal, Sandra Sajeev, Ye Yu, Matthew Hall, Vishnu Naresh Boddeti, and Mei Chen. Protege: Untrimmed pretraining for video temporal grounding by video temporal grounding. In *CVPR*, 2023. 3, 6

- [35] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, and et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2024. 6
- [36] Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. Hawkeye: Training video-text llms for grounding text in videos. *ArXiv*, 2024. 2, 5, 6
- [37] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcía, Brais Martínez, Xiatian Zhu, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. In *ICCV*, 2021. 3, 6
- [38] Yifang Xu, Yunzhuo Sun, Zien Xie, Benxiang Zhai, and Sidan Du. Vtg-gpt: Tuning-free zero-shot video temporal grounding with gpt. *Applied Sciences*, 2024. 2, 6, 7
- [39] Yifang Xu, Yunzhuo Sun, Benxiang Zhai, Ming Li, Wenxin Liang, Yang Li, and Sidan Du. Zero-shot video moment retrieval via off-the-shelf multimodal large language models. In *AAAI*, 2025. 2, 6
- [40] Zhe Xu, Kun-Juan Wei, Xu Yang, and Cheng Deng. Point-supervised video temporal grounding. *TMM*, 2023. 2, 3
- [41] Yitian Yuan, Xiaohan Lan, Long Chen, Wei Liu, Xin Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In *ACM MM Workshop*, 2021. 2
- [42] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*, 2019. 2
- [43] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, 2023. 6
- [44] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, 2020. 2
- [45] Xing Zhang, Jiayi Gu, Haoyu Zhao, Shicong Wang, Hang Xu, Renjing Pei, Songcen Xu, Zuxuan Wu, and Yungang Jiang. Autotvg: A new vision-language pre-training paradigm for temporal video grounding. *ArXiv*, 2024. 6
- [46] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. Cross-modal interaction networks for query-based moment retrieval in videos. In *ACM SIGIR*, 2019. 2
- [47] Minghang Zheng, Shaogang Gong, Hailin Jin, Yuxin Peng, and Yang Liu. Generating structured pseudo labels for noise-resistant zero-shot video sentence localization. In *ACL*, 2023. 5, 6, 7
- [48] Minghang Zheng, Xinhao Cai, Qingchao Chen, Yuxin Peng, and Yang Liu. Training-free video temporal grounding using large-scale pre-trained models. In *ECCV*, 2024. 2, 3, 5, 6
- [49] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. Dsnet: A flexible detect-to-summarize network for video summarization. *TIP*, 2021. 2