applied sciences

MDPI

Article

VTG-GPT: Tuning-Free Zero-Shot Video Temporal Grounding with GPT

Yifang Xu [1], Yunzhuo Sun [2], Zien Xie [1], Benxiang Zhai [1] and Sidan Du [1,*]

[1] School of Electronic Science and Engineering, Nanjing University, Nanjing 210093, China; xyf@smail.nju.edu.cn (Y.X.); xze@smail.nju.edu.cn (Z.X.); zbx@smail.nju.edu.cn (B.Z.)
[2] School of Physics and Electronics, Hubei Normal University, Huangshi 435002, China; sunyunzhuo98@gmail.com
* Correspondence: coff128@nju.edu.cn

Abstract: Video temporal grounding (VTG) aims to locate specific temporal segments from an untrimmed video based on a linguistic query. Most existing VTG models are trained on extensive annotated video-text pairs, a process that not only introduces human biases from the queries but also incurs significant computational costs. To tackle these challenges, we propose **VTG-GPT**, a **GPT**-based method for zero-shot **VTG** without training or fine-tuning. To reduce prejudice in the original query, we employ Baichuan2 to generate debiased queries. To lessen redundant information in videos, we apply MiniGPT-v2 to transform visual content into more precise captions. Finally, we devise the proposal generator and post-processing to produce accurate segments from debiased queries and image captions. Extensive experiments demonstrate that VTG-GPT significantly outperforms SOTA methods in zero-shot settings and surpasses unsupervised approaches. More notably, it achieves competitive performance comparable to supervised methods. The code is available on GitHub.

Keywords: video temporal grounding; generative pre-trained transformer; tuning-free strategy; query debiasing

Citation: Xu, Y.; Sun, Y.; Xie, Z.; Zhai, B.; Du, S. VTG-GPT: Tuning-Free Zero-Shot Video Temporal Grounding with GPT. *Appl. Sci.* **2024**, *14*, 1894. https://doi.org/10.3390/app14051894

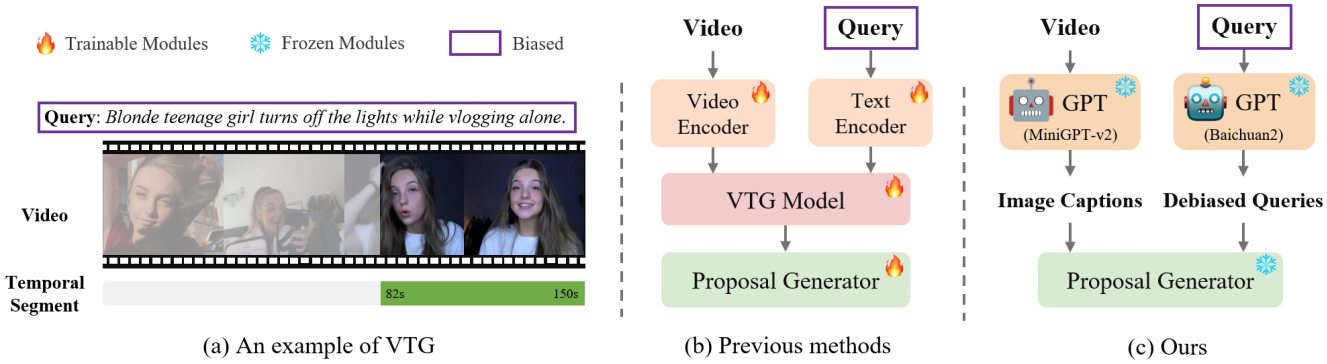
Academic Editor: Andrea Prati

Received: 18 January 2024
Revised: 16 February 2024
Accepted: 17 February 2024
Published: 25 February 2024

1. Introduction

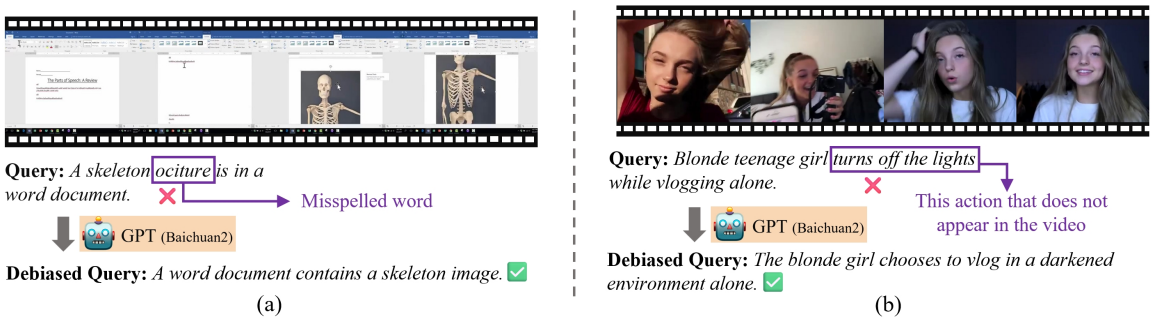
Given a linguistic query, video temporal grounding (VTG) aims to locate the most relevant temporal segments from an untrimmed video, each containing a start and end timestamp. An illustrative example of VTG is shown in Figure 1a. This task [1,2] has numerous practical applications in daily life, such as how it can help video platform users easily skip to relevant portions of a video. The field of natural language has witnessed a significant leap forward with the advent of GPT-4 [3]. This development has spurred the rise of large language models (LLMs) such as LLaMA [4] and Baichuan2 [5]. Concurrently, GPT-based (Generative Pre-trained Transformer) models like MiniGPT4 [6] and LLaVA [7] have made significant strides in vision and multimodal applications. A recent work, LLaViLo [8], reveals that training adapters alone can effectively leverage the video understanding capabilities of LLMs. However, this method requires designing a sophisticated fine-tuning strategy specifically for VTG, thereby introducing additional computing costs.

Existing VTG methods [1,9–12] primarily adopt supervised learning, which demands massive training resources and numerous annotated video-query pairs, as illustrated in Figure 1b. However, developing datasets for VTG is time-consuming and expensive; for instance, Moment-DETR [1] spent 1455 person-hours and USD 16,600 to create the QVhighlights dataset. Furthermore, ground-truth (GT) queries often contain human biases, such as (1) Bias from erroneous word spellings, as depicted in Figure 2a. The misspelled word "*ociture*" in original query would be tokenized by language models into "*o*", "*cit*", "*ure*", leading to model misunderstanding; (2) Bias due to incorrect descriptions, as shown in Figure 2b. Here, the action "*turns off the lights*" mentioned in the query does not occur in the video.



(a) An example of VTG (b) Previous methods (c) Ours

Figure 1. (**a**) An illustrative example of a video temporal grounding (VTG) task. (**b**) Previous methods require training for all modules. (**c**) Our proposed VTG-GPT operates without any training or fine-tuning. Moreover, it employs GPT to reduce bias in human-annotated queries.



(a) (b)

Figure 2. Human biases in ground-truth queries arise from (**a**) misspelled words and (**b**) incorrect descriptions. Our approach effectively mitigates these biases by leveraging GPT to optimize raw queries.

In this paper, we propose a tuning-free zero-shot method named VTG-GPT to address the above issues. As shown in Figure 1c, VTG-GPT completely satisfies zero-shot settings, adopting a direct feed-forward approach without training or fine-tuning. To minimize biases arising from human-annotated queries, we employ Baichuan2 [5] to rephrase the original query and obtain debiased queries. As illustrated in Figure 2, the erroneous word "*ociture*" in query (a) has been accurately revised to "*image*", and the non-existent action "*turn off the lights*" in query (b) has been effectively refined to "*a darkened environment*". Furthermore, considering that videos inherently contain more redundant information than text, and inspired by the human approach to understanding video linguistically, we apply MiniGPT-v2 [6] to transform visual content into more precise textual descriptions. Finally, to generate accurate temporal proposals, we design a proposal generator that models debiased queries and image captions in the textual domain. In summary, our main contributions include:

(1) To the best of our knowledge, we are the first zero-shot method to utilize GPT on VTG without training or fine-tuning.

(2) We present a novel framework, VTG-GPT, which effectively leverages GPT to mitigate human prejudice in annotated queries. Furthermore, VTG-GPT distinctively models debiased queries and video content within the linguistic domain to generate temporal segments.

(3) Comprehensive experiments demonstrate that VTG-GPT significantly surpasses SOTA (State-of-the-Art) methods in zero-shot settings. More importantly, this method achieves competitive performance comparable to supervised methods.

2. Related Work

2.1. Video Temporal Grounding

For fully-supervised VTG, prior works [1,9–11,13–16] typically employ encoders to extract visual and textual features, followed by designing a VTG model (e.g., transformer encoder-decoder) to interact and align two modalities, as depicted in Figure 1b. UniVTG [13] designs a multi-modal and multi-task learning pipeline, undergoing pre-training or fine-tuning on dozens of datasets. To accelerate the training convergence of VTG, GPTSee [14] introduces LLMs to generate prior positional information for the transformer decoder. However, these supervised approaches inevitably rely on extensive human-annotated data and training resources. To alleviate the dependence on annotations, PSVL [17], DSCNet [18], and Gao et al. [19] propose unsupervised frameworks that employ clustering to generate pseudo queries from video features. Similarly, PZVMR [20] and Kim et al. [21] leverage CLIP [22] for pseudo query generation. Yet, the above unsupervised methods unavoidably introduce biases from mismatched video-query pairs. In this paper, we adhere to the definitions of unsupervised and zero-shot settings as discussed by Luo et al. [23], classifying these approaches [17,20,21] as unsupervised.

To avoid any training or fine-tuning of the model, Diwan et al. [2] design the first zero-shot framework utilizing CLIP, but its reliance on shot transition detectors for obtaining temporal segments limits performance. Considering that CLIP (InternVideo [24]) pre-trained on 400 M image-text (12 M video-text) pairs can align visual and textual inputs in a shared feature space, Luo et al. [23] develop a bottom-up pipeline to leverage the capabilities of vision-language models. Wattasseril et al. [25] employ the sparse frame-sampling strategy and BLIP2 [26] to reduce the computational cost of inference. However, these zero-shot methods [2,23,25] tend to generate redundant video features, introducing new biases that impair model performance. A recent study [27] found that masking over 75% of the input images can effectively train large self-supervised models. Moreover, SeViLA [28] demonstrates that transforming visual signals into textual representations significantly reduces redundant information, thereby boosting performance in tasks such as video question answering and VTG.

2.2. Generative Pre-Trained Transformer

The groundbreaking success of GPT-4 [3] in the language domain has led to the development of a series of open-source LLMs [4,5,29,30]. Baichuan2 [5], containing 7 billion parameters and 2.6 trillion tokens, excels in vertical domains such as technology and daily conversation. MiniGPT4 [6,31] introduces a large multi-modal model (LMM) based on GPT, adept at performing visual-linguistic tasks like image captioning and visual question answering. Recent studies demonstrate that leveraging GPT models effectively reduces prejudice originating from ground truth labels, while simultaneously enhancing model performance in zero-shot multimodal tasks. This advancement is particularly notable in areas such as relation detection and information extraction, showcasing the robust generalization capabilities of GPT in these complex scenarios. To further capitalize on GPT's capabilities in video understanding, LLaViLo [8] designs specialized adapters for VTG, but this method still necessitates model training. To overcome these limitations, this paper proposes a novel zero-shot VTG pipeline aiming to eliminate human biases from GT queries while fully harnessing the visual comprehension capabilities of GPT, achieving a tuning-free framework.

3. Our Method

In this section, we first formulate the VTG task and then present the overall architecture of our VTG-GPT. Subsequently, we provide details of each module in the model.

3.1. Overview

Given an untrimmed video $V \in \mathbb{R}^{N_v \times H \times W \times 3}$ consisting of N_v frames and a natural language query $T \in \mathbb{R}^{L_t}$ formed by L_t words, the objective of video temporal grounding

(VTG) is to precisely identify time segments $[t_s, t_e] \in \mathbb{R}^{N_s \times 2}$ in V that semantically correspond to T, where each segment starts at timestamp t_s and ends at timestamp t_e. The overview of our proposed VTG-GPT is illustrated in Figure 3.

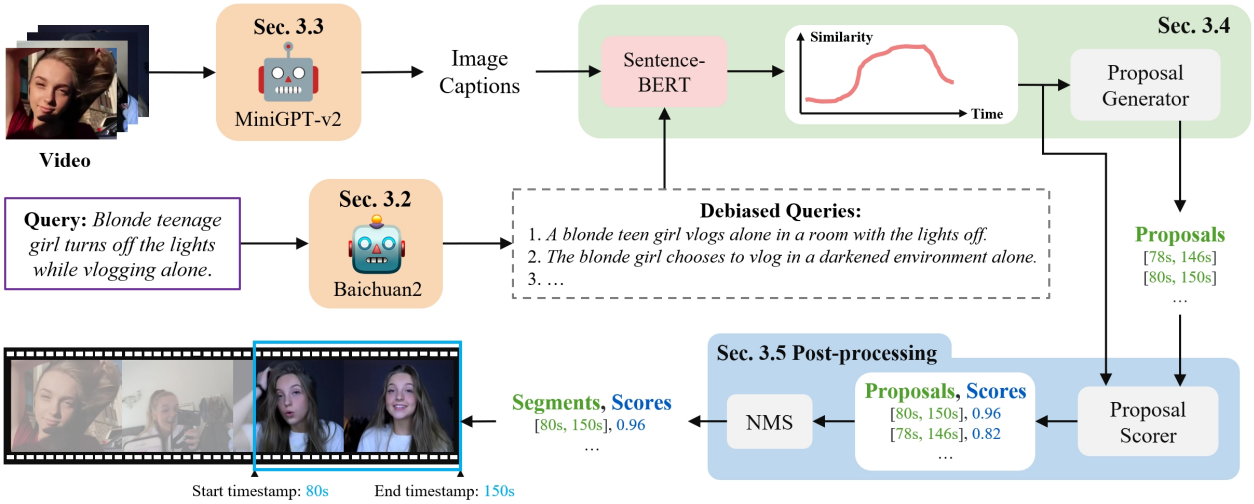


Figure 3. An overview of our proposed VTG-GPT, the framework contains four key phases: query debiasing (Section 3.2), image captioning (Section 3.3), proposal generation (Section 3.4), and post-processing (Section 3.5).

The core aim of VTG-GPT is to implement a tuning-free framework while reducing human bias in the process. To this end, our first step is employing Baichuan2 (Section 3.2) to refine raw query T, resulting in debiased queries $Q \in \mathbb{R}^{N_q \times L_t}$. Then, we leverage MiniGPT-v2 (Section 3.3) to convert visual content in each frame into image captions $C \in \mathbb{R}^{N_v \times L_c}$, effectively reducing redundant information in video V. In Section 3.4, we compute similarity scores $S_s \in \mathbb{R}^{N_q \times N_v}$ between Q and C, which is to say, in the linguistic domain via Sentence-BERT assess query-frame correlation. Following this, a proposal generator is designed to yield temporal proposals $P \in \mathbb{R}^{N_p \times 2}$. Finally, in the post-processing stage (Section 3.5), we calculate final scores $S_f \in \mathbb{R}^{N_p}$ for each proposal while removing excessively overlapping proposals to produce predicted segments $Seg \in \mathbb{R}^{N_s \times 2}$.

3.2. Query Debiasing

Mitigating biases in ground-truth queries represents a crucial and challenging problem for VTG, as these biases often originate from inherent human subjectivity. Such biases often include errors like misspellings and inaccurate descriptions of video content, as shown in Figure 2. Moreover, different annotators may characterize the same video segment in varying ways. A minority might opt for a formal language style, while others might gravitate towards colloquial or slang expressions. This difference in descriptions can inadvertently lead the model to prefer certain types of queries, thus introducing human prejudice and potentially diminishing the model's performance.

To address the aforementioned challenges, we utilize Baichuan2 to eliminate human biases inherent in original queries, as demonstrated in Figure 4a. In line with human linguistic comprehension [32], our first step is to rectify spelling and grammatical inaccuracies in original query T, thus producing the corrected version T_c. We direct GPT with the instruction: *Please correct spelling and grammatical errors in the original query.* Subsequently, we instruct Baichuan2 to rewrite T_c to remove incorrect descriptions. The corresponding command is *Please rephrase the corrected query using different wording while maintaining the same intent and information.* Finally, we generate five semantically similar yet syntactically diverse queries Q to prevent the model from relying on a specific query type. The command for this is *Provide five different rephrasings.* Although it is generally advisable to issue only one command per message in GPT dialogues to avoid model errors, as noted in [30], we

discover in our tests that aggregating all instructions into a single message to GPT proved more effective, as shown in Figure 4a. It is important to note that the red font is not present in the code. A case involving misspelled words is shown in Figure 5a, where the incorrectly spelled word "*ociture*" is corrected to "*image*" or "*picture*". Figure 5b demonstrates a scenario involving a non-existent action, where "*turn off the lights*" is optimized to "*lights off*" or "*a darkened environment*", where "*a darkened environment*" is more congruent with the original video segment. In short, this debiasing strategy, featuring variations that differ in structure and word choice, deeply explores semantic information and enables the model to process various real-world queries effectively.

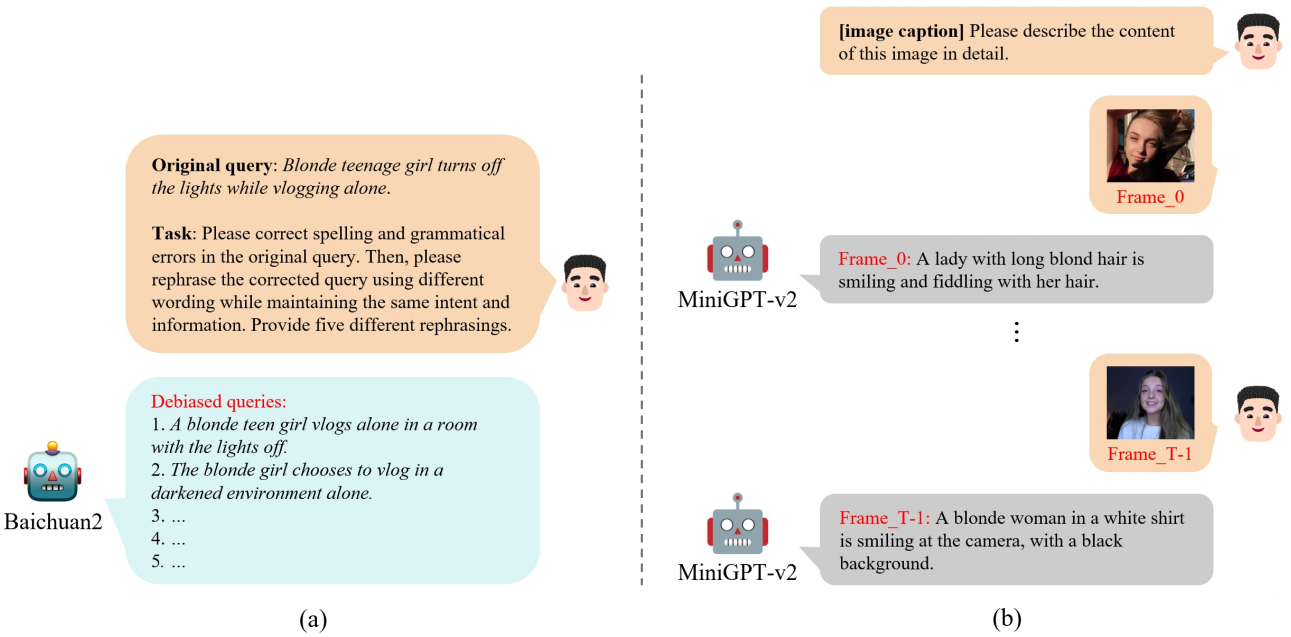


Figure 4. (**a**) An example of query refinement using Baichuan2. (**b**) An example of image captioning using MiniGPT-v2. The red font employed here is for demonstration purposes only and is not present in the actual code.

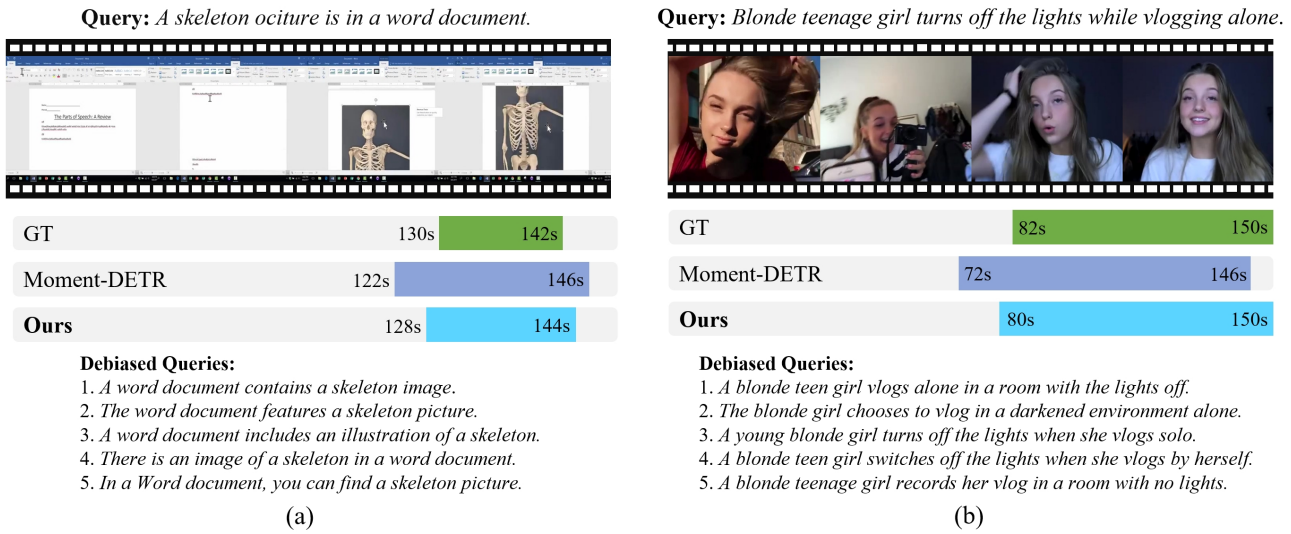


Figure 5. Visualization of predictions on QVHighlights *val* split. (**a**) misspelled words. (**b**) incorrect descriptions. Our VTG-GPT achieves more precise localization compared to Moment-DETR [1], as it can correct errors in the original queries through rewriting and generate debiased queries, thereby facilitating more accurate grounding.

3.3. Image Captioning

To retrieve corresponding video segments *Seg* based on the query, traditional zero-shot methods [2,23] initially employ pre-trained multi-modal models [22,24] for feature extraction from visual and textual modalities. These features are then used to calculate similarities to derive *Seg*. However, our preliminary experiments utilizing CLIP and InternVideo to assess cross-modal similarity, as shown in the upper part of Table 1, yielded mediocre results. We attribute this to the over-reliance of traditional methods on directly modeling raw frames, which is often influenced by background details, thereby reducing the accuracy of primary content recognition. Some recent works [28,33] suggest that videos contain abundant non-essential information and that translating visual signals into more abstract descriptions can enhance VTG performance.

Inspired by the above research, we incorporate a large multi-modal model (LMM), MiniGPT-v2 [6], to obtain more detailed image descriptions. As demonstrated in Figure 4b, our initial instruction to MiniGPT-v2 is *[image caption] Please describe the content of this image in detail.*, where *[]* emphasizes the task to be performed. Subsequently, we sequentially send frames in video V to MiniGPT-v2, which provides us with detailed captions $C \in \mathbb{R}^{N_v \times L_c}$. Following this, we use the CLIP text encoder (CLIP-T) to extract linguistic features from C and Q and calculate their similarities. As illustrated in the third row of Table 1, the results are surprisingly effective, achieving significant gains with this straightforward approach. We ascribe this to the LMM's focus on capturing key image content, thereby reducing irrelevant background interference and enhancing semantic similarities between queries and frames. For instance, the last frame in Figure 4b, depicting "*A blonde woman in a white shirt is smiling at the camera, with a black background.*", is succinctly translated into text, closely matching the query: "*The blonde girl chooses to vlog in a darkened environment alone.*" semantically.

Table 1. Preliminary experiment with different similarity models on QVHighlights *val* split, using proposal generator and proposal scorer but without NMS. CLIP-T is short for using CLIP [22] text encoder only. Please refer to Section 4.1 for a detailed explanation of the evaluation metrics.

Similarity Models	R1		mAP		
	@0.5	@0.7	@0.5	@0.75	Avg.
CLIP [22]	45.59	26.03	45.56	23.14	24.91
InternVideo [24]	49.13	32.49	48.65	25.82	26.94
CLIP-T [22]	52.85	34.82	48.07	28.05	28.29
RoBERTa [34]	**54.99**	37.58	53.77	29.18	30.15
Sentence-BERT [35]	54.26	**38.45**	**53.96**	**29.25**	**30.38**

3.4. Proposal Generation

Computing query-frame similarity. In Section 3.3, we have articulated the significance of image captioning within VTG-GPT and employed CLIP-T to model debiased queries Q and image captions C within the textual domain. Subsequently, taking into account CLIP-T, as a multi-modal model, does not outperform specialized language models in NLP (natural language processing) tasks, as outlined in previous research [36]. Therefore, we explore the use of a language-specific model. We opt for RoBERTa [34] (Sentence-BERT [35]) to extract normalized pooling features of $Q \in \mathbb{R}^{N_q \times L_t}$ and $C \in \mathbb{R}^{N_v \times L_c}$, denoted as $f_q \in \mathbb{R}^{N_q \times d}$ and $f_c \in \mathbb{R}^{N_v \times d}$, respectively, where d represents the dimensionality. We then compute the cosine scores between f_q and f_c as similarities $S_s \in \mathbb{R}^{N_q \times N_v}$:

$$S_s = \cos(f_q, f_c) = \frac{f_q \cdot f_c}{\|f_q\| \|f_c\|} \tag{1}$$

As demonstrated in rows four to five of Table 1, the leverage of expert NLP models yielded significant improvements, which also validates the viewpoints presented in the report [36].

Proposal generator. After obtaining query-frame similarity scores S_s, we move towards generating temporal proposals $P \in \mathbb{R}^{N_p \times 2}$. A straightforward method would be to apply a fixed threshold, considering frames with similarity scores exceeding this threshold as potential start or end timestamps. However, each query-video pair exhibits a unique similarity distribution. To adaptively obtain proposals, we introduce a dynamic mechanism within our devised proposal generator. For clarity, we denote the similarity between the i-th debiased query Q^i and video V as $S_s^i \in \mathbb{R}^{N_v}$, and the similarity between Q^i and the j-th frame in V as $S_s^{i,j} \in \mathbb{R}^1$.

To be specific, the generator begins by computing a histogram of S_s^i with N_b bins. It then selects the bins containing the top k highest similarities as the dynamic threshold θ:

$$\theta = \text{top\_k}(S_s^i, N_b, k), \tag{2}$$

where N_b and k are hyperparameters. For their specific values, please refer to the implementation details (Section 4.1) and ablations (Section 4.3). Next, we iteratively assess each frame; if $S_s^{i,j}$ exceeds θ, its corresponding timestamp is considered the proposal's starting point. When more than λ consecutive frames are all lower than θ, the last frame with a similarity greater than θ is marked as the end timestamp of this proposal. Here, λ denotes the continuity threshold. Finally, we produce proposals for all debiased queries in the same video using this process to form final temporal proposals $P \in \mathbb{R}^{N_p \times 2}$ (representing potentially relevant video segments).

3.5. Post-Processing

Proposal scorer. In Section 3.4, we generate a set of temporal proposals P through our designed proposal generators. To identify the most fitting video segments from P, it is essential to compute and rank each proposal's confidence score. Intuitively, a straightforward approach could be averaging the similarity scores for each frame within a proposal, or only considering frames exceeding dynamic threshold θ. However, these methods overlook the impact of proposal length on their scoring. In our experiments, we observe that within certain ground-truth segments containing scene transitions, the similarity of some frames significantly exceeded that of adjacent frames. This led to an excessively high dynamic threshold, resulting in the predicted segments being truncated or fragmented. To address this issue, we develop a length-aware scoring mechanism for proposals, encouraging the model to generate longer segments. Specifically, the evaluation of each proposal considers both its duration and the query-frame similarity, and the final score of each proposal $S_f \in \mathbb{R}^{N_p}$ is calculated as follows:

$$S_f = \alpha \times S_l + (1 - \alpha) \times S_s, \tag{3}$$

where $S_l = L_p / L_n$. Here, L_p represents the count of frames within a proposal exceeding θ, and L_n denotes the total number of frames exceeding θ across the entire video. The balancing coefficient α is adjustable to optimize for the influence of length and similarity in the final score calculation.

NMS. In the final stage, considering that multiple debiased queries will produce numerous overlapping proposals, we employ non-maximum suppression (NMS) to reduce redundant overlaps and derive the final predicted video segments $Seg \in \mathbb{R}^{N_s \times 2}$:

$$Seg = \text{NMS}(P, S_f, \mu), \tag{4}$$

where segments exceeding the intersection over union (IoU) threshold μ are selectively eliminated. This method ensures that only the most representative and distinct video segments are retained, enhancing the accuracy and relevance of our VTG-GPT output.

4. Experiments

4.1. Experimental Settings

Datasets. To demonstrate the superiority and effectiveness of our proposed tuning-free VTG-GPT framework, we conduct extensive experiments on three publicly available datasets: QVHighlights [1], Charades-STA [37], and ActivityNet-Captions [38], as these datasets encompass diverse types of videos. **QVHighlights** consists of 10,148 distinct YouTube videos, each accompanied by human annotations that include a textual query, a temporal segment, and frame-level saliency scores. Here, the saliency scores serve as the output for the highlight detection (HD) task, quantifying the relevance between a query and its corresponding frames. QVHighlights encompasses a wide array of themes, ranging from daily activities and travel in everyday vlogs to social and political events in news videos. For evaluation, Moment-DETR [1] allocates 15% of the data for validation and another 15% for testing, with consistent data distribution across both sets. Due to limitations on the online test server (https://codalab.lisn.upsaclay.fr/competitions/6937, accessed on 1 September 2023) allowing a maximum of five submissions, all our ablation studies are conducted on the validation split. **Charades-STA**, derived from the original Charades [39] dataset, includes 9848 videos of human indoor activities, accompanied by 16,128 annotations. For this dataset, a standard split of 3720 annotations is specifically designated for testing. **ActivityNet-Captions**, built upon the raw ActivityNet [40] dataset, comprises 19,994 long YouTube videos from various domains. Since the test split is reserved for competitive evaluation, we follow the setup used in 2D-TAN [16], utilizing 17,031 annotations for testing.

Metrics. To effectively evaluate performance on VTG, we employ several metrics, including Recall-1 at Intersection over Union (IoU) thresholds (R1@m), mean average precision (mAP), and mean IoU (mIoU). R1@m measures the percentage of queries in the dataset where the highest-scoring predicted segment has an IoU greater than m with the ground truth. mIoU calculates the average IoU across all test samples. For a fair comparison, our results on the QVHighlights dataset report R1@m with m values of 0.5 and 0.7, mAP at IoU thresholds of 0.5 and 0.75, and the average mAP across multiple IoU thresholds [0.5:0.05:0.95]. For the Charades-STA dataset, we report R1@m for m values of 0.3, 0.5, and 0.7, along with mIoU. Finally, we employ mAP and HIT@1 to evaluate the results of HD, thereby measuring the query-frame relevance. Here, HIT@1 represents the accuracy of the highest-scoring frame.

Implementation details. To mitigate video information redundancy, we downsample QVHighlights and Charades-STA datasets to a frame rate of 0.5 per second. Considering the extended duration of videos in the ActivityNet-Captions, we extract one frame every three seconds. In the image captioning stage, we utilize MiniGPT-v2 [6] based on the LLaMa-2-Chat-7B [4]. For query debiasing, we employ Baichuan2-7B-Chat [5], also based on LLaMa-2 [5], generating five debiased queries ($N_q = 5$) per instance. The temperature coefficients for MiniGPT-v2 and Baichuan2 are set at 0.1 and 0.2, respectively. Drawing from the preliminary experiments in Section 3.4, we select Sentence-BERT [35] as our similarity model to evaluate query-frame correlations using cosine similarity. The histogram in our proposal generator is configured with ten bins (N_b), with a selection of the top eight values ($k = 8$) and a continuity threshold $\lambda = 6$. During the post-processing phase, the balance coefficient (α) in the proposal scorer is set to 0.5, and the IoU threshold (μ) for non-maximum suppression (NMS) is determined at 0.75. All pre-processing and experiments are conducted on eight NVIDIA RTX 3090 GPUs. It is important to note that our VTG-GPT is purely inferential, involving no training phase.

4.2. Comparisons to the State-of-the-Art

In this section, we present a comprehensive comparison of our VTG-GPT with state-of-the-art (SOTA) methods in VTG. Firstly, we disclose results on the QVHighlights validation and test splits, as shown in Table 2. The approaches are categorized into fully supervised (FS), weakly supervised (WS), unsupervised (US), and zero-shot (ZS) methods. Notably,

VTG-GPT significantly outperforms the previous SOTA zero-shot model (Diwan et al. [2]), demonstrating substantial improvements across five metrics. Specifically, R1@0.7 saw an increase of +7.49 and mAP@0.5 improved by +7.23. Remarkably, VTG-GPT also vastly exceeds all WS methods. Most impressively, our approach surpasses the FS baseline (Moment-DETR [1]) in most metrics, even achieving competitive performance compared with FS methods. Unlike these methods, VTG-GPT requires only a single inference pass, eliminating the need for training data and resources.

Table 2. Performance comparison on QVHighlights *test* and *val* split. FS means fully-supervised method, WS means weakly supervised, and ZS means zero-shot.

Method	Year	Setup	QVHighlights Test					QVHighlights val				
			R1		mAP			R1		mAP		
			@0.5	@0.7	@0.5	@0.75	Avg.	@0.5	@0.7	@0.5	@0.75	Avg.
Moment-DETR [1]	2021	FS	52.89	33.02	54.82	29.40	30.73	53.94	34.84	-	-	32.20
LLaViLo [8]	2023	FS	59.23	41.42	59.72	-	36.94	-	-	-	-	-
UMT [9]	2022	FS	56.23	41.18	53.83	37.01	36.12	-	-	-	-	37.79
MH-DETR [10]	2023	FS	60.05	42.48	60.75	38.13	38.38	60.84	44.90	60.76	39.64	39.26
QD-Net [11]	2023	FS	62.32	45.61	63.15	42.05	41.46	61.71	44.76	61.88	39.84	40.34
EaTR [15]	2023	FS	-	-	-	-	-	61.36	45.79	61.86	41.91	41.74
CNM [41]	2022	WS	14.11	3.97	11.78	2.12	-	-	-	-	-	-
CPL [42]	2022	WS	30.72	10.75	22.77	7.48	-	-	-	-	-	-
CPI [43]	2023	WS	32.26	11.81	23.74	8.25	-	-	-	-	-	-
UniVTG [13]	2023	ZS	25.16	8.95	27.42	7.64	10.87	-	-	-	-	-
Diwan et al. [2]	2023	ZS	-	-	-	-	-	48.33	30.96	46.94	25.75	27.96
VTG-GPT (Ours)	2023	ZS	**53.81**	**38.13**	**54.13**	**29.24**	**30.50**	**54.26**	**38.45**	**54.17**	**29.73**	**30.91**

Subsequently, we report the performance on the Charades-STA test set and ActivityNet-Captions test set in Table 3. In Charades-STA, VTG-GPT surpasses the SOTA zero-shot method (Luo et al. [23]) with a +5.81 increase in R1@0.7 and a +1.89 improvement in mIoU. Furthermore, VTG-GPT significantly outperforms the best US method (Kim et al. [21]) across all metrics. However, on the ActivityNet-Captions dataset, our method falls slightly behind Luo et al. in two metrics, which we attribute to the high downsampling rate used for this dataset. Moreover, VTG-GPT approaches the performance of the fully supervised Moment-DETR, validating its capacity to handle diverse and complex video contexts without any training or fine-tuning. This underscores the robustness and adaptability of VTG-GPT in zero-shot VTG scenarios, demonstrating its potential as a versatile and efficient tool for video understanding.

Table 3. Performance comparison on Charades-STA *test* split and ActivityNet-Captions *test* split. Where FS means fully-supervised setting, WS means weakly-supervised, US means unsupervised, and ZS means zero-shot.

Method	Year	Setup	Charades-STA				ActivityNet-Captions			
			R1@0.3	R1@0.5	R1@0.7	mIoU	R1@0.3	R1@0.5	R1@0.7	mIoU
2D-TAN [16]	2020	FS	57.31	45.75	27.88	41.05	60.32	43.41	25.04	42.45
Moment-DETR [1]	2021	FS	65.83	52.07	30.59	45.54	-	-	-	-
VDI [12]	2023	FS	-	52.32	31.37	-	-	48.09	28.76	-
CNM [41]	2022	WS	60.04	35.15	14.95	38.11	55.68	33.33	13.29	37.55
CPL [42]	2022	WS	65.99	49.05	22.61	43.23	55.73	31.37	13.68	36.65
Huang et al. [44]	2023	WS	69.16	52.18	23.94	45.20	58.07	36.91	-	41.02
PSVL [17]	2021	US	46.47	31.29	14.17	31.24	44.74	30.08	14.74	29.62
Gao et al. [19]	2021	US	46.69	20.14	8.27	-	46.15	26.38	11.64	-
DSCNet [18]	2022	US	44.15	28.73	14.67	-	47.29	28.16	-	-
PZVMR [20]	2022	US	46.83	33.21	18.51	32.62	45.73	31.26	17.84	30.35
Kim et al. [21]	2023	US	52.95	37.24	19.33	36.05	47.61	32.59	15.42	31.85
UniVTG [13]	2023	ZS	44.09	25.22	10.03	27.12	-	-	-	-
Luo et al. [23]	2023	ZS	56.77	42.93	20.13	37.92	**48.28**	27.90	11.57	**32.37**
VTG-GPT (Ours)	2023	ZS	**59.48**	**43.68**	**25.94**	**39.81**	47.13	**28.25**	**12.84**	30.49

To qualitatively validate the effectiveness of our VTG-GPT model, we present visual comparisons of grounding results from the Ground-Truth (GT), Moment-DETR, and VTG-GPT in Figure 5. Observations indicate that the tuning-free VTG-GPT achieves more precise localization than the supervised Moment-DETR. The primary reason is that Moment-DETR relies solely on the original queries, which contain human-annotated errors, thus failing to fully align with the video's semantic information. In contrast, VTG-GPT can correct erroneous queries and reduce the bias introduced by human annotations, leading to more accurate grounding. To be more specific, in Figure 5a, our model detects a spelling mistake in the query, where "*ociture*" is corrected to "*image*" or "*picture*". In Figure 5b, the action "*turns off*" is refined to terms more congruent with the video context, such as "*lights off*", "*darkened environment*", and "*no lights*". Additionally, the five rephrasings of each original query, in contrast to the original phrasing, exhibit more flexible grammatical structures, enabling the text encoder to comprehensively capture the semantic information of the original query.

4.3. Ablation Studies

To demonstrate the effectiveness of each module within our VTG-GPT framework, we perform in-depth ablation studies on the QVHighlights dataset.

Effect of debiased query. Firstly, we report saliency scores used to evaluate query-frame correlation. As delineated in Table 4, row three corresponds to VTG-GPT without debiasing, where we directly employ the similarity generated by Sentence-BERT as the saliency scores. Conversely, row four is VTG-GPT with debiasing, wherein we average the similarity of five debiased queries as saliency scores. The comparison reveals that row four significantly outperforms row three, demonstrating the efficacy of our debiasing strategy in mitigating human biases inherent in the original queries. Furthermore, comparing row two (UMT [9]) and row four, our VTG-GPT achieves a notable increase in HIT@1, recording a score of 62.29 (+2.3). This enhancement underscores VTG-GPT's superior reasoning capabilities in discerning challenging cases, affirming the value of our debiasing approach in refining model performance.

Table 4. Comparison of video highlight detection (HD) on QVHighlights *val* split. VG is the abbreviation of very good. ✓ and ✗ respectively represent the use and non-use of debiased queries.

Methods	Setup	Debiasing	HD (≥VG)	
			mAP	HIT@1
Moment-DETR [1]	FS	✗	35.69	55.60
UMT [9]	FS	✗	**38.18**	59.99
VTG-GPT	ZS	✗	34.84	60.48
		✓	36.08	**62.29**

Then, we investigate the impact of different numbers of debiased queries (N_q) generated by Baichuan2 on the performance of the VTG-GPT model. Our findings, as depicted in Figure 6a, indicate that the model achieves optimal results when utilizing five debiased queries ($N_q = 5$). Compared to using solely the original biased query, implementing five debiased queries resulted in a notable increase in R1@0.5 to 54.26 (+3.87) and an improvement in mAP Avg. to 30.91 (+2.59). This evidence suggests that removing bias from queries significantly enhances the model's accuracy. However, an intriguing observation emerged: the performance metrics decline when N_q exceeds 5. This pattern suggests that excessive rephrasing does not continually yield improvements, likely due to the finite number of synonymous rewrites and syntactic variations available to maintain the original intent of the query. Over-rephrasing can introduce irrelevant content, deviating from the semantic intent of the raw query, and potentially diminishing model performance. This finding underscores the critical need to balance the number of query rewrites, ensuring that debiased queries capture a spectrum of semantic nuances while retaining the essence of the original

query. Future research should focus on developing advanced query debiasing techniques to enhance this equilibrium.

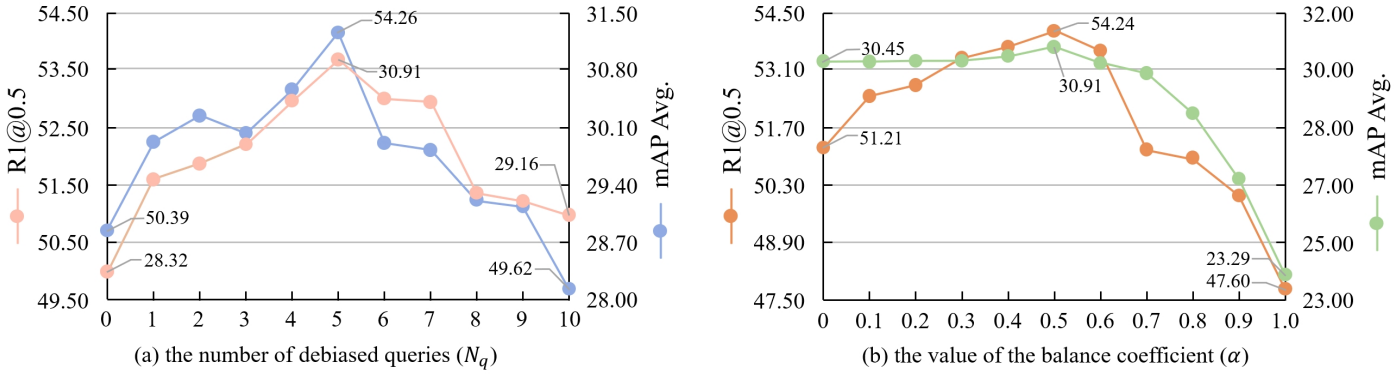


Figure 6. Ablation experiments on the QVHighlights *val* split focus on R1@0.5 and mAP Avg. (**a**) Utilizing debiased queries can enhance model performance, yet increasing the number of debiased queries (N_q) does not always lead to better results. The model achieves optimal performance when N_q is set to 5. (**b**) In the proposal scorer, proposal length significantly impacts the final outcomes, with the model performing optimally when $\alpha = 0.5$.

LLMs and LMMs. In Table 5, we evaluate the capabilities of LLMs (LLaMA-v2 [4] and Baichuan2 [5]), alongside LMMs (MiniGPT-4 [31] and MiniGPT-v2 [6]) in handling biased queries and generating image captions. A comparison between rows two and five reveals that Baichuan2 outperforms LLaMa-v2, since it is trained on a more diverse dataset and tasks based on LLaMa-v2, enhancing its sentence rewriting capabilities. As illustrated in row three, MiniGPT-v2, also developed on the foundations of LLaMa-v2, shows moderate results in text dialogue. Comparing rows four and five, we observe an improvement in the performance of MiniGPT-v2 over MiniGPT-4. Overall, the results suggest that the integration of Baichuan2 for query debiasing combined with MiniGPT-v2 for image captioning emerges as the most effective strategy. This effectiveness stems from their complementary capabilities: Baichuan2 excels in handling complex multi-turn text dialogues, while MiniGPT-v2 is adept at detailed multimodal dialogues. This synergy maximizes the text comprehension ability of LLMs and the video understanding capacity of LMMs, thereby enhancing the overall performance of our framework.

Proposal generator. In our study, top-k and the continuity threshold λ within the proposal generator play a critical role. The parameter k, acting as a count threshold in our dynamic mechanism, directly influences the identified length of relevant proposals. In contrast, λ determines the number of irrelevant consecutive frames. To optimize these parameters, we conducted a series of ablation experiments on the proposal generator, as illustrated in Figure 7. The visualized results indicate that a combination of $k = 8$ and $\lambda = 6$ yields the most favorable outcomes. This specific pairing strikes a balance between segment length and threshold sensitivity. It skillfully avoids the drawbacks of excessively low thresholds, which could incorporate irrelevant frames into prediction results. Simultaneously, it averts the "tolerance trap" where an overly high number of discontinuous frames makes it difficult to determine when the segment ends.

Proposal scorer. To balance the quality and length of segments, we conduct experiments on our proposal scorer, as shown in Figure 6b. We explore integrating the length score S_l into the scoring mechanism. Initially, without including the length score ($\alpha = 0$), mAP Avg. is 30.45. Upon incorporating S_l, mAP Avg. peak at 30.91. Similarly, R1@0.5 increases from 51.21 to 54.24, indicating that incorporating a length-based scoring mechanism is crucial for generating the final segment scores.

Table 5. Ablation study of different LLMs and LMMs (Large Multi-modal Models) on QVHighlights *val* split.

Debiasing	Captioning	R1@0.5	R1@0.7	mAP Avg.
LLaMA-v2 [4]	MiniGPT-4 [31]	50.78	30.56	27.20
LLaMA-v2	MiniGPT-v2 [6]	**54.65**	34.08	30.15
MiniGPT-v2	MiniGPT-v2	50.46	29.87	27.48
Baichuan2 [5]	MiniGPT-4	52.78	33.84	28.54
Baichuan2	**MiniGPT-v2**	54.26	**38.45**	**30.91**

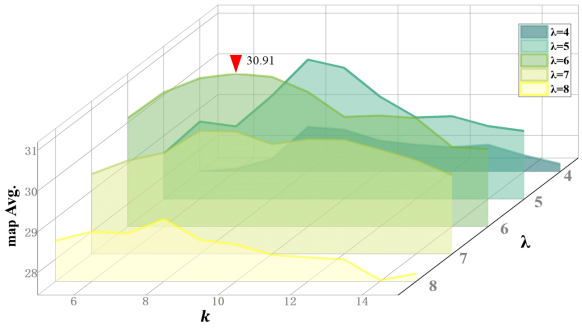


Figure 7. Ablation experiments for top-k and continuity threshold (λ) in proposal generator on QVHighlights *val* split. When $k = 8$ and $\lambda = 6$, the model achieves the best performance (red triangle).

IoU threshold μ. Finally, we assess the effectiveness of IoU thresholds μ in the NMS process, focusing on their role in reducing segment overlap. It is important to note that NMS does not alter the values of R1@0.5 and R1@0.75. Therefore, we report only the mAP metrics in Table 6. As illustrated in Table 6, setting μ to 0.75, compared to not employing NMS ($\mu = 1$), results in an increase of +0.53 in mAP Avg. This increment underscores the significance of eliminating excessively overlapping segments, affirming that reducing such overlaps can notably enhance the model's performance.

Table 6. Comparison of different IoU thresholds (μ) in NMS on QVHighlights *val* split.

μ	mAP@0.5	mAP@0.75	mAP Avg.
0.6	53.71	28.48	30.06
0.7	54.12	29.60	30.54
0.75	**54.17**	29.73	**30.91**
0.8	54.02	**29.87**	30.68
0.9	53.81	29.63	30.41
1.0	53.96	29.25	30.38

5. Conclusions

This paper proposes a tuning-free framework named VTG-GPT for zero-shot video temporal grounding. To minimize the bias from mismatched videos and queries, we employ Baichuan2 for refining human-annotated queries. Recognizing the inherent redundancy in video compared to text, we utilize MiniGPT-v2 to transform visual inputs into more exact descriptions. Moreover, we develop the proposal generator and post-processing to produce temporal segments from debiased queries and image descriptions. Comprehensive experiments validate that VTG-GPT significantly surpasses current SOTA methods in zero-shot settings. Remarkably, it achieves a level of performance on par with supervised approaches.

6. Discussion

Limitations. In our study, constrained by computational resources, we downsample frames in the long-video dataset ActivityNet-Captions, which adversely affected perfor-

mance. Future work should focus on developing a more efficient and rapid GPT model to address this challenge. Moreover, due to the limitations imposed by the context length in video-based GPT, our framework relies solely on image-based GPT, thus needing more temporal information modeling.

In future work, we will explore applying video-based GPT (such as VideoChat-GPT [45]) to enhance the capabilities of zero-shot VTG. In addition, crafting a more efficient module for query debiasing and proposal generation is paramount. Finally, leveraging GPT to implement a zero-shot framework on other data-driven tasks (such as video summarization [13], depth estimation [46,47] and transformer diagnosis [48]) is very promising.

Ethical considerations. Our work is based on open-source LLMs and LMMs which require direct inference without training, thereby reducing the carbon footprint. Additionally, we utilize common and safe prompts, and have not observed the generation of harmful or offensive content by the model.

Author Contributions: Conceptualization, Y.X. and Y.S.; methodology, Y.X.; software, Y.X. and Y.S.; validation, Z.X. and B.Z.; formal analysis, Y.X.; investigation, Y.X.; resources, Y.S.; data curation, Y.X.; writing—original draft preparation, Y.X. and Y.S.; writing—review and editing, Z.X. and B.Z.; visualization, Z.X. and B.Z.; supervision, S.D.; project administration, S.D.; funding acquisition, S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/YoucanBaby/VTG-GPT.

Acknowledgments: Many thanks to Youyao Jia for his discussion and help in polishing this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lei, J.; Berg, T.L.; Bansal, M. Detecting Moments and Highlights in Videos via Natural Language Queries. *NeurIPS* **2021**, *34*, 11846–11858.
2. Diwan, A.; Peng, P.; Mooney, R. Zero-shot Video Moment Retrieval with Off-the-Shelf Models. In *Transfer Learning for Natural Language Processing Workshop*; PMLR: New Orleans, LA, USA, 2023; pp. 10–21.
3. Introducing ChatGPT. Available online: https://openai.com/blog/chatgpt (accessed on 1 December 2023).
4. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:2307.09288.
5. Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; et al. Baichuan 2: Open Large-scale Language Models. *arXiv* **2023**, arXiv:2309.10305.
6. Chen, J.; Zhu, D.; Shen, X.; Li, X.; Liu, Z.; Zhang, P.; Krishnamoorthi, R.; Chandra, V.; Xiong, Y.; Elhoseiny, M. MiniGPT-v2: Large language model as a unified interface for vision-language multi-task learning. *arXiv* **2023**, arXiv:2310.09478.
7. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual Instruction Tuning. *arXiv* **2023**, arXiv:2304.08485.
8. Ma, K.; Zang, X.; Feng, Z.; Fang, H.; Ban, C.; Wei, Y.; He, Z.; Li, Y.; Sun, H. LLaViLo: Boosting Video Moment Retrieval via Adapter-Based Multimodal Modeling. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, 2–6 October 2023; pp. 2798–2803.
9. Liu, Y.; Li, S.; Wu, Y.; Chen, C.W.; Shan, Y.; Qie, X. UMT: Unified Multi-modal Transformers for Joint Video Moment Retrieval and Highlight Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 3042–3051.
10. Xu, Y.; Sun, Y.; Li, Y.; Shi, Y.; Zhu, X.; Du, S. MH-DETR: Video Moment and Highlight Detection with Cross-modal Transformer. *arXiv* **2023**, arXiv:2305.00355.
11. Xu, Y.; Sun, Y.; Xie, Z.; Zhai, B.; Jia, Y.; Du, S. Query-Guided Refinement and Dynamic Spans Network for Video Highlight Detection and Temporal Grounding. *IJSWIS* **2023**, *19*, 20. [CrossRef]
12. Luo, D.; Huang, J.; Gong, S.; Jin, H.; Liu, Y. Towards Generalisable Video Moment Retrieval: Visual-Dynamic Injection to Image-Text Pre-Training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 23045–23055.

13. Lin, K.Q.; Zhang, P.; Chen, J.; Pramanick, S.; Gao, D.; Wang, A.J.; Yan, R.; Shou, M.Z. UniVTG: Towards Unified Video-Language Temporal Grounding. In Proceedings of the International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 2794–2804.
14. Sun, Y.; Xu, Y.; Xie, Z.; Shu, Y.; Du, S. GPTSee: Enhancing Moment Retrieval and Highlight Detection via Description-Based Similarity Features. *IEEE Signal Process. Lett.* **2023**, *31*, 521–525. [CrossRef]
15. Jang, J.; Park, J.; Kim, J.; Kwon, H.; Sohn, K. Knowing Where to Focus: Event-aware Transformer for Video Grounding. In Proceedings of the International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 13846–13856.
16. Zhang, S.; Peng, H.; Fu, J.; Luo, J. Learning 2d temporal adjacent networks for moment localization with natural language. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12870–12877.
17. Nam, J.; Ahn, D.; Kang, D.; Ha, S.J.; Choi, J. Zero-shot natural language video localization. In Proceedings of the International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 1470–1479.
18. Liu, D.; Qu, X.; Wang, Y.; Di, X.; Zou, K.; Cheng, Y.; Xu, Z.; Zhou, P. Unsupervised temporal video grounding with deep semantic clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 1683–1691.
19. Gao, J.; Xu, C. Learning Video Moment Retrieval Without a Single Annotated Video. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 1646–1657. [CrossRef]
20. Wang, G.; Wu, X.; Liu, Z.; Yan, J. Prompt-based Zero-shot Video Moment Retrieval. In Proceedings of the The 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 413–421. . [CrossRef]
21. Kim, D.; Park, J.; Lee, J.; Park, S.; Sohn, K. Language-free Training for Zero-shot Video Grounding. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 2539–2548.
22. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8748–8763.
23. Luo, D.; Huang, J.; Gong, S.; Jin, H.; Liu, Y. Zero-shot video moment retrieval from frozen vision-language models. *arXiv* **2023**, arXiv:2309.00661.
24. Wang, Y.; Li, K.; Li, Y.; He, Y.; Huang, B.; Zhao, Z.; Zhang, H.; Xu, J.; Liu, Y.; Wang, Z.; et al. InternVideo: General Video Foundation Models via Generative and Discriminative Learning. *arXiv* **2022**, arXiv:2212.03191.
25. Wattasseril, J.I.; Shekhar, S.; Döllner, J.; Trapp, M. Zero-Shot Video Moment Retrieval Using BLIP-Based Models. In Proceedings of the International Symposium on Visual Computing, Lake Tahoe, NV, USA, 16–18 October 2023; pp. 160–171.
26. Li, J.; Li, D.; Savarese, S.; Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv* **2023**, arXiv:2301.12597.
27. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
28. Yu, S.; Cho, J.; Yadav, P.; Bansal, M. Self-Chained Image-Language Model for Video Localization and Question Answering. *arXiv* **2023**, arXiv:2305.06988.
29. Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; Jiang, D. WizardLM: Empowering Large Language Models to Follow Complex Instructions. *arXiv* **2023**, arXiv:2304.12244.
30. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, Ma.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
31. Zhu, D.; Chen, J.; Shen, X.; Li, X.; Elhoseiny, M. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *arXiv* **2023**, arXiv:2304.10592.
32. Zheng, Y.; Mao, J.; Liu, Y.; Ye, Z.; Zhang, M.; Ma, S. Human behavior inspired machine reading comprehension. In Proceedings of the The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 425–434.
33. Tong, Z.; Song, Y.; Wang, J.; Wang, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 10078–10093.
34. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
35. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084.
36. Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. A Survey on Evaluation of Large Language Models. *arXiv* **2023**, arXiv:2307.03109.
37. Gao, J.; Sun, C.; Yang, Z.; Nevatia, R. Tall: Temporal activity localization via language query. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5267–5275.
38. Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; Carlos Niebles, J. Dense-captioning events in videos. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 706–715.
39. Sigurdsson, G.A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; Gupta, A. Hollywood in homes: Crowdsourcing data collection for activity understanding. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 510–526.

40. Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 961–970.
41. Zheng, M.; Huang, Y.; Chen, Q.; Liu, Y. Weakly supervised video moment localization with contrastive negative sample mining. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 3517–3525.
42. Zheng, M.; Huang, Y.; Chen, Q.; Peng, Y.; Liu, Y. Weakly supervised temporal sentence grounding with gaussian-based contrastive proposal learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
43. Kong, S.; Li, L.; Zhang, B.; Wang, W.; Jiang, B.; Yan, C.C.; Xu, C. Dynamic Contrastive Learning with Pseudo-samples Intervention for Weakly Supervised Joint Video MR and HD. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 538–546. [CrossRef]
44. Huang, Y.; Yang, L.; Sato, Y. Weakly supervised temporal sentence grounding with uncertainty-guided self-training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 18908–18918.
45. Maaz, M.; Rasheed, H.; Khan, S.; Khan, F.S. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. *arXiv* **2023**, arXiv:2306.05424.
46. Xu, Y.; Peng, C.; Li, M.; Li, Y.; Du, S. Pyramid Feature Attention Network for Monocular Depth Prediction. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; pp. 1–6.
47. Xu, Y.; Li, M.; Peng, C.; Li, Y.; Du, S. Dual Attention Feature Fusion Network for Monocular Depth Estimation. In Proceedings of the CAAI International Conference on Artificial Intelligence, Hangzhou, China, 5–6 June 2021; pp. 456–468.
48. Jiang, P.; Zhang, Z.; Dong, Z.; Yang, Y.; Pan, Z.; Yin, F.; Qian, M. Transient-steady state vibration characteristics and influencing factors under no-load closing conditions of converter transformers. *Int. J. Electr. Power Energy Syst.* **2024**, *155*, 109497. [CrossRef]