Moment Detection in Long Tutorial Videos

Ioana Croitoru* Simion-Vlad Bogolin[3] Samuel Albanie[2] Yang Liu[4]
Zhaowen Wang[1] Seunghyun Yoon[1] Franck Dernoncourt[1] Hailin Jin[1] Trung Bui[1]
[1]Adobe Research [2] Department of Engineering, University of Cambridge
[3] Filtir [4] Wangxuan Inst. of Computer Technology, Peking University

Abstract

Tutorial videos play an increasingly important role in professional development and self-directed education. For users to realise the full benefits of this medium, tutorial videos must be efficiently searchable. In this work, we focus on the task of moment detection, in which the goal is to localise the temporal window where a given event occurs within a given tutorial video. Prior work on moment detection has focused primarily on short videos (typically on videos shorter than three minutes). However, many tutorial videos are substantially longer (stretching to hours in duration), presenting significant challenges for existing moment detection approaches.

To study this problem, we propose the first dataset of untrimmed, long-form tutorial videos for the task of Moment Detection called the Behance Moment Detection (BMD) dataset. BMD videos have an average duration of over one hour and are characterised by slowly evolving visual content and wide-ranging dialogue. To meet the unique challenges of this dataset, we propose a new framework, LONGMOMENT-DETR*, and demonstrate that it outperforms strong baselines. Additionally, we introduce a variation of the dataset that contains YouTube Chapter annotations and show that the features obtained by our framework can be successfully used to boost the performance on the task of chapter detection. Code and data can be found at* https://github.com/ioanacroi/longmoment-detr.

1. Introduction

Enabled by cheaper disk storage and networking technology, long-form videos of tutorial content are proliferating. As such, there is a pressing need to develop effective tools for searching *within* videos. In this work, we consider this problem through the lens of *moment detection*— given a video and a natural language query, our task is to find the temporal span of the video that best matches the query (as shown in Fig. 1). Beyond tutorial content, this task also has applications in domains such as security and entertainment.

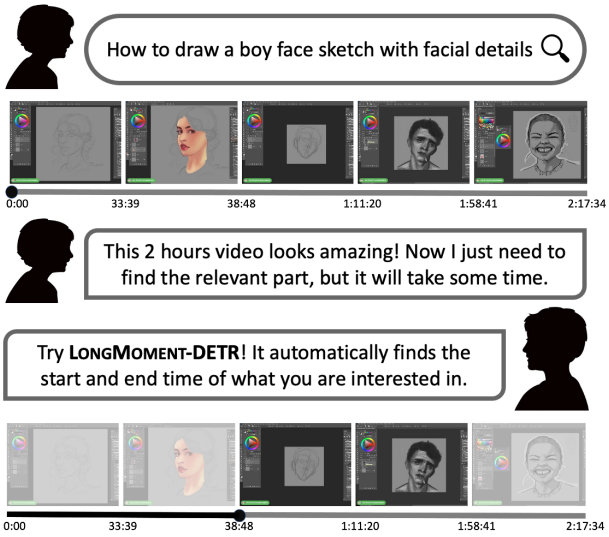


Figure 1. **LONGMOMENT-DETR.** Our framework performs moment detection and unlike previous state of the art methods, it works on long tutorial videos.

To study moment detection in the long-form setting, we introduce, the first database of long tutorial videos with manual annotations for validation and testing, called Behance Moment Detection (BMD). The videos, which are collected from the Behance platform[1], consist of tutorial videos that teach skills with various creative tools such as drawing, movie editing, animation and photo editing.

Existing moment detection datasets predominantly feature short videos centered on human activities, such as cooking or swimming. In contrast, our BMD dataset emphasizes long tutorial videos that explore the use of software tools for digital artistry. Typically, these tutorials involve screen-sharing sessions detailing creative processes, often spanning several hours. Efficiently localizing specific segments in such long videos can greatly enhance user nav-

*Work done during an internship at Adobe. Now at V7 Labs.

[1] https://behance.net

igation experience. This makes the BMD dataset unique for the task of moment detection. Moreover, to scale up training beyond costly manual annotation, we propose a framework for moment detection that leverages weak supervision derived from ASR (Automatic Speech Recognition) and video segmentation. In this new long video setup, using the timings from ASR, similar to previous works [24, 17], produces poor results, as we will show experimentally. So, in order to adjust for long videos setup, we rely on a mix of using video segmentation methods [36, 37] and summarization [50, 27, 5, 3] in order to generate good timings and textual descriptions to use for training.

Lastly, in order to further validate our approach, we propose a second dataset, called YouTube Chapters (YTC) that contains full chapter annotation for all splits. In this way, we are able to assess the quality of the learnt features on BMD on the downstream task of chapter detection.

We summarise our contributions as: (1) we introduce the first two long tutorial video datasets: Behance Moment Detection (BMD) a multi-modal dataset, suitable for weak supervision with manual annotations for validation and testing and YouTube Chapters (YTC) with chapters annotations for all splits (2) we propose an effective way to automatically generate moments for the training split of BMD that leverages Automatic Speech Recognition (ASR) and employs large language models (LLMs) to eliminate the inherent noise that appears in ASR. (3) we show the effectiveness of using BMD to improve the performance on the downstream task of YouTube chapter detection.

Leveraging the multi-modal features of our new dataset and utilizing automatic annotations, we present LONGMOMENT-DETR, the first framework for moment detection in long tutorial videos consisting of the segment timing generator and query generator.

2. Related work

Moment detection from video. Many works have been proposed for the task of moment localization [17, 24, 14, 12, 19]. There are two popular families of approaches for moment detection on short videos: two-stage approaches and one-stage approaches. In the two-stage approaches [28, 39, 51, 14, 9, 49, 47, 46], firstly a list of potential candidates is generated and then the candidates are ranked according to various scores. In the one-stage approaches [17, 24, 18, 19, 48], the methods directly process the entire video and output the moment localization. The most recent methods are based on the transformer encoder-decoder architecture. Additionally, the methods [24, 17] are designed to jointly predict moment detection and highlight detection, while [28] tackles zero-shot moment detection. [28] is tailored for short videos and it consists of various modules (temporal event proposal, object detector, pseudo-query generation module). For our use case, running an object detector is highly expensive, since the videos are very long. Moreover, the produced results are usually of poor quality for tutorial videos since general object detectors are not tailored for this type of content where the host shares their screen and presents how to use specific software applications. Very recently, [29, 30] propose architectural changes to accommodate for long-formed videos, specifically within the movie domain. However, given the scarcity of datasets dedicated to long videos, particularly those with lengthy segments, this area remains underexplored. To address this issue, we are the first to propose the task of moment detection in long tutorial videos, along with two datasets containing very long untrimmed tutorial videos: one having manual segments annotations for evaluation (Behance Moment Detection) and the other having YouTube Chapters annotations for all splits.

Large Language Models (LLMs). These models are trained on vastly large corpora of text consisting of tens of TB of data and have shown surprising zero-shot capabilities on a variety of NLP tasks [50, 3, 27, 40, 34, 6]. Notable among these are OPT175B [50], Bloom175B [38], and GPT3 [3], each boasting roughly 175B parameters. While GPT3 is available only through a paid API, OPT and Bloom models are open-source and are freely accessible to the research community. Even without task-specific finetuning, they exhibit strong zero-shot performance on multiple NLP benchmarks [20, 35, 10]. A distinguishing trait of LLMs is their prompt-driven adaptability to varied tasks. In our case, for tutorial videos, the available transcripts, although informative, can be noisy. These transcripts, generated using efficient automatic speech recognition tools [32, 2], are refined by LLMs, which we employ to extract relevant information and eliminate noise.

Video segmentation. The task of temporal video segmentation aims to split the whole video into smaller sub-videos based on the semantic content. This is an important task in computer vision, being an essential pre-processing step for various other video understanding downstream tasks, such as video retrieval, moment detection or video summarization. In our case, we aim to use such models for segment timing generation. Various methods have been proposed for this task [44, 37, 36, 45], however they are designed and trained on short videos databases [42, 13]. Because of this, we have chosen to use unsupervised temporal video segmentation methods [36, 37, 15], that do not require retraining for our particular case.

3. Dataset

Both datasets, Behance Moment Detection (BMD) and YouTube Chapters (YTC), consist of long tutorial videos and are collected from the Behance platform. These are the first two datasets containing untrimmed, long-form tutorial videos for the tasks of moment detection and chapters detec-

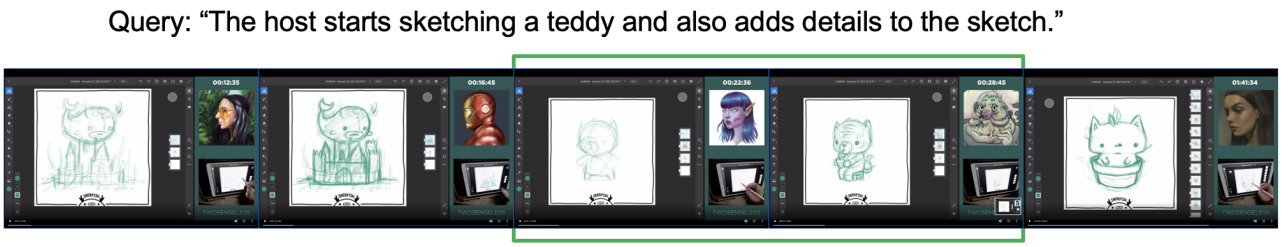


Figure 2. **Behance Moment Detection example.** We propose a moment retrieval dataset containing long tutorial videos. Each video is split into multiple segments and each segment has an associated query. We highlight with green the moment associated to the given query.

tion, respectively. While these tasks are important in videos of any duration, moment localization in long-form tutorial videos can add extra value since it improves the user experience and eases video navigation by saving significant searching time (for example a user can only be interested where a specific character is drawn, but it is time-consuming to find that part in a long video).

The validation and testing splits of BMD are manually annotated, while the training split is automatically curated. For YTC, we sourced human-annotated chapters from YouTube. All data is publicly accessible online.

3.1. Behance Moment Detection

We collected videos from Behance Livestream, a platform for creative users who share their work, having over 30 million users. The videos from our dataset were livestreamed and then made publicly available on the platform. We choose Behance since the videos are of high quality with relatively little noise and the type of the videos are aligned with our target videos, namely long tutorial videos. The usual flow for these videos is: 1) the host greets the viewers and talks about what they will showcase in the video (usually they show how to use different creative apps such as Adobe Fresco or Photoshop), 2) the host shares their screen and 3) they start using the chosen app while talking and giving useful tips and details about the creative process. For example, the host can use drawing tools for creating cartoon characters or can showcase how to use video editing tools for creating a movie, how to add shadows, etc. One such example can be seen in Fig. 2. These tutorial livestreams are usually very long, the majority being over one hour long, as presented in Fig. 3. The videos represent source materials that teach skills for educational purposes. There are thousands of long tutorial videos available on Behance and there are thousands more on other video platforms such as YouTube, which can directly benefit from our method of moment detection in long videos.

In Tab. 1 we present a comparison with other moment detection datasets. There is a striking difference when comparing the average length of the videos and of the segments with other existing datasets. There is only one other dataset which contains long videos, namely the MAD [41] dataset.

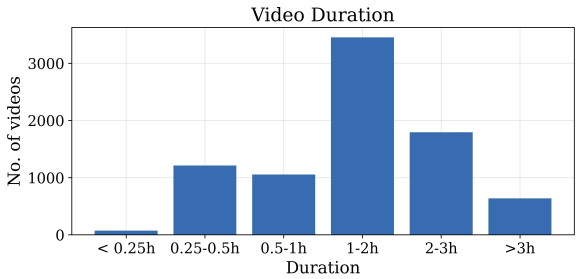


Figure 3. **Histogram of video duration.** The majority of the videos in the BMD dataset are longer than one hour.

MAD contains videos extracted from movies and leverages the audio descriptions of movies for visually impaired audiences, having the average length of the segment of around 4.1s, while ours has a much longer segment duration of over 17 minutes. This makes our dataset more suitable for downstream tasks, such as chapter detection. Another major difference is that the videos from our datasets are untrimmed, while for some other datasets such as QVHighlights [17], the videos are automatically split into 3-minute segments. This can result in loss of context or some actions being trimmed in an unnatural way, which can lead to annotations issues. The BMD dataset contains over $13,000$ hours of videos (with over 630 hours of human-annotated videos). There are 7.9k videos used for training, 117 for validation and 171 for testing. While other moment retrieval datasets suffer from temporal biases [14, 9, 19] where the segments tend to occur in the beginning of the video, our dataset contains segments that span the entire video. All the videos in the dataset have transcripts, extracted with Azure speech-to-text. On average each transcript contains 1500 sentences.

Validation and testing split. For the validation and testing splits we outsourced the annotation task to a specialised company having a team of qualified and experienced annotators. The annotators were asked to watch the tutorial video, segment it into high-level chapters and write a natural language description for each segment. The description is a brief summary of what happens in the segment and should contain no more than a few sentences. The cost for annotating the validation and testing split was approximately $\$10,000$ and took around 3 months for the 288 videos.

Training split. Due to the high cost of collecting the an-

Dataset	Domain	#videos	#queries	Avg len videos(sec)	Avg len segment	Total number of hours
DiDeMo [14]	Flickr	10.6k	41.2K	29.3	6.5	88
ActivityNet [16]	Activity	15K	72K	117.6	36.2	490
CharadesSTA [12]	Activity	6.7k	16.1K	30.6	8.1	57
TVR [19]	TV show	21.8K	109K	76.2	9.1	461
QVHighlights [17]	Vlog/News	10.2K	10.3K	150	24.6	425
MAD [41]	Movies	650	384.6K	6646(1.8 hours)	4.1	1207.3
BMD-Train†	Livestream	7.9K	35.9K	5768 (1.6 hours)	1022 (17 mins)	12.7k
BMD-Eval	Livestream	288	1499	7957 (2.2 hours)	1537 (25.6 mins)	638
YouTube Chapters	Livestream	467	4391	6472 (1.8 hours)	687 (11.5 mins)	840

Table 1. **Comparison between BMD and YTC to existing moment retrieval datasets.** The segments from our datasets are significantly longer than the ones from any other existing datasets. Moreover, in our datasets the segments span the entire duration of the video eliminating potential biases. †BMD-Train contains automatically curated annotation.

notations, the training set was annotated automatically by leveraging the transcript together with pretrained large language models. Since the videos are livestreamed, the host interacts with the viewers and verbally explains the steps involved in the creative process for making digital art, including a detailed description of what they create or how to efficiently use the tools to obtain the desired outcome. While parts of the transcripts are extremely informative regarding what is happening in the video, there is also a lot of noise, which includes redundant information or information that is irrelevant to the tutorial (for example chit-chat about events that happened lately). Because the average length of the transcript is around 1500 sentences, we need an efficient way of condensing and extracting the relevant information. We achieve this by using GPT3 through the API provided by OpenAI. This incurs an additional cost of approximately $\$1,000$ for all training videos. More details about the summarization procedure and a detailed comparison between different LLMs can be found in Sec. 4.3.

While this gives us the ability to generate high-quality queries for the segments, another challenge is how to generate timings and split the video into meaningful parts. This step is equally important and affects the final performance of the model as we will show later. In order to address this challenge, we considered various state-of-the-art methods for video segmentation [36, 37] in order to generate segment timings. As the final model, we use the OSG [36] method. More details can be found in Sec. 4.3.

3.2. YouTube Chapters

Recently, YouTube introduced the functionality to add video chapters as a way to improve the user experience, especially for long videos. The creator of the video has the possibility to segment the video into several sections and to add a short description to each section. This facilitates easy rewatching or skipping to the desired content in the video. Some of the videos from our Behance Moment Detection dataset are also published on the YouTube platform and contain manual chapter labels. In order to further test the quality of our method, we collected chapter annotations from almost 500 videos (379 in training, 17 in validation and 71 in the testing split). Most of the visual content of the videos from YouTube Chapters already exists in the Behance Moment Detection dataset, which assures the fact that the videos are from the domain we are interested in (namely tutorial videos). However, there are clear differences between the two annotations. Firstly, the queries from YouTube Chapters are very brief and consist of just a few words (such as *How to remove chromatic aberration* or *Creating GIFs in Photoshop*) while the annotations from BMD dataset are fully formed sentences describing the actions happening in that segment of the video, as presented in Fig. 4. Secondly, the temporal segmentation of the video is different (the average segment length in BMD is approx. 20 minutes long, while in YTC is approx. 11 minutes long, as illustrated in Tab. 1), thus being a domain gap between the BMD and YTC annotations.

4. Method

We first define the task of moment retrieval in long tutorial videos (Sec. 4.1). Then, we introduce our full framework (Sec. 4.2), providing additional insights about our automatic moment generation pipeline (Sec. 4.3). Finally, we present architecture details (Sec. 4.4) and elaborate about our design choices.

4.1. Task definition

Given a video $v = \{f_1, f_2, ..., f_V\}$ (where f_i represents the i-th frame of the video) and a query q in natural language, the objective of moment detection is to find the pair $(i, j), i < j$ so that the video segment starting at f_i and finishing at f_j is best described by the query q. Our goal is to train a model that receives as input the whole video v and query q and outputs the start and end time (i, j).

Let $D = \{v_l | l = 1..N\}$ be a dataset of N videos with

segment annotations $A = \bigcup_{l=1}^{N} A_l$ where

$$A_l = \{(q_k, (i_k, j_k)) | f_{i_k}, f_{j_k} \in v_l, i_k < j_k, k = 1..Q_l\}$$

denotes the segment annotations for video v_l. Q_l represents the number of segments in v_l while q_k represents the query written in natural language for the segment starting at i_k and finishing at j_k. In order to train the model M for the task of moment detection we use a similar setup as Moment-DETR [17], which is described in more detail in Sec. 4.4.

4.2. The LONGMOMENT-DETR framework

In Algorithm 1 we present the process for our LONGMOMENT-DETR framework. Since the cost (both time and financial resources) of collecting manual annotations for thousands of videos is high, an effective way of decreasing this cost is to curate automatic annotations for the training split D. After gathering the videos from the Behance platform, the first step is to automatically generate the transcripts from the audio modality using the Azure speech recognition tool. Next, the videos are split into several segments representing different key parts of the video. After having the timings for the segments, we use the corresponding transcript for that timespan and we summarize it in order to extract the essential information. We then proceed to train our final model M using a standard approach for this task, similarly to [17].

Algorithm 1 LONGMOMENT-DETR framework

Phase 1: $generateMomentAnnotations(D) \rightarrow A$
  $A = \emptyset$
  for video v_l in D **do**
    Generate transcript s_l for v_l
    $T_l = segmentsGeneration(v_l)$
    $A_l = \emptyset$
    for timing $(i_k, j_k) \in T_l$ **do**
      $q_k = queryGeneration((i_k, j_k), s_l)$
      Add $(q_k, (i_k, j_k))$ to A_l
    end for
    $A = A \cup A_l$
  end for
Phase 2: *Train the final model*, M
  for minibatch of paired samples (v, a), $a = (q_k, (i_k, j_k)) \in A, v \in D$ **do**
    Extract video features x from v
    Extract text features p_k from q_k
    Feed x and q_k to M and get output o.
    Compute the $loss$ between o and (i_k, j_k).
    Update M base on $loss$
  end for

4.3. Moment generation

Our approach has two steps: segments timing generation and query generation which will be discussed next.

Segment timing generation. Temporally segmenting the video according to topics is an important step in our pipeline which affects the final performance of the system. We considered and compared various potential solutions: starting directly from using the timings given by the speech recognition tool or splitting the video into random segments to using state of the art unsupervised methods such as OSG [36] and ONG [37]. The last two video segmentation methods receive as input the video features (in our case, SlowFast [11] features) and a parameter s, representing the number of segments to split the video in. For choosing the parameter s, we tested two approaches: estimate the number of segments based on SVD (as described in [36]) or using a constant. Based on the results of these experiments (for full results, see Sec. 5.2) for the final method we use OSG [36] with five scenes per video. This step acts as the $segmentsGeneration$ function in Algorithm 1.

Query generation. For this task, the simplest approach is to use directly the transcript. However, while parts of the transcript are quite informative, it also contains noise and information that is not relevant to the visual content or to the tutorial (for example the host may converse about the weather). It would therefore be useful to have an automatic method to extract a condensed version of the relevant information from the transcript associated with a video segment. For this, we considered various techniques such as extractive summarization [5] or using large language models (LLMs). Models such as Bloom175B [27], OPT175B [50] and GPT3 [3] are able to perform a large variety of tasks, including summarization and information extraction from large chunks of text. We begin by concatenating all the sentences from the transcript that span during the start and end time of a segment proposal and feed it to the language model along with the instructions to summarize the text. We used the following prompt: *"Summarize this tutorial transcript:"*. In Sec. 5.2 we present detailed ablations studies between different summarization methods. For the final model, we used GPT3 (Curie model) to obtain the query for each generated segment. In cases where the segment transcript is too long to be processed by GPT3, the final query is the concatenation of several splits of the transcript, each processed by GPT3 independently. This step acts as the $queryGeneration$ function in Algorithm 1.

4.4. Architecture details

We start from the Moment-DETR [17] architecture which uses an encoder-decoder transformer [43] and is based on the DETR architecture [4]. The model receives as input the concatenation between the video and text features and outputs the center and width of the predicted moment.

Method	$\#scenes$	Precision	Recall	F1
Transcript timing	200	97.8	8.7	15.0
Random split	5	54.5	71.6	53.5
ONG [37]	SVD	48.19	86.70	43.42
OSG [36]	SVD	67.08	76.45	57.18
ONG [37]	5	45.95	89.21	42.69
OSG [36]	5	62.17	83.37	59.51

Table 2. **Comparison of different video segmentation methods.** Using directly the timings from the transcript has very poor results, while OSG performs the best in terms of F1 score.

Segmentation Method	$R1@0.5 \uparrow$	$R1@0.7 \uparrow$
Transcript timing	$3.1_{\pm 0.4}$	$0.5_{\pm 0.2}$
Random	$9.4_{\pm 1.6}$	$2.6_{\pm 0.8}$
ONG	$7.1_{\pm 0.6}$	$1.7_{\pm 1.4}$
OSG	$13.4_{\pm 0.5}$	$6.3_{\pm 0.3}$

Table 3. **Comparison between different video segmentation methods combined with LONGMOMENT-DETR.** The timings generation has a direct impact on performance. The number of scenes considered as input parameter for OSG and ONG is 5.

One limitation of [17] is that it assumes the video has a length of 3 minutes. In order to make it suitable for running on longer videos, we adjust the meta-parameters and make low-level changes to the model implementation such as removing hard-coded constraints. Please see Suppl.Mat. for more details. This modified version becomes the architecture used in our whole LONGMOMENT-DETR framework. For features, we employ SlowFast [11] video understanding features extracted every 8 seconds and GPT2-xl [33] features for the text.

Implementation details. Our model is trained in PyTorch, using the AdamW optimizer [25] with a learning rate of 1e-4 and weight decay of 1e-4. All the models were trained on an NVIDIA A100 GPU. Due to memory limitations, we trained the model with a batch size of 16.

5. Experiments

5.1. Metrics

We report the standard metrics used in moment detection: $R1@0.5$ and $R1@0.7$, where a prediction is considered correct if it has an intersection over union (IoU) with the ground truth ≥ 0.5, respectively ≥ 0.7. When it comes to assessing the performance of video segmentation methods we report the standard metrics for this task: precision, recall and F1-score, while for comparing query generations we report the ROUGE metrics [22] which are commonly used for the task of summarization [26, 8, 1] and are proven to have a good correlation with the human judgment [23].

5.2. Ablations

For all the ablation experiments, we report results on the validation split, unless otherwise stated.

Segment timing generation. We begin by testing several methods and assess their video segmentation performance in Tab. 2. This performance correlates with the moment detection results shown in Tab. 3. Moreover, we observe that following prior works [17, 24] and using only the timings from the transcripts gives poor results. This is expected, since for BMD the average segment length is considerably longer than the segment obtained from transcript timing. In fact, just by randomly splitting the video into 5 segments, the performance improves significantly. We obtain the best results using the OSG [36] video segmentation method using five scenes.

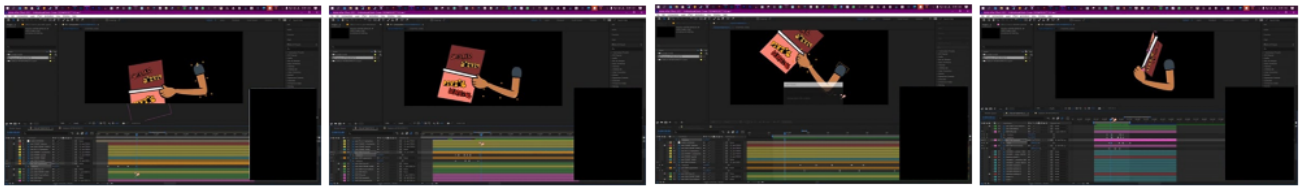
Method	R-1	R-2	R-L
Transcript	2.31	0.34	1.85
Captioning BLIP [21]	7.56	0.5	6.8
Extractive [5]	8.54	0.59	6.17
Bloom175B [38]	9.09	0.52	7.36
OPT175B [50]	10.99	0.74	8.23
GPT3 [3]	15.49	1.36	11.85

Table 4. **Evaluation of query generation using ROUGE metrics.** We assess the performance of different query generation methods. The transcript, being significantly longer than the human annotation, has a low score, while the LLMs perform the best.

Query generation. Another important step in our method is automatic query generation. Since the image and video captioning systems are trained on visual data that is out of domain as compared to our tutorial videos, we expect those systems to perform worse than the systems based on textual information. In Tab. 4 we present summarization results as compared to the human annotation for the validation split and observe that GPT3 obtains the best results. Also, in Fig. 4 we present a qualitative example for the query generation task using LLMs.

In Tab. 5 we present results for LONGMOMENT-DETR using various query generation techniques. The best score is obtained using the GPT3 model which correlates with the results from Tab. 4. However, while there is a correlation between the results, we notice that there is a significantly smaller gap between the performance of using LLMs such as Bloom175B or OPT175B and using the raw transcript for the task of moment detection. An explanation for this, is that even though the transcript is very long, which directly affects the ROUGE score, for the task of moment detection it contains enough relevant information in order to make up for the differences in performance. For the final model, we use the GPT3 model as the query generation method.

Influence of pre-trained features. Our model receives at input pre-trained features for both the visual and textual component. For this ablation study, we assess the influence of such features on the final performance as seen in Tab. 6. For the visual side, we consider the SlowFast [11] and CLIP [31] features while for the textual side, we consider GPT2-xl [33], DiffCSE [7] and CLIP [31]. The best



Human: "The host starts working on the video on After effect. The host sync the video with music and sync the movement of the arm to the music. The host adjust the animation movement of the flag according to the action of the arm."
GPT3: "In this tutorial, the speaker demonstrates how to animate a character's hand using the software After Effects. This tutorial is about creating an animation in which a character's arms move up and down in time with a beat."
OPT: "I wanted to take you through the process Learn how to improve the animation and timing in your 3D walk cycle by using the graph editor. First learn the basics of changing a keyframe. Finally, learn the In this tutorial, you learned how to add a simple effect to your hand movements."
Bloom: "How to make a simple animation in After Effects. We learned how to create a simple animation using the Pen Tool, how to add a simple text animation. how to make a cartoon character walk in After Effects. We learned how to create a simple walk cycle, how to animate the walk cycle."

Figure 4. **Comparison of summaries generated by GPT3, OPT175B and Bloom175B.** During the summarization process no human annotation was used. In this figure, the human description and video frames are shown only for reference. While the AI generated summaries are obtained only from the transcript, the human had access to the whole video and audio when making the annotation.

Summ Method	$R1@0.5 \uparrow$	$R1@0.7 \uparrow$
Transcript	$13.6_{\pm 0.5}$	$6.3_{\pm 0.3}$
Bloom175B	$13.4_{\pm 0.6}$	$7.0_{\pm 0.2}$
OPT175B	$14.0_{\pm 0.6}$	$6.7_{\pm 0.8}$
GPT3	$16.8_{\pm 0.5}$	$9.2_{\pm 1.0}$

Table 5. **Comparison between different LLM query generation methods in conjunction with LONGMOMENT-DETR.** As it can be seen, GPT3 performs the best. What is interesting is that by using the raw transcript, the performance is only slightly worse than using other power full summarization methods such as Bloom175B and OPT175B.

Video feats	Text feats	$R1@0.5 \uparrow$	$R1@0.7 \uparrow$
SlowFast	GPT2-xl	$16.8_{\pm 0.5}$	$9.2_{\pm 1.0}$
SlowFast	DiffCSE	$13.1_{\pm 1.5}$	$6.5_{\pm 1.2}$
SlowFast	CLIP	$11.2_{\pm 0.7}$	$4.9_{\pm 0.9}$
CLIP	CLIP	$12.7_{\pm 0.6}$	$6.2_{\pm 0.3}$
CLIP	GPT2-xl	$13.8_{\pm 1.2}$	$7.1_{\pm 1.8}$
CLIP	DiffCSE	$13.8_{\pm 1.9}$	$7.5_{\pm 0.7}$

Table 6. **Influence of pre-trained features on performance.** The best performance is obtained using SlowFast features for the visual side and GPT2-xl features for the textual side.

results are obtained using SlowFast for the visual side and GPT2-xl for the textual side.

Influence of different components. In Tab. 7 we present an overview of the influence of different components on the testing split (the results on validation split are presented previously in each corresponding section). We start with a Moment-DETR [17] model as Baseline. We observe that both the segment timing generation and the query generation have a strong impact on performance. We obtain the best results by combining the timing generation from OSG with the GPT3 query generation which represents our final LONGMOMENT-DETR framework.

Component	Segments	Queries	$R1@0.5 \uparrow$	$R1@0.7 \uparrow$
Baseline [17]	No	No	$2.7_{\pm 0.6}$	$0.3_{\pm 0.1}$
Random seg	Random	No	$9.2_{\pm 1.8}$	$2.6_{\pm 0.4}$
Summarize	Random	GPT3	$13.2_{\pm 0.3}$	$5.6_{\pm 0.6}$
ShotDetect	OSG	No	$12.6_{\pm 0.4}$	$5.5_{\pm 0.4}$
LONGMOMENT-DETR	OSG	GPT3	$16.3_{\pm 0.3}$	$8.3_{\pm 0.5}$

Table 7. **Effect of different components on performance.** Both the segment timing generation and query generation have a strong impact on performance. Hence, in the final model, we use OSG and GPT3, thus obtaining our final model LONGMOMENT-DETR. The results are presented on the testing split.

5.3. Comparison with others

In this section, we compare our method against two recent methods: Moment-DETR [17] and UMT [24] adjusted for ingesting long videos. For UMT we use the *video only* version as introduced in the paper [24]. This comparison can be seen in Tab. 8. By training using the raw transcripts, which is similar to the pre-training stage of [17], the results in our case for long tutorial videos are poor. However, by adjusting the base architecture and use the automatic generated segments tailored for long videos, the results improve significantly for Moment-DETR as observed in the table. Moreover, if we use our full LONGMOMENT-DETR framework, as described in Sec. 4.2, the results are significantly better than previously published methods.

5.4. YouTube Chapters

To further evaluate the quality of our features obtained using the automatic annotations, we have collected new data from YouTube Chapters as described in Sec. 3.2.

Experimental setup. We initially train the network using BMD. Once trained, these weights serve as the founda-

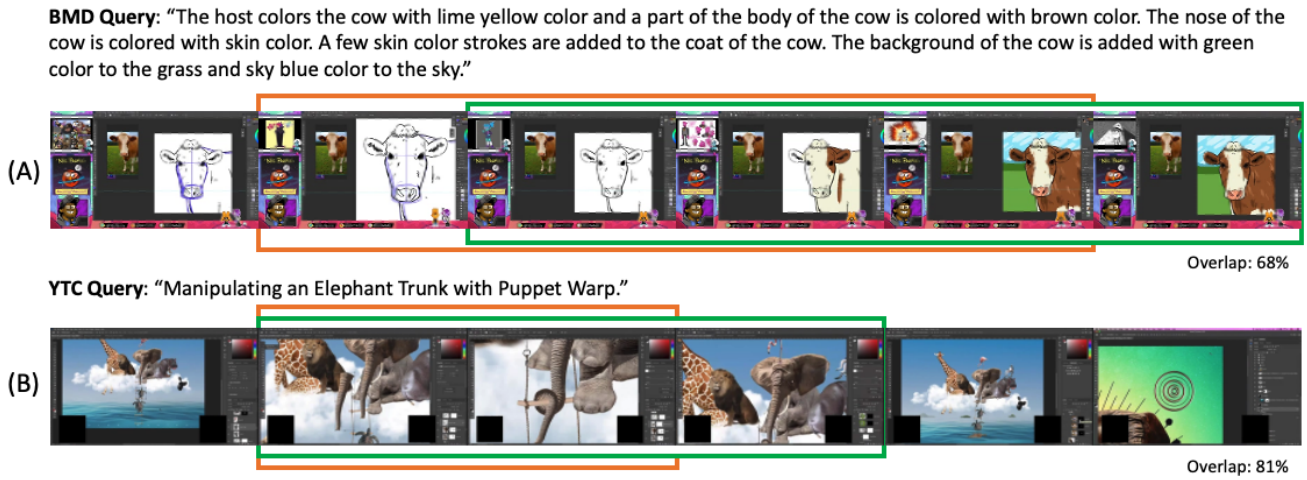


Figure 5. **Qualitative examples for BMD (A) and YTC (B).** Along with the query, we show several video frames, the prediction results in orange and the ground truth in green. We also specify the overlap between the prediction and ground truth segments. Please note how different the queries between BMD and YTC are. Also, while the prediction results have an offset from the ground truth, we want to highlight how subtle is the change in scenery for our proposed datasets.

Model	$R1@0.5 \uparrow$	$R1@0.7 \uparrow$
Moment-DETR [17]	$2.7_{\pm 0.6}$	$0.3_{\pm 0.1}$
UMT [24]†	$11.5_{\pm 1.0}$	$4.6_{\pm 1.3}$
Moment-DETR [17]†	$12.6_{\pm 0.4}$	$5.5_{\pm 0.4}$
LONGMOMENT-DETR	$\mathbf{16.3}_{\pm 0.3}$	$\mathbf{8.3}_{\pm 0.5}$

Table 8. **Comparison with other methods.** As it can be seen, LONGMOMENT-DETR achieves better results than other moment detection methods. †denotes training the models with the timings obtained by our proposed method using OSG. The results are presented on the testing split.

Model	Pre-training	Training	$R1@0.5 \uparrow$	$R1@0.7 \uparrow$
LONGMOMENT-DETR	BMD	-	$5.0_{\pm 2.4}$	$2.2_{\pm 1.5}$
CHAPTER-DETR	-	YTC	$12.6_{\pm 0.3}$	$5.8_{\pm 0.6}$
CHAPTER-DETR	BMD	YTC	$\mathbf{16.1}_{\pm 0.5}$	$\mathbf{6.6}_{\pm 0.3}$

Table 9. **Results on YouTube-Chapters.** If we evaluate LONGMOMENT-DETR trained with BMD directly on the task of chapter detection on the YTC dataset, the performance is low. However, if we use BMD as pre-training data and then train fully supervised for the task of chapter detection, there is a significant boost in performance as opposed to randomly initializing the model (4th line vs 3rd line).

tion for fine-tuning on the subsequent task of chapter detection. While the BMD training data is automatically curated, when we transition to training on the manually annotated data from the YTC dataset, we distinguish this model by referring to it as CHAPTER-DETR. This model uses the same architecture as LONGMOMENT-DETR.

Results. The results for chapter detection are presented in Tab. 9. We observe that when only pre-training on BMD and directly assessing performance, the performance is low, the main reason being the distribution shift between the queries and segments in BMD and the ones in YTC. The queries from BMD are often composed of several sentences while the chapter annotation only contains a few words (as showcased in Fig. 5). However, if we first pre-train on BMD and then finetune with chapter annotations, there is a major boost in performance. These results prove that not only LONGMOMENT-DETR can be used on its own for moment detection, but it can be used to significantly improve the performance for the task of chapter detection.

Limitations. Since our method relies on automatic annotation generation, various problems such as text hallucination or incorrect segment timing generation directly affect the results of our system. Other limitations regarding particular components are discussed throughout the paper.

Societal impact. Moment retrieval systems enable efficient content discovery for learning. However, these systems can exhibit biases towards particular groups or they can be used for spreading misinformation. Another aspect is the notable cost and carbon footprint required to train modern models.

6. Conclusions

In this paper we presented the LONGMOMENT-DETR framework which tackles the task of moment detection in long tutorial videos. We introduced, the first two long tutorial video datasets, Behance Moment Detection (BMD) and YouTube Chapters (YTC). BMD has human annotations for testing and validation splits and we propose a novel system to automatically curate the annotations for the training split by leveraging the information from the transcript. Lastly, we introduced YTC, a dataset containing human annotated chapters, and we further test and prove the effectiveness of the features obtained from BMD for chapter detection.

References

[1] Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293, 2021. 6

[2] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. Unsupervised speech recognition. *Advances in Neural Information Processing Systems*, 34:27826–27839, 2021. 2

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2, 5, 6

[4] Nicolas Carion et al. End-to-end object detection with transformers. In *ECCV 2020*. 5

[5] Sangwoo Cho, Franck Dernoncourt, Tim Ganter, Trung Bui, Nedim Lipka, Walter Chang, Hailin Jin, Jonathan Brandt, Hassan Foroosh, and Fei Liu. StreamHover: Livestream transcript summarization and annotation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6457–6474, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 2, 5, 6

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311, 2022. 2

[7] Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States, 2022. Association for Computational Linguistics. 6

[8] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004. 6

[9] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *ArXiv preprint*, abs/1907.12763, 2019. 2, 3

[10] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy, 2019. Association for Computational Linguistics. 2

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6201–6210. IEEE, 2019. 5, 6

[12] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: temporal activity localization via language query. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5277–5285. IEEE Computer Society, 2017. 2, 4

[13] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014. 2

[14] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5804–5813. IEEE Computer Society, 2017. 2, 3, 4

[15] Hussain Kanafani, Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. Unsupervised video summarization via multi-source features. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 466–470, 2021. 2

[16] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 706–715. IEEE Computer Society, 2017. 4

[17] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021. 2, 3, 4, 5, 6, 7, 8

[18] Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8211–8225, Online, 2020. Association for Computational Linguistics. 2

[19] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *European Conference on Computer Vision*, pages 447–463. Springer, 2020. 2, 3, 4

[20] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012. 2

[21] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 6

[22] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. 6

[23] Feifan Liu and Yang Liu. Correlation between ROUGE and human evaluation of extractive meeting summaries. In *Proceedings of ACL-08: HLT, Short Papers*, pages 201–204, Columbus, Ohio, 2008. Association for Computational Linguistics. 6

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3042–3051, 2022. 2, 6, 7, 8

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 6

[26] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, 2004. Association for Computational Linguistics. 6

[27] Margaret Mitchell, Pistilli Giada, Jernite Yacine, Ozoani Ezinwanne, Gerchick Marissa, and Rajani et al Nazneen. Bloom: Bigscience large open-science open-access multilingual language model. 2022. 2, 5

[28] Jinwoo Nam, Daechul Ahn, Dongyeop Kang, Seong Jong Ha, and Jonghyun Choi. Zero-shot natural language video localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1470–1479, 2021. 2

[29] Chau Nguyen, Tim French, Wei Liu, and Michael Stewart. SConE: Simplified cone embeddings with symbolic operators for complex logical queries. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11931–11946, Toronto, Canada, July 2023. Association for Computational Linguistics. 2

[30] Yulin Pan, Xiangteng He, Biao Gong, Yiliang Lv, Yujun Shen, Yuxin Peng, and Deli Zhao. Scanning only once: An end-to-end framework for fast temporal grounding in long videos. *arXiv preprint arXiv:2303.08345*, 2023. 2

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 6

[32] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. Technical report, Tech. Rep., Technical report, OpenAI, 2022. 2

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 6

[34] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv preprint*, abs/2112.11446, 2021. 2

[35] Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019. 2

[36] Daniel Rotman, Dror Porat, and Gal Ashour. Robust and efficient video scene detection using optimal sequential grouping. In *2016 IEEE international symposium on multimedia (ISM)*, pages 275–280. IEEE, 2016. 2, 4, 5, 6

[37] Daniel Rotman, Dror Porat, Gal Ashour, and Udi Barzelay. Optimally grouped deep features using normalized cost for video scene detection. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 187–195, 2018. 2, 4, 5, 6

[38] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *ArXiv preprint*, abs/2211.05100, 2022. 2, 6

[39] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 200–216, 2018. 2

[40] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *ArXiv preprint*, abs/2201.11990, 2022. 2

[41] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035, 2022. 3, 4

[42] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5179–5187. IEEE Computer Society, 2015. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. 5

[44] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions*

on pattern analysis and machine intelligence, 41(11):2740–2755, 2018. 2

[45] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *European Conference on Computer Vision*, pages 34–51. Springer, 2020. 2

[46] Huijuan Xu, Kun He, Bryan A. Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 9062–9069. AAAI Press, 2019. 2

[47] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S. Davis. MAN: moment alignment network for natural language moment retrieval via iterative graph adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1247–1257. Computer Vision Foundation / IEEE, 2019. 2

[48] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6543–6554, Online, 2020. Association for Computational Linguistics. 2

[49] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 12870–12877. AAAI Press, 2020. 2

[50] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068, 2022. 2, 5, 6

[51] Minghang Zheng, Yanjie Huang, Qingchao Chen, Yuxin Peng, and Yang Liu. Weakly supervised temporal sentence grounding with gaussian-based contrastive proposal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15555–15564, 2022. 2