UniVTG: Towards Unified Video-Language Temporal Grounding

Kevin Qinghong Lin[1], Pengchuan Zhang[2], Joya Chen[1], Shraman Pramanick[3], Difei Gao[1], Alex Jinpeng Wang[1], Rui Yan[1], Mike Zheng Shou[1]✉

[1]Show Lab, National University of Singapore [2]Meta AI [3]Johns Hopkins University

Abstract

Video Temporal Grounding (VTG), which aims to ground target clips from videos (such as consecutive intervals or disjoint shots) according to custom language queries (e.g., sentences or words), is key for video browsing on social media. Most methods in this direction develop task-specific models that are trained with type-specific labels, such as moment retrieval (time interval) and highlight detection (worthiness curve), which limits their abilities to generalize to various VTG tasks and labels. In this paper, we propose to Unify the diverse VTG labels and tasks, dubbed UniVTG*, along three directions: Firstly, we revisit a wide range of VTG labels and tasks and define a unified formulation. Based on this, we develop data annotation schemes to create scalable pseudo supervision. Secondly, we develop an effective and flexible grounding model capable of addressing each task and making full use of each label. Lastly, thanks to the unified framework, we are able to unlock temporal grounding pretraining from large-scale diverse labels and develop stronger grounding abilities e.g., zero-shot grounding. Extensive experiments on three tasks (moment retrieval, highlight detection and video summarization) across seven datasets (QVHighlights, Charades-STA, TACoS, Ego4D, YouTube Highlights, TVSum, and QFVS) demonstrate the effectiveness and flexibility of our proposed framework. The codes are available at* https://github.com/showlab/UniVTG.

1. Introduction

With the increasing interest in sharing daily lives, video has emerged as the most informative yet diverse visual form on social media. These videos are collected in a variety of settings, including untrimmed instructional videos [28], and well-edited vlogs [18]. With massive scales and diverse video forms, automatically identifying relevant moments based on user queries has become a critical capability in the industry for efficient video browsing.

This significant demand has given rise to a number of video understanding tasks, including moment retrieval [66, 63, 30], highlight detection [52, 16, 56], and video summarization [14, 45, 42]. As depicted in Fig. 1, moment retrieval tends to localize consecutive temporal windows (interval-level) by giving natural sentences; highlight detection aims to pick out the key segment with highest worthiness (curve-level) that best reflects the video gist; video summarization collects a set of disjoint shots (point-level) to summarize the video, with general or user-specific queries. Despite task-specific datasets [10, 5, 46, 45] and models [66, 63, 56] have been developed, these tasks are typically studied separately. In general, these tasks share a common objective of grounding various scale clips based on customized user queries, which we refer to as Video Temporal Grounding (VTG).

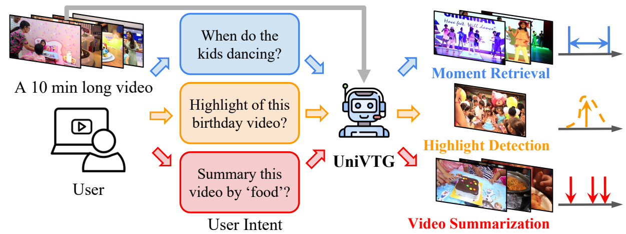


Figure 1: Given a video and a specific user query, UniVTG serves as a general video browsing helper that assists users by returning different scale target clips to support various VTG tasks.

Though these tasks are closely related, their relationship has not been explicitly studied until recently. [20] introduces the first unified benchmark QVHighlights for moment retrieval and highlight detection, and presents the first model Moment-DETR for jointly learning. On this basis, UMT [26] expands audio inputs, and QD-DETR [29] develops negative-pairs and saliency tokens. Nevertheless, these studies solely focus on designing models that intersect two subtasks and learn grounding capabilities rely on specific labels. This means that they lack the ability to generalize the VTG across diverse temporal labels, such as unique point-level narrations in Ego4D [13]. Furthermore, we have witnessed promising progress in Vision-Language Pretraining (VLP). One notable work is GLIP [23, 64], which develops a unified model via joint utilizing large-scale diverse image annotations such as image captions and bounding boxes for spatial grounding. However, we do not observe similar progress in video-language pretraining. Most works in this area are designed for video-level tasks such as video-text retrieval [54, 47] rather than temporal grounding. This

✉: Corresponding Author.

is largely due to the manual cost of fine-grained temporal annotations is expensive, making it challenging to obtain open-source, scalable yet diverse annotations to support grounding pretraining along the temporal axis in videos.

Therefore, we see a clear motivation to pursue a Unified VTG framework and propose our UniVTG, which aims to unify diversity in VTG along three directions: **(i) From the label and task aspect**, we first define a formulation for VTG where each video is decomposed as a clip sequence that each clip is assigned three basic query-conditional elements. Such a formulation enables us to unify various VTG labels and tasks under the same framework. Moreover, to address the limitation of temporal labels, we propose a data annotation scheme based on CLIP [36] to produce scalable fine-grained pseudo labels. **(ii) From the model aspect**, we develop a flexible yet effective grounding model that inherits the principles of our formulation. Our model devises single-stream and dual-stream pathways for modality fusion and modality alignment respectively, and is equipped with three heads to decode three key elements. This favorable design is capable of addressing each task and utilizing each label. **(iii)** Lastly, thanks to the unified framework and the availability of pseudo labels, we can perform **large-scale temporal grounding pretraining** across various labels to enhance our grounding abilities. This empowers us to address various VTG downstream tasks across multiple domains, including zero-shot inference.

To validate the effectiveness of our proposed framework, we conduct experiments not only on joint moment retrieval and highlight detection benchmark (QVHighlights [20]), but also on three individual tasks for moment retrieval (Ego4D [13], Charades-STA [10], TACoS [38]), highlight detection (YouTube Highlights [46], TVSum [45]) and video summarization (QFVS [42]). Our UniVTG, one unified model with 4.2M samples for temporal grounding pretraining, has achieved remarkable results, outperforming state-of-the-art methods that are specifically tailored for each task. Overall, our contributions are four folds:

- To the best of our knowledge, our UniVTG is the first video temporal grounding pretraining that across varied domains and tasks, including moment retrieval, highlight detection and video summarization.
- We introduce a unified VTG framework that can fully leverage rich supervision from open-source, scalable yet diverse temporal annotations, such as point-level, interval-level, and curve-level labels.
- To address the limitations of pretraining corpus, we develop an efficient annotation method that uses CLIP as a teacher to produce scalable pseudo temporal labels.
- We demonstrate the effectiveness and flexibility of the proposed framework across four settings and seven datasets. Detailed ablation studies validate the superiority of the proposed components.

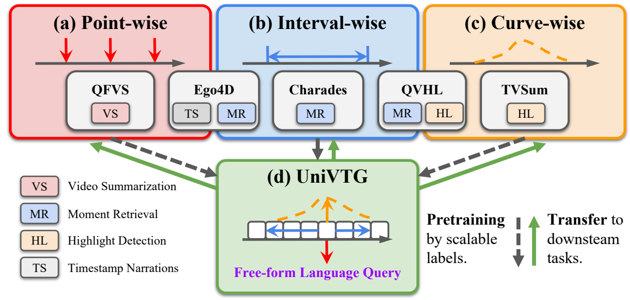


Figure 2: Diverse VTG labels can be divided into three types, each mainly associated with specific benchmarks: **(a) point-level labels** for video summarization [42] and timestamp narrations [13]; **(b) interval-level labels** for moment retrieval [13, 10, 20]; **(c) curve-level labels** for highlight detection [45, 20]. **(d) UniVTG** unifies diverse labels and tasks within one framework, enabling large-scale pretraining with diverse labels (*dotted gray line*) that can be transferred to various downstream tasks (*solid green line*).

2. Related Work

2.1. Video Temporal Grounding

We review three VTG tasks: moment retrieval, highlight detection, and video summarization, and compare them as different variations of a common problem.

Moment Retrieval aims to localize target moments *i.e.,* one [10] or many [20] continuous intervals within a video by a language query, as shown in Fig. 2 (b). Previous methods fall into two categories: proposal-based and proposal-free. The proposal-based methods [2, 10, 66] employ a two-stage process of scanning the entire video to generate candidate proposals, which are then ranked based on their matching to the text query. In contrast, the proposal-free methods [7, 61, 11, 63, 30] learn to regress the start and end boundaries directly without requiring proposal candidates. Our UniVTG borrows from proposal-free approaches but extends it by incorporating diverse temporal labels and tasks with a concise design.

Highlight Detection aims to assign a worthiness score to each video segment *e.g.,* Fig. 2 (c), and then return the top highest scoring segment as the highlight. Previous highlight detection datasets [40, 46, 45] tend to be domain-specific and query-agnostic, in which many efforts [15, 52, 16, 56, 3] treat this task as a visual or visual-audio scoring problem. Nevertheless, video highlights typically have a theme, which is often reflected in the video titles [45] or topics [46] *e.g.,* "surfing". Recently, [20] proposes a joint moment retrieval and highlight detection benchmark QVHighlights that enables users to produce various highlights for one video conditional on different text queries.

Video Summarization aims to summarize the whole video by a set of shots to provide a quick overview *e.g.,* Fig.2 (a), which contains two forms: Generic video summarization [14, 45, 27, 17] that captures the important scene using visual clues merely, while Query-focused video summariza-

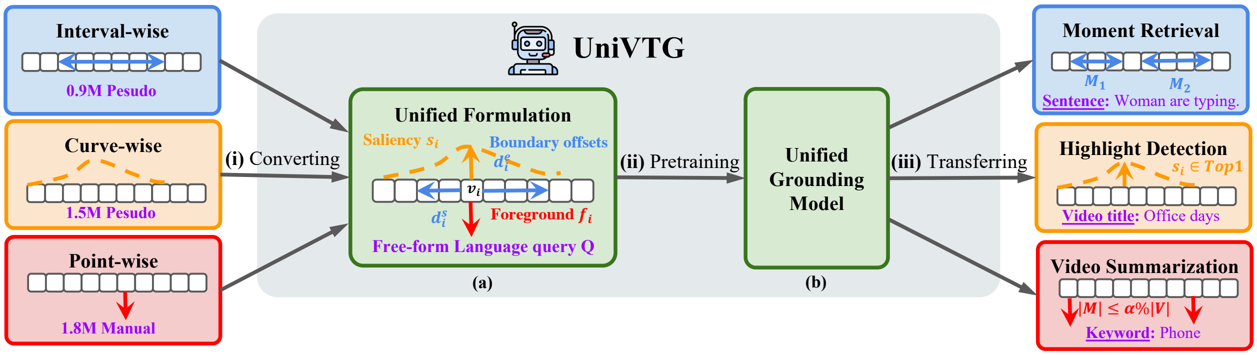


Figure 3: **Illustration of UniVTG pipeline.** **(i)** Given any kind of labels, such as interval label, we first convert it into our **(a) unified formulation** (§ 3.1) by deriving other two labels (point and curve labels). **(ii)** Once we have collect a large-scale diverse labels (§ 3.2), we leverage them to pretrain a **unified grounding model** (§ 4). **(iii)** Next, the unified model is transferred to various VTG downsteam tasks *e.g.,* highlight detection.

tion [42, 32, 49] that allows users to customize the summary by specifying text keywords (*e.g.,* tree and cars). The latter is closer to practical usage hence we focus on it. Recently, IntentVizor [49] proposes an interactive approach allowing users to adjust their intents to obtain a superior summary.

In general, each of the three tasks represents a specific form of VTG that grounds different scales of clips from videos (*e.g.,* a consecutive clip set, a single clip or a disjoint clip set) by offering customized text queries (*e.g.,* sentences, titles or keywords). However, previous methods address some subtasks solely. Based on this insight, our goal is to develop a unified framework to handle all of them.

2.2. Vision-Language Pretraining

The emergence of large-scale vision-language datasets, such as [43, 41, 28, 4], has paved the way for the development of VLP [36, 22, 19, 34, 21] to enhance video-text representation for various vision-language tasks [60, 54, 53]. The representative CLIP [36] has shown that *image-level* visual representations can be effectively learned using large-scale noisy image-text pairs. Furthermore, GLIP [23, 64] makes an effort along the spatial axis, which leverages various image annotations, such as image labels, captions, and bounding boxes, to develop strong *region-level* understanding capacity for spatial grounding tasks. However, due to the expensive manual cost of fine-grained *temporal-level* annotations *i.e.,* temporal bounding box, this grounding pretraining has not been extended to the temporal axis in videos, limiting its progress to match the spatial counterparts. To address this limitation, we explore alternative approaches that leverage accessible timestamp narrations [13] and derive pseudo supervision as the pretraining corpus.

On the other hand, there are several efforts have been made to perform temporal-friendly video pretraining [1, 55, 6, 62] to pursue a better video representation for grounding tasks. But the resulting pretraining model still requires an additional grounding model such as 2D-TAN [66] to perform video grounding. In contrast, powered by our unified framework and scalable pseudo annotations, we can directly conduct VLP with grounding as a pretraining task. This way eliminates the need for additional grounding models and enables zero-shot grounding capacity.

3. Towards Unified VTG: Tasks and Labels

The UniVTG pipeline is displayed in Fig. 3. In this section, we start by introducing the unified formulation.

3.1. Unified Formulation

Given a video V and a language query Q, we first divide V into a sequence of L_v fixed-length clips $\{v_1, \cdots, v_{L_v}\}$, where each clip v_i is of length l and has a centered timestamp t_i. The free-form text query Q has L_q tokens, denoted as $Q = \{q_1, \cdots, q_{L_q}\}$. We then define three elements for each clip $v_i = (f_i, d_i, s_i)$, described as follows:

- **Foreground indicator** $f_i \in \{0, 1\}$: a binary value indicating whether the i-th clip v_i belongs to the foreground or not. If clip v_i is the foreground of Q, then $f_i = 1$, otherwise $f_i = 0$.
- **Boundary offsets** $d_i = [d_i^s, d_i^e] \in \mathbb{R}^2$: the temporal distance that converts the clip timestamp t_i to its interval boundaries. Here, d_i is valid when $f_i = 1$. The d_i^s is the distance between the starting of the interval and t_i, whereas d_i^e is the distance between the ending and t_i. Thus, the whole temporal interval b_i of v_i can be represented as $b_i = [t_i - d_i^s, t_i + d_i^e]$
- **Saliency score** $s_i \in [0, 1]$: a continuous score determining the relevance between the visual content of clip v_i and the query Q. If the clip and query are highly correlated, $s_i = 1$; If they are totally irrelevant, then $s_i = 0$. Notably, it is reasonable to assume that $s_i > 0$ if a clip is in the foreground of Q, otherwise $s_i = 0$.

In Fig.3 (a), we draw a schematic diagram to represent these three elements of clip v_i in our definition.

3.2. Revisiting Various VTG Tasks and Labels

Treating clips as the atom composition of a video, we define the VTG problem as collecting a target clip set $M =$

$\{v_i \in V|Q\}$ from V, conditional on language query Q. We next illustrate how to extend this definition to various tasks and labels. Especially, for each label, we answer:

1. How to collect scalable label corpus for pretraining?
2. When using the unified formulation, how can we obtain unknown elements based on the available one?

3.2.1 Moment Retrieval and Interval-wise Label.

Moment retrieval aims to localize one [10] or many [20] intervals in a video corresponding to *a sentence* Q. As shown in Fig. 3 (Right blue), moment retrieval aims to select m consecutive clip sets $M = M_1 \cup \cdots \cup M_m$, where $m \geq 1$, and M_j is the j-th target moment. M can be simplified as the boundary set of foreground clips $\{b_i|f_i = 1\}$.

The temporal interval with specific target boundaries is a common label for moment retrieval. However, annotating intervals requires manually reviewing the full video, which is expensive. A solution is ASR [28, 57] that provide start and end timestamps, but ASR is often too noisy and poorly aligned with the visual content, making it suboptimal. Here, we sought an alternative solution. We found that visual captions [43, 4] tend to be descriptive, making them well-suited as grounding queries, thus if we can know how these videos are cut from the raw source, we can use this information to create pseudo intervals. We find that VideoCC [31] is a viable option for this purpose. It is worth noting that VideoCC is initially developed for video-level pretraining (*e.g.,* power video-text retrieval), and we are the pioneer to investigate its potential in temporal grounding pretraining.

Once we obtain intervals, we convert interval labels into the proposed formulation by defining $f_i = 0$ and $s_i = 0$ for clips that are not in target interval, and we assign $f_i = 1$ and assume $s_i > 0$ for clips that belongs to the target interval.

3.2.2 Highlight Detection and Curve-wise Label.

Highlight detection aims to assign an importance score to each video clip (making its annotations like a curve), then return the few highest-scoring clips as the highlight, where queries may [20] or may not [46, 45] be provided as input. For video highlighting datasets without language queries, we can use *video titles* [45] or *video domain name* [46] as Q because they are highly related to the topic of the video. Then, this task is equivalent to picking clips with the top highest saliency scores *i.e.* $M = \{v_i|s_i \in \text{top-}K\}$.

Due to the interestingness contain subjectivity, the same video usually needs to be labeled by several people to eliminate bias. This makes curve labels the most expensive yet informative temporal annotations. Therefore, we are motivated to find an efficient way of producing scalable curve labels. Intuitively, interestingness reflects how each clip is relevant to the video gist. As depicted in Fig. 4 (a), we first define a concept bank using an open-world detection class list [41]. Next, we use CLIP as a teacher to get the clip-level cosine similarities between each concept. Then, we select top-5 concepts as the video gist, and save their CLIP similarities as pseudo curve labels, *i.e.,* Fig. 4 (b).

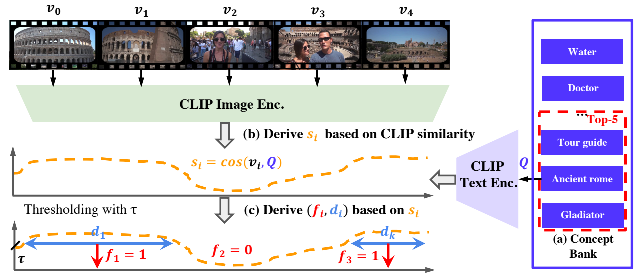


Figure 4: **Process of using CLIP to produce temporal labels. (a)** We first use a concept bank to cover diverse open-world concepts. **(b)** Next, we use CLIP as teacher to calculate the clip-level scores between each concept to get top-5 concepts as video gist, and treat their clip scores as saliency s_i. **(c)** Based on s_i, we further derive the interval and point labels via thresholding.

As shown in Fig. 4 (c), after obtaining curve labels, we assign $f_i = 1$ for clips with s_i greater than a threshold τ, otherwise $f_i = 0$. The τ is estimated based on the similarity of each video, refer to Supp. for details. The offsets d_i are defined as the distance between the foreground clip and its nearest neighboring clips where $f_i = 0$.

3.2.3 Video Summarization and Point-wise Label.

Query-focused video summarization [42] aims to summarize the entire video with a set of shots to provide a quick overview, with user-specific concepts (for example, trees and cars). The generated summary should be succinct while representative of the entire video around the given query. We define this task by regarding *keywords* as Q, and select a set of clips $M = \{v_i|f_i = 1\}$, where the size of M is required to not exceed $\alpha\%$ of the original video length $|M| \leq \alpha\%|V|$ *e.g.,* $\alpha = 2\%$.

The annotations in QFVS [42] are point labels that indicate whether each shot belongs to the concept or not. The cost of point labels is much cheaper than that of interval and curve labels since people only need to glance at a specific time. The recently Ego4D [13] dataset uses this point labeling to annotate massive-scale data by assigning a narration to an exact timestamp, such as "I am opening the washing-machine" at $t_i = 2.30$ sec. Due to the favorable scale, it is natural to adapt them for large-scale pretraining. Recently, there have been attempts to improve video-text representation using point-wise annotations to improve the video-text representation [24, 67, 35] and augment NLQ [13] baselines [37]. Despite this, these methods mainly focus on transferring within the same domain.

For point labels, we derive $s_i > 0$ if clip $f_i = 1$, otherwise $s_i = 0$. During pretraining, we estimate its temporal label b_i based on the average distance between consecutive narrations within the video [24, 37, 35].

4. Towards Unified VTG: Model

We here introduce our unified model which seamlessly inherits our proposed unified formulation.

4.1. Overview

As shown in Fig. 5, our model mainly comprises a frozen video encoder, a frozen text encoder, and a multi-modal encoder. The video and text encoders are keep consistent with Moment-DETR [18], which employs the concatenation of CLIP [36] (ViT-B/32) and SlowFast [9] (R-50) features as video representation, and use the CLIP text encoder [36] to extract token level features. Our multi-modal encoder contains k self-attention blocks that followed by three specific heads to decode the prediction.

Given an input video V with L_v clips and a language query Q with L_q tokens, we first apply the video encoder and the text encoder to encode the video and text respectively, then project them to the same dimension D by two Feed-Forward Networks (FFN), and thus obtain video features $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{L_v} \in \mathbb{R}^{L_v \times D}$ and text features $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^{L_q} \in \mathbb{R}^{L_q \times D}$. Next, we design two pathways for cross-modal alignment and cross-modal interaction.

(i) For cross-modal alignment, we first adopt an attentive pooling operator to aggregate the query tokens $\mathbf{Q} \in \mathbb{R}^{L_q \times D}$ into a sentence representation $\mathbf{S} \in \mathbb{R}^{1 \times D}$. Especially,

$$\mathbf{S} = \mathbf{A}\mathbf{Q}, \tag{1}$$

where the weight $\mathbf{A} = \text{Softmax}\left(\mathbf{W}\mathbf{Q}\right) \in \mathbb{R}^{1 \times L_q}$ and $\mathbf{W}^{1 \times L_q}$ is a learnable embedding. Then $\mathbf{V}$ and $\mathbf{S}$ are sent to perform contrastive learning (described in § 4.2).

(ii) For cross-modal interaction, learnable position embeddings $\mathbf{E}^{pos}$ and modality-type embeddings $\mathbf{E}^{type}$ are added to each modality to retain both positional and modality information:

$$\begin{aligned} \tilde{\mathbf{V}} &= \mathbf{V} + \mathbf{E}_V^{pos} + \mathbf{E}_V^{type}, \\ \tilde{\mathbf{Q}} &= \mathbf{Q} + \mathbf{E}_T^{pos} + \mathbf{E}_T^{type}. \end{aligned} \tag{2}$$

Next, the text and video tokens are concatenated and get a joint input $\mathbf{Z}^0 = [\tilde{\mathbf{V}}; \tilde{\mathbf{Q}}] \in \mathbb{R}^{L \times D}$, where $L = L_v + L_q$. Further, $\mathbf{Z}^0$ is fed into the multi-modal encoder, which contains k transformer layers with each layer consisting of a Multi-headed Self-Attention and FFN blocks.

$$\mathbf{Z}^d = \text{MLP}\left(\text{MSA}\left(\mathbf{Z}^{d-1}\right)\right), \quad d \in \{1 \ldots k\}. \tag{3}$$

We take the video tokens $\tilde{\mathbf{V}}^k \in \mathbb{R}^{L_v \times D}$ from the multi-modal encoder E_m as output $\mathbf{Z}^k = [\tilde{\mathbf{V}}^k; \tilde{\mathbf{Q}}^k] \in \mathbb{R}$, and feed $\mathbf{Z}^k$ into the following heads for prediction.

4.2. Pretraining Objectives

To match the previous unified formulation *i.e.,* (f_i, d_i, s_i), we devise three different heads to decode each element respectively, each one calling a capability.

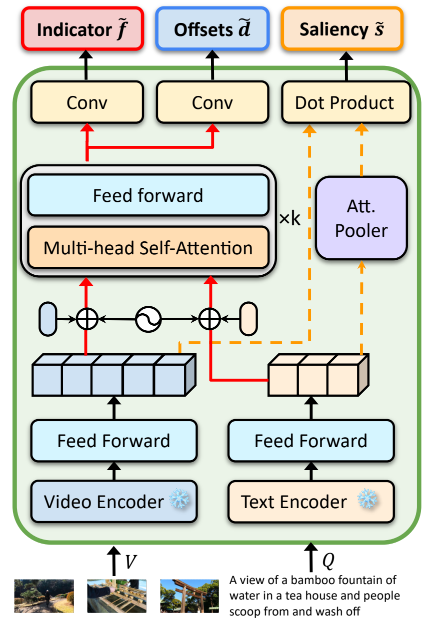


Figure 5: **Unified grounding model** contains a video encoder, a text encoder, and a multi-modal encoder followed by three output heads, corresponding to three key elements $\left(\tilde{f}_i, \tilde{d}_i, \tilde{s}_i\right)$. Besides, our model has two pathways: one for cross-modal interaction (solid red line) and the other for cross-modal alignment (broken orange line).

Foreground head for Matching. Taking the output $\tilde{\mathbf{V}}^k \in \mathbb{R}^{L_v \times D}$ from the multi-modal encoder, this head applies three 1×3 Conv layers, each with D filters and followed by a ReLU activation. Finally, sigmoid activations are attached to output the prediction $\tilde{f}_i$ per clip. We use the binary cross-entropy loss as a training objective.

$$\mathcal{L}_{\text{f}} = -\lambda_{\text{f}}\left(f_i \log \tilde{f}_i + (1 - f_i)\log\left(1 - \tilde{f}_i\right)\right). \tag{4}$$

Boundary head for Localization. The design of this head is similar to the foreground head except for the last layer, which has 2 outputs channel for the left and right offsets. Taking the $\tilde{\mathbf{V}}^k \in \mathbb{R}^{L_v \times D}$, this head outputs offsets $\{\tilde{d}_i\}_i^{L_v}$ per clip. Then, we devise the predicted boundary $\tilde{b}_i$ and use the combination of smooth $L1$ loss [12] and generalized IoU loss [39] as our training objectives.

$$\mathcal{L}_{\text{b}} = \mathbb{1}_{f_i=1}\left[\lambda_{\text{L1}}\mathcal{L}_{\text{SmoothL1}}\left(\tilde{d}_i, d_i\right) + \lambda_{\text{iou}}\mathcal{L}_{\text{iou}}\left(\tilde{b}_i, b_i\right)\right]. \tag{5}$$

Notably, this regression objective is only devised for foreground clips *i.e.,* $f_i = 1$.

Saliency head for Contrasting. Since we define saliency as the relevance between visual context and text query, it is natural to interpret this score as a similar measurement between video and text modalities. Taking the video tokens $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{L_v} \in \mathbb{R}^{L_v \times D}$ and sentence representation $\mathbf{S} \in \mathbb{R}^{1 \times D}$, we define the predicted saliency score $\tilde{s}_i$ between clip v_i and text query Q as their cosine similarities:

$$\tilde{s}_i = \cos(\mathbf{v}_i, \mathbf{S}) := \frac{\mathbf{v}_i^T \mathbf{S}}{\|\mathbf{v}_i\|_2 \|\mathbf{S}\|_2}, \tag{6}$$

where $\|\cdot\|_2$ represents the $L2$-norm of a vector.

For each video $\mathbf{V}$, we randomly sample a foreground clip $\mathbf{v}_p$ with $f_p = 1$ and $s_p > 0$ as a positive sample; we treat other clips in the same video $\mathbf{v}_j$ with saliency s_j less than s_p as negative samples, *i.e.*, $\Omega = \{j | s_j < s_p, 1 \leq j \leq L_v\}$, and perform **intra-video** contrastive learning:

$$\mathcal{L}_{\text{s}}^{\text{intra}} = -\log \frac{\exp\left(\tilde{s}_p/\tau\right)}{\exp\left(\tilde{s}_p/\tau\right) + \sum_{j\in\Omega} \exp\left(\tilde{s}_j/\tau\right)}, \tag{7}$$

where τ is a temperature parameter and set as 0.07.

Besides, we regard sentences from other samples within batches $k \in B$ as negative samples, and develop the **inter-video** contrastive learning for cross-sample supervision:

$$\mathcal{L}_{\text{s}}^{\text{inter}} = -\log \frac{\exp\left(\tilde{s}_p/\tau\right)}{\sum_{k\in B} \exp\left(\tilde{s}_p^k/\tau\right)}, \tag{8}$$

where B is the training batch size and $\tilde{s}_p^k = \cos(\mathbf{v}_i, \mathbf{S}_k)$.

Our saliency score head training loss is the combination of inter- and intra-video contrastive learning:

$$\mathcal{L}_{\text{s}} = \lambda_{\text{inter}} \mathcal{L}_{\text{s}}^{\text{inter}} + \lambda_{\text{intra}} \mathcal{L}_{\text{s}}^{\text{intra}}. \tag{9}$$

To this end, our total training objective is the combination of each head loss overall clips in the training set.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{L}_{\text{f}} + \mathcal{L}_{\text{b}} + \mathcal{L}_{\text{s}}\right), \tag{10}$$

where N is the clip number of the training set.

4.3. Inference

During inference, given a video V and a language query Q, we first feed forward the model to obtain $\{\tilde{f}_i, \tilde{b}_i, \tilde{s}_i\}_{i=1}^{L_v}$ for each clip v_i from three heads. Next, we describe how we carry out output for individual VTG tasks respectively.

Moment Retrieval. We rank clips predicted boundaries $\{\tilde{b}_i\}_{i=1}^{L_v}$ based on their $\{\tilde{f}_i\}_{i=1}^{L_v}$ probabilities. Since the predicted L_v boundaries are dense, we adopt a 1-d Non-Max Suppression (NMS) with a threshold 0.7 to remove highly overlapping boundary boxes, yielding a final prediction.

Highlight Detection. For each clip, to fully utilize the foreground and saliency terms, we rank all clips based on their $\{\tilde{f}_i + \tilde{s}_i\}_{i=1}^{L_v}$ scores, and then return the few top clip (*e.g.*, Top-1) as predictions.

Video Summarization. Using the same preprocessing settings [42, 51], the videos are first divided as multiple segments via KTS algorithm [33]. Then the clip scores from each segment are computed, and these scores are integrated. We rank all clips based on their foreground $\{\tilde{f}_i\}_{i=1}^{L_v}$ and return the Top-2% clips as a video summary.

Dataset	Task	Pseudo?	Label	# Samples	Domain
Ego4D [13]	PT	✗	Point	1.8M	Egocentric
VideoCC [31]	PT	✓	Interval	0.9M	Web
CLIP teacher	PT	✓	Curve	1.5M	Open
QVHighlights [18]	MR + HL	✗	Interval + Curve	10.3K	VLog, News
NLQ [13]	MR	✗	Interval	15.1K	Egocentric
Charades-STA [10]	MR	✗	Interval	16.1K	Indoor
TACoS [38]	MR	✗	Interval	18.2K	Kitchens
YoutubeHL [46]	HL	✗	Curve	600	Web
TVSum [45]	HL	✗	Curve	50	Web
QFVS [42]	VS	✗	Point	4	Egocentric

Table 1: **Dataset statistics. The upper side** datasets are used for pretraining (PT) which cover three label types, two of which are pseudo. **The lower side** datasets are used for downstream tasks (MR: Moment Retrieval, HL: Highlight Detection, VS: Video Summarization).

5. Experiments

In this section, we conduct experiments on various benchmarks to evaluate our approach. Mainly, we design the experiments to study the following questions:

Q1: How much improvement could be made by UniVTG grounding pretraining?

Q2: What are the effects of using different pretraining corpus from various labels?

Q3: Is it necessary to use the proposed unified formulation and unified model?

More ablation studies can be found in Supplementary.

5.1. Datasets and Settings

We have summarized the dataset information in Tab.1. For pretraining, we gather 1.8M point labels from Ego4D and 0.9M interval labels from VideoCC [31]. For curve labels, we apply CLIP teacher method (Fig. 4) to Ego4D and VideoCC datasets to get 1.5M pseudo labels. Therefore, a total of 4.2M temporal annotations are used for grounding pretraining. For downstream tasks, we assess our methods on four VTG tasks across seven datasets, spanning **(i)** Jointly moment retrieval and highlight detection; **(ii)** Moment Retrieval; **(iii)** Highlight Detection; **(iv)** Video Summarization. Additional details are listed in Supp.

Evaluation Metrics. For QVHighlights, we follow official [20], Recall@1 with IoU thresholds 0.5 and 0.7, mean average precision (mAP) with IoU thresholds 0.5 and 0.75, and the average mAP over a series of IoU thresholds [0.5:0.05:0.95] are used for moment retrieval. For highlight detection, mAP and HIT@1 are used, a clip is treated as a true positive if it has the saliency score of Very Good. For Charades-STA, NLQ, TACoS, Recall@1 with IoU thresholds 0.3, 0.5 and 0.7, and mIoU are used. For YouTube Highlights and TVSum, we follow [26] and use mAP and Top-5 mAP, respectively. For QFVS, we follow [49] that reports F1-score per video as well as an average.

Implementation Details. We set $k = 4$ multi-modal transformer encoder layers, with $d = 1024$ hidden size and 8 attention heads. The drop path rates are 0.1 for trans-

Method	Moment Retrieval					HD	
	R1		mAP			≥ Very Good	
	@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
BeautyThumb [44]	–	–	–	–	–	14.36	20.88
DVSE [25]	–	–	–	–	–	18.75	21.79
MCN [2]	11.41	2.72	24.94	8.22	10.67	–	–
CAL [8]	25.49	11.54	23.40	7.65	9.89	–	–
CLIP [36]	16.88	5.19	18.11	7.0	7.67	31.30	61.04
XML [20]	41.83	30.35	44.63	31.73	32.14	34.49	55.25
XML+ [18]	46.69	33.46	47.89	34.67	34.90	35.38	55.06
MDETR [18]	52.89	33.02	54.82	29.40	30.73	35.69	55.60
MDETR w/ PT	59.78	40.33	60.51	35.36	36.14	37.43	60.17
UMT†[26]	56.23	41.18	53.83	37.01	36.12	38.18	59.99
UMT† w/ PT	60.83	43.26	57.33	39.12	38.08	39.12	62.39
UniVTG	58.86	40.86	57.60	35.59	35.47	38.20	60.96
UniVTG w/ PT	**65.43**	**50.06**	**64.06**	**45.02**	**43.63**	**40.54**	**66.28**
UniVTG ZS	25.16	8.95	27.42	7.64	10.87	35.96	53.50

Table 2: **Jointly Moment Retrieval and Highlight Detection results on QVHighlights test split**[1]**.** †: introduce audio modality. w/ PT: fine-tuning after pre-training; ZS: zero-shot inference.

former layers and 0.5 for input FFN projectors. During the pretraining stage, our experiments are carried out on 8 A100 GPUs. When it comes to downstream tasks, we use one GPU. For moment retrieval, all baselines and UniVTG use the same video and text features. For highlight detection and video summarization, we report results following [26] and [49]. See Supp. for more details.

5.2. Comparison with State-of-the-arts (Q1)

5.2.1 Joint Moment Retrieval and Highlight Detection

As illustrated in Tab. 2, we first evaluate our UniVTG on QVHighlights test split: (i) Without pretraining, UniVTG has shown comparable performance to two joint optimization counterparts Moment-DETR [18] and UMT [26], demonstrating its superior model design for joint task optimization. (ii) With large-scale pretraining, UniVTG exhibits a significant improvement on all metrics, such as +8.16 Avg. mAP and +5.32 HIT@1. As a result, UniVTG surpasses all baselines by a large margin. Notably, UMT introduces audio modality and ASR pretraining [26], but it is still worse than us by Avg. mAP of 5.55 and HIT@1 of 3.89. (iii) Due to the large-scale pretraining, UniVTG can perform zero-shot grounding and outperforms several supervised baselines without any training samples.

[1] Codalab QVHighlights Evaluation

5.2.2 Moment Retrieval

In Tab. 3, we compare the results of our method and the mainstream moment retrieval methods on three widely used benchmarks. (i) Similar to the observation made by QVHighlights, without pretraining, we find that UniVTG is still superior to other compared methods. This demonstrates once more the effectiveness of our concise architecture. (ii) Large-scale grounding pretraining has resulted in significant improvements, leading to a considerable increase in the mIoU *i.e.,* +2.97 in NLQ, +2.07 in Charades-STA, and +5.03 in TACoS. (iii) Notably, in NLQ, our zero-shot result has outperformed all the baselines methods due to the close pretraining domain. However, it is worth mentioning that the zero-shot performance on TACoS is inferior. This could be because the videos have scenes that are very similar to each other, with only small spatial variations, making it difficult to effectively apply zero-shot methods.

5.2.3 Highlight Detection

In Tab. 4 and Tab. 5, we conduct highlight detection experiments on YouTube Highlights and TVSum respectively, where the baselines with † (rows 6-9) are incorporate with audio features. We observe that (i) grounding pretraining brings improvement on UniVTG and surpasses all baselines in Avg. mAP. (ii) In TVSum, gain discrepancy among domains may stem from its small scale (50 samples) and scoring subjectivity. In contrast, the larger YouTube dataset (600 videos) yields more consistent pretraining gains. (ii) Moreover, in zero-shot setting, UniVTG beats several video-only baselines such as [46, 48].

5.2.4 Video Summarization

In Tab. 6, we present the QFVS benchmark results. Our pretrained UniVTG achieves a 0.8% higher Avg. F1-score than IntentVizor [49], where the latter is an interactive method and being tailored for the video summarization task. This result demonstrates the generalization of our method on video summarization task.

Method	NLQ [13]				Charades-STA [10]				TACoS [38]			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
2D TAN [66]	4.33	1.83	0.60	3.39	58.76	46.02	27.50	41.25	40.01	27.99	12.92	27.22
VSLNet [63]	4.54	2.40	1.01	3.54	60.30	42.69	24.14	41.58	35.54	23.54	13.15	24.99
MDETR [18]	4.34	1.81	0.65	3.53	65.83	52.07	30.59	45.54	37.97	24.67	11.97	25.49
UniVTG	7.28	3.95	1.32	4.91	70.81	58.01	35.65	50.10	51.44	34.97	17.35	33.60
UniVTG w/ PT	**11.74**	**7.54**	**3.25**	**7.88**	**72.63**	**60.19**	**38.55**	**52.17**	**56.11**	**43.44**	**24.27**	**38.63**
UniVTG ZS	6.48	3.48	1.16	4.63	44.09	25.22	10.03	27.12	5.17	1.27	0.27	4.40

Table 3: **Moment Retrieval results on NLQ, Charades-STA, and TACoS benchmarks.** All baselines use the same video features (CLIP ViT-B/32 and SlowFast R-50) and text features (CLIP text enc.). w/ PT means fine-tuning after pre-training; ZS means zero-shot inference.

Method	Dog	Gym.	Par.	Ska.	Ski.	Sur.	**Avg.**
RRAE [58]	49.0	35.0	50.0	25.0	22.0	49.0	38.3
GIFs [15]	30.8	33.5	54.0	55.4	32.8	54.1	46.4
LSVM [46]	60.0	41.0	61.0	62.0	36.0	61.0	53.6
LIM-S [52]	57.9	41.7	67.0	57.8	48.6	65.1	56.4
SL-Module [56]	70.8	53.2	77.2	72.5	66.1	76.2	69.3
MINI-Net† [16]	58.2	61.7	70.2	72.2	58.7	65.1	64.4
TCG† [59]	55.4	62.7	70.9	69.1	60.1	59.8	63.0
Joint-VA† [3]	64.5	71.9	80.8	62.0	73.2	78.3	71.8
UMT†[26]	65.9	75.2	**81.6**	71.8	72.3	82.7	74.9
UniVTG	71.8	76.5	73.9	73.3	73.2	82.2	75.2
UniVTG w/ PT	**74.3**	**79.0**	74.4	**84.9**	**75.1**	**83.9**	**78.6**
UniVTG ZS	36.8	62.8	65.9	39.2	64.5	54.0	53.9

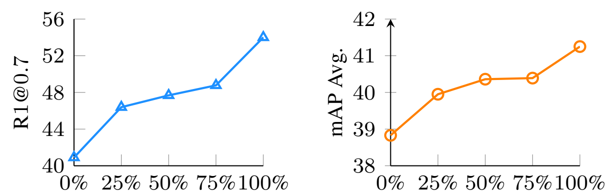
Table 4: **Highlight Detection results of mAP on YouTube HL.** † denotes using audio modality.

Method	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS	**Avg.**
sLSTM [65]	41.1	46.2	46.3	47.7	44.8	46.1	45.2	40.6	47.1	45.5	45.1
SG [27]	42.3	47.2	47.5	48.9	45.6	47.3	46.4	41.7	48.3	46.6	46.2
LIM-S [52]	55.9	42.9	61.2	54.0	60.4	47.5	43.2	66.3	69.1	62.6	56.3
Trailer [48]	61.3	54.6	65.7	60.8	59.1	70.1	58.2	64.7	65.6	68.1	62.8
SL-Module [56]	86.5	68.7	74.9	**86.2**	79.0	63.2	58.9	72.6	78.9	64.0	73.3
MINI-Net† [16]	80.6	68.3	78.2	81.8	78.1	65.8	57.8	75.0	80.2	65.5	73.2
TCG† [59]	85.0	71.4	81.9	78.6	80.2	75.5	71.6	77.3	78.6	68.1	76.8
Joint-VA† [3]	83.7	57.3	78.5	86.1	80.1	69.2	70.0	73.0	**97.4**	67.5	76.3
UMT†[26]	87.5	81.5	88.2	78.8	81.5	**87.0**	**76.0**	86.9	84.4	**79.6**	83.1
UniVTG	83.9	**85.1**	89.0	80.1	**84.6**	81.4	70.9	91.7	73.5	69.3	81.0
UniVTG w/ PT	**92.0**	77.8	**89.8**	83.8	82.2	85.8	74.3	**91.8**	90.5	77.6	**84.6**
UniVTG ZS	78.5	67.0	75.3	63.6	67.0	66.8	35.4	85.3	83.1	50.0	67.2

Table 5: **Highlight Detection results of Top-5 mAP on TVSum.** † denotes using audio modality.

Method	V1	V2	V3	V4	**Avg.**
QC-DPP [42]	48.68	41.66	36.51	29.96	44.19
CHAN [51]	49.14	46.53	58.65	33.42	46.94
QSAN [50]	48.52	46.64	56.93	34.25	46.59
WHM [32]	50.96	48.28	58.41	39.18	49.20
IntentVizor [49]	51.27	53.48	**61.58**	37.25	50.90
UniVTG	**52.54**	54.48	56.73	40.37	51.03
UniVTG w/ PT	49.85	**56.97**	59.35	**40.62**	**51.70**

Table 6: **Video Summarization results of F-score on QFVS.**



(a) R1@0.7 of Moment Retrieval. (b) mAP Avg. of Highlight Detection.

Figure 6: **Effect of pretraining scale on QVHighlights dataset.**

5.3. Ablation Studies

Effect of different labels for pretraining (Q2). In Tab. 7 top half, we investigate the effect of different labels corpus for pretraining. The results here are before unified formulation *i.e.,* the original label provided by the pretraining set. Our findings (rows 1-4) indicate that (i) incorporating any type of label for pretraining yields considerable performance gains on most benchmarks. (ii) Combining all three types of data (row 5) for pretraining further boost the outcomes, such as +5.2 MR's mAP and +1.1 HL's mAP over baseline (row 1) on QVHighlights.

Effect of unified formulation (Q3). In Tab. 7 bottom half, we further study the impacts of unified formulation *i.e.,* the benefits of deriving unknown elements for pretraining. From rows 2-4 vs rows 6-8, We find that (i) training corpora receive performance gains in most settings, which proves that the label converting methods are crucial for better utilizing temporal labels. (ii) Among all settings, curve labels appear to be the most effective ones, and beat the manual point labels except in a few domains *e.g.,* TACoS. (iii) We get the optimal result (row 9) by using full three converted corpus for pretraining, with 4.62 MR's mAP and 1.28 HL's mAP increase over counterparts (row 5) on QVHighlights.

Effect or pretraining scale. In Fig. 6, we explore the effect of utilizing various scales of labels for pretraining. We observe a steady performance improvement on both moment retrieval and highlight detection tasks as the training sample size increases. It also shows that unifying labels to construct a large training corpus can greatly benefit the VTG.

row	**Pretraining Corpus** Ego4D Point	VideoCC Interval	CLIP Curve	**Unified Labels?** Point	Interval	Curve	**QVHighlights** MR mAP	HL mAP	**TACoS** MR mIoU	**YouTube** HL mAP
1							36.13	38.83	33.60	75.15
2	✓			✓			39.89	39.48	35.33	75.32
3		✓			✓		39.81	39.75	35.11	74.76
4			✓			✓	39.16	39.80	35.68	75.44
5	✓	✓	✓	✓	✓	✓	41.37	39.97	35.87	75.66
6	✓			✓	$\underline{\checkmark}$	$\underline{\checkmark}$	41.53	39.66	36.52	75.27
7		✓		$\underline{\checkmark}$	✓	$\underline{\checkmark}$	40.96	40.10	36.78	76.10
8			✓	$\underline{\checkmark}$	$\underline{\checkmark}$	✓	42.19	40.43	35.85	77.48
9	✓	✓	✓	$\underline{\checkmark}$	$\underline{\checkmark}$	$\underline{\checkmark}$	**45.99**	**41.25**	**38.63**	**78.56**

Table 7: **Ablation studies on pretraining corpus.** $\underline{\checkmark}$ denotes the elements derived by us, which are not provided in vanilla training corpus: Ego4D, VideoCC, and CLIP teacher.

6. Conclusion

This paper introduces UniVTG, a framework that unifies diverse VTG tasks and labels by addressing three key challenges: (i) We define a unified formulation for VTG to convert various labels and tasks under a single framework, and propose a label scaling scheme. (ii) We develop an effective yet flexible model that can handle various VTG tasks and training labels. (iii) Due to the unified framework and availability of scalable labels, it becomes feasible to perform large-scale temporal grounding pretraining over diverse labels. We demonstrate the effectiveness and flexibility of our UniVTG on four settings across seven datasets, spanning joint optimization as well as individual tasks.

7. Acknowledgements

This project is supported by the National Research Foundation, Singapore under its NRFF Award NRF-NRFF13-2021-0008, the DSO National Laboratories, Mike Zheng Shou's Start-Up Grant from NUS. The computational work for this article was partially performed on resources of the National Super computing Centre, Singapore.

References

[1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *ICCV*, pages 3173–3183, 2021.

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, pages 5803–5812, 2017.

[3] Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Joint visual and audio learning for video highlight detection. In *ICCV*, pages 8127–8137, 2021.

[4] 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*, pages 1728–1738, 2021.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[6] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. Locvtp: Video-text pre-training for temporal localization. In *ECCV*, pages 38–56, 2022.

[7] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *EMNLP*, pages 162–171, 2018.

[8] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*, 2019.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019.

[10] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, pages 5267–5275, 2017.

[11] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander G. Hauptmann. Excl: Extractive clip localization using natural language descriptions. In *NAACL-HLT*, pages 1984–1990, 2019.

[12] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[13] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, pages 18995–19012, 2022.

[14] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, pages 505–520, 2014.

[15] Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. In *CVPR*, pages 1001–1009, 2016.

[16] Fa-Ting Hong, Xuanteng Huang, Wei-Hong Li, and Wei-Shi Zheng. Mini-net: Multiple instance ranking network for video highlight detection. In *ECCV*, pages 345–360, 2020.

[17] Hao Jiang and Yadong Mu. Joint video summarization and moment localization by cross-task sample transfer. In *CVPR*, pages 16388–16398, 2022.

[18] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *NeurIPS*, pages 11846–11858, 2021.

[19] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, pages 7331–7341, 2021.

[20] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, pages 447–463, 2020.

[21] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[22] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021.

[23] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022.

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

[25] Wu Liu, Tao Mei, Yongdong Zhang, Cherry Che, and Jiebo Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *CVPR*, pages 3707–3715, 2015.

[26] Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *CVPR*, pages 3042–3051, 2022.

[27] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *CVPR*, pages 202–211, 2017.

[28] 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*, pages 2630–2640, 2019.

[29] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033, 2023.

[30] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *CVPR*, pages 10810–10819, 2020.

[31] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *ECCV*, pages 407–426, 2022.

[32] Saiteja Nalla, Mohit Agrawal, Vishal Kaushal, Ganesh Ramakrishnan, and Rishabh Iyer. Watch hours in minutes: Summarizing videos with user intent. In *ECCV*, 2020.

[33] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *ECCV*, pages 540–555, 2014.

[34] Shraman Pramanick, Li Jing, Sayan Nag, Jiachen Zhu, Hardik Shah, Yann LeCun, and Rama Chellappa. Volta: Vision-language transformer with weakly-supervised local-feature alignment. *arXiv preprint arXiv:2210.04135*, 2022.

[35] Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. *arXiv preprint arXiv:2307.05463*, 2023.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[37] Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Naq: Leveraging narrations as queries to supervise episodic memory. In *CVPR*, pages 6694–6703, 2023.

[38] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Trans. Assoc. Comput. Linguistics*, 1:25–36, 2013.

[39] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019.

[40] Yong Rui, Anoop Gupta, and Alex Acero. Automatically extracting highlights for tv baseball programs. In *MM*, pages 105–115, 2000.

[41] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pages 8430–8439, 2019.

[42] Aidean Sharghi, Jacob S Laurel, and Boqing Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *CVPR*, 2017.

[43] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018.

[44] Yale Song, Miriam Redi, Jordi Vallmitjana, and Alejandro Jaimes. To click or not to click: Automatic selection of beautiful thumbnails from videos. In *CIKM*, 2016.

[45] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, pages 5179–5187, 2015.

[46] Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, pages 787–802, 2014.

[47] Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Kevin Qinghong Lin, Satoshi Tsutsui, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, et al. All in one: Exploring unified video-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6598–6608, 2023.

[48] Lezi Wang, Dong Liu, Rohit Puri, and Dimitris N Metaxas. Learning trailer moments in full-length

movies with co-contrastive attention. In *ECCV*, pages 300–316, 2020.

[49] Guande Wu, Jianzhe Lin, and Claudio T Silva. Intentvizor: Towards generic query guided interactive video summarization. In *CVPR*, pages 10503–10512, 2022.

[50] Shuwen Xiao, Zhou Zhao, Zijian Zhang, Ziyu Guan, and Deng Cai. Query-biased self-attentive network for query-focused video summarization. *IEEE Trans. Image Process.*, 29:5889–5899, 2020.

[51] Shuwen Xiao, Zhou Zhao, Zijian Zhang, Xiaohui Yan, and Min Yang. Convolutional hierarchical attention network for query-focused video summarization. In *AAAI*, volume 34, pages 12426–12433, 2020.

[52] Bo Xiong, Yannis Kalantidis, Deepti Ghadiyaram, and Kristen Grauman. Less is more: Learning highlight detection from video duration. In *CVPR*, pages 1258–1267, 2019.

[53] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *MM*, pages 1645–1653, 2017.

[54] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016.

[55] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcia, Brais Martinez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. In *ICCV*, pages 7220–7230, 2021.

[56] Minghao Xu, Hang Wang, Bingbing Ni, Riheng Zhu, Zhenbang Sun, and Changhu Wang. Cross-category video highlight detection via set-based learning. In *ICCV*, pages 7970–7979, 2021.

[57] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning, 2023.

[58] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo. Unsupervised extraction of video highlights via robust recurrent autoencoders. In *ICCV*, pages 4633–4641, 2015.

[59] Qinghao Ye, Xiyue Shen, Yuan Gao, Zirui Wang, Qi Bi, Ping Li, and Guang Yang. Temporal cue guided video highlight detection with low-rank audio-visual fusion. In *ICCV*, pages 7950–7959, 2021.

[60] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguistics*, 2:67–78, 2014.

[61] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *CVPR*, pages 10287–10296, 2020.

[62] Can Zhang, Tianyu Yang, Junwu Weng, Meng Cao, Jue Wang, and Yuexian Zou. Unsupervised pretraining for temporal action localization tasks. In *CVPR*, pages 14031–14041, 2022.

[63] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *ACL*, pages 6543–6554, 2020.

[64] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *NeurIPS*, 2022.

[65] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, pages 766–782, 2016.

[66] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, volume 34, pages 12870–12877, 2020.

[67] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023.