# Temporal Localization of Moments in Video Collections with Natural Language

Victor Escorcia<sup>1\*</sup> Mattia Soldan<sup>1</sup> Josef Sivic<sup>2,3\*</sup> Bernard Ghanem<sup>1</sup> Bryan Russell<sup>2</sup>

<sup>1</sup>KAUST <sup>2</sup>Adobe Research <sup>3</sup>INRIA

#### **Abstract**

In this paper, we introduce the task of retrieving relevant video moments from a large corpus of untrimmed, unsegmented videos given a natural language query. Our task poses unique challenges as a system must efficiently identify both the relevant videos and localize the relevant moments in the videos. This task is in contrast to prior work that localizes relevant moments in a single video or searches a large collection of already-segmented videos. For our task, we introduce Clip Alignment with Language (CAL), a model that aligns features for a natural language query to a sequence of short video clips that compose a candidate moment in a video. Our approach goes beyond prior work that aggregates video features over a candidate moment by allowing for finer clip alignment. Moreover, our approach is amenable to efficient indexing of the resulting clip-level representations, which makes it suitable for moment localization in large video collections. We evaluate our approach on three recently proposed datasets for temporal localization of moments in video with natural language extended to our video corpus moment retrieval setting: DiDeMo [16], Charades-STA [10], and ActivityNetcaptions [22]. We show that our CAL model outperforms the recently proposed Moment Context Network [16] on all criteria across all datasets on our proposed task, obtaining an 8%-85% and 11%-47% boost for average recall and median rank, respectively, and achieves  $5 \times$  faster retrieval and  $8 \times$  smaller index size with a 1M video corpus.

#### 1. Introduction

Consider the natural language query shown in Figure 1a. Recent work has introduced the task of natural language moment retrieval in video [10, 16], where the goal is to return a relevant moment in an untrimmed, unsegmented single video corresponding to a natural language query. While current methods retrieve moments from a single video, users often have large stores of untrimmed, unsegmented videos that they want to query. In this paper, we propose the task of temporally localizing relevant moments in a large

*corpus* of videos given a natural language query. Progress on this task could enable applications in video search and retrieval, such as video editing and surveillance.

Our task is challenging as we need to efficiently and accurately find both the video and the exact moment in the video that aligns with a natural language query. While one could attempt to scale prior approaches for localizing a relevant moment in a single, untrimmed video given a natural language query [4, 10, 16, 17, 23, 24] to a large video corpus, such an attempt would face two difficulties. First, we need the ability to index and efficiently retrieve relevant moments in videos. As current efficient indexing techniques rely on approximating the Euclidean distance between descriptors [11, 13, 18], they cannot be readily plugged into video moment retrieval systems that rely on computing similarities using, often complicated, neural network architectures [4, 10, 23, 24]. Second, the index size needs to scale efficiently relative to the size of the video corpus. While the Moment Context Network (MCN) [16] allows for efficient retrieval due to the model's use of Euclidean distance for comparing language and video features, it requires indexing and storing all possible-length moments in a video. Such a requirement yields large and non-practical video index sizes. While indexing only action proposals [8, 9] may be a solution to reducing the index size, such methods may discard relevant moments that a user may want to query.

In this work, we propose Clip Alignment with Language (CAL), a model that represents a video moment as a series of short video clips and aligns a natural language query to the moment's clips with a clip-alignment cost. Our approach is illustrated in Figure 1b. Our clip-alignment cost compares language and clip features using squared-Euclidean distance, which allows for efficient indexing and retrieval of the video clips. Moreover, aligning language features to short video clips within a video moment allows for finer temporal alignment compared to methods that extract only an aggregate feature from the entire video moment. At query time, we propose a two-stage approach consisting of efficient retrieval followed by more expensive reranking to maintain recall accuracy. We achieve efficiency by an approximate strategy that retrieves relevant candidate clips for a language query using efficient approximate near-

<sup>\*</sup>Work done at Adobe during VE's internship.

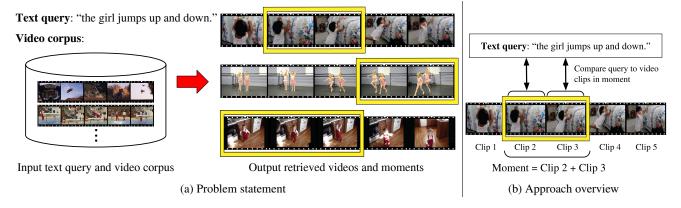


Figure 1: **Problem statement and approach overview.** (a) Given a natural language query, we seek to find relevant videos from a large corpus of untrimmed, unsegmented videos and temporally localize relevant moments within the returned videos. (b) Our approach aligns natural language queries to a sequence of short video clips that compose the candidate moment.

est neighbour search. Then, for re-ranking, we apply the full clip-alignment cost on all variable-length moments in the temporal proximity of the retrieved candidate clips. Furthermore, representing moments as a series of short video clips allows us to overcome the need for indexing all possible variable-length moments while at the same time retrieveing any possible moment in a video.

Contributions. Our contributions are twofold: we propose (i) the task of natural language video corpus moment retrieval and (ii) a model (CAL) that aligns video clips to a language query while allowing for efficient retrieval followed by re-ranking in the large-scale video corpus moment retrieval setting. We demonstrate the effectiveness of our approach by extending three datasets to the video corpus retrieval setting: DiDeMo [16], Charades-STA [10], and ActivityNet-captions [22]. We show that our CAL model in an exhaustive setting out-performs MCN [16] on all criteria across all datasets, yielding an 8%-85% and 11%-47% boost for average recall and median rank, respectively. Furthermore, for a corpus of 1M videos, we achieve 5× faster retrieval and 8× smaller index size over MCN.

## 2. Related work

cessing and video, an area that has received much recent attention. Our work is closest to the tasks of, given a natural language query, retrieving short video clips from a large collection and localizing moments in a single untrimmed, unsegmented video. We describe related work for both tasks. **Video clip retrieval with natural language.** Recently, datasets of short video clips with accompanying natural language have emerged. Examples include the MPII movie description dataset as part of the large scale movie description challenge (LSMDC) dataset [30] and the MSR-VTT dataset [36]. Example recent approaches leverage detected

Our work lies at the intersection of natural language pro-

concepts in videos [38], hierarchical alignment and attention [37], learning a mixture of embedding experts [26], and dual deep encoding for zero-example retrieval [7]. However, all of these approaches do not search for moments within untrimmed, unsegmented videos.

Localizing moments in a single video with natural language. Datasets of videos with temporally aligned text [10, 16, 17, 22, 29] have been used for aligning movie scripts, textual instructions, and sentences in a paragraph with a single video [3, 25, 32, 39], video object segmentation [20], and retrieving moments in a single video given a text query [4, 5, 10, 16, 17, 23, 24]. Our work is closest to the latter. As we will discuss in Section 3, the MCN [16] and CTRL [10] models aggregate features over a video moment before comparing to a feature for the language query. Our clip-based alignment approach allows for finer alignment between the moment and query. More recent approaches have integrated alignment of clips with language queries inside a neural network as part of a temporal modular network [23] or joint alignment with temporal attention [4, 5, 24, 35]. As we will show, these approaches are not amenable to efficient search and retrieval at large scale. Our approach overcomes both limitations and allows for efficient indexing and retrieval over large video collections.

## 3. Clip Alignment with Language (CAL)

Our goal is, given a natural language query q, to return a video  $v \in \mathcal{V}$  from a corpus  $\mathcal{V}$  and temporal endpoints  $\tau = \left(\tau^{(S)}, \tau^{(E)}\right)$  that temporally localize the language query in the video where  $\tau^{(S)}$  and  $\tau^{(E)}$  are start and end points, respectively. If the video corpus  $\mathcal{V}$  comprises a single video, then the task is *single video moment retrieval* (as proposed in [10, 16]). If it is a collection of videos, the task is *video corpus moment retrieval* (our proposed task). Our approach for the video corpus moment retrieval task

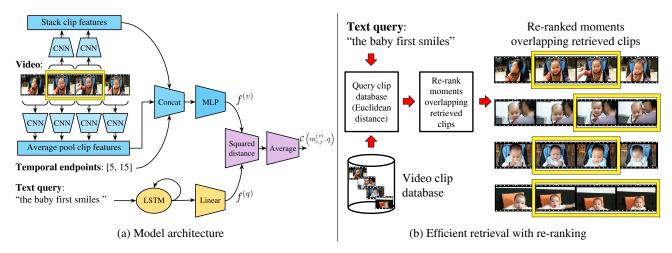


Figure 2: **Model and system for indexing and retrieval.** (a) Our model is a neural network that aligns video clip features with a language query feature. (b) Our approach allows for efficient retrieval and storage of video moments in a database. See text for details.

consists of two stages – efficient retrieval followed by more expensive re-ranking. We first describe our Clip Alignment with Language (CAL) model and then describe how it is used for efficient two-stage retrieval with re-ranking (Section 3.1).

We cast the temporal localization problem as one of retrieving a sequence of relevant short clips from a video. Let video v be comprised of a sequence of  $N_v$  short uniformlength clips  $v=\{c_1,\cdots,c_{N_v}\}$  ordered in time with corresponding temporal endpoints  $\mathbf{T}^{(v)}=\{\tau_1,\cdots,\tau_{N_v}\}$ . Depending on the dataset, the clips may be 3-5 seconds in duration. We seek to return a relevant moment  $m_{i,j}^{(v)}\subseteq v$  from a video v consisting of a consecutive sequence of clips  $m_{i,j}^{(v)}=\{c_i,\cdots,c_j\}$  for  $1\leq i\leq j\leq N_v$  with moment temporal endpoints  $\tau_{i,j}^{(M)}=\left(\tau_i^{(S)},\tau_j^{(E)}\right)$  for  $\tau_i,\tau_j\in\mathbf{T}^{(v)}$ , which closely correspond to the ground truth temporal endpoints  $\tau_q$  for input natural language query q. Our temporal localization problem can be formulated as an optimization over an alignment cost  $\mathcal{C}$ ,

$$\min_{\substack{v \in \mathcal{V}, i, j \\ \text{s.t. } 1 \le i \le j \le N_v}} \mathcal{C}\Big(m_{i,j}^{(v)}, q\Big), \tag{1}$$

where we aim to find the best video v and sequence of clips in the video  $m_{i,j}^{(v)} = \{c_i, \cdots, c_j\}$  minimizing the alignment cost between the clips and language query q. In general, alignment cost  $\mathcal C$  may align two variable-length sequences of features extracted over video clips and words in a sentence. In this work, we consider a special case of alignment by learning to match a single feature extracted over a language query to a sequence of features extracted over video clips. Figure 1b illustrates our overall approach.

Let 
$$f^{(v)} = \left\{ f_1^{(v)}, \cdots, f_{N_v}^{(v)} \right\}$$
 be a set of features for the

clips  $\{c_1, \dots, c_{N_v}\}$  of a video v and  $f^{(q)}$  be a feature for the language query q. As we may have a variable number of clips in a moment  $m_{i,j}^{(v)}$ , we define the alignment cost for the moment as the average squared-Euclidean distance between the language feature and the moment's clip features,

$$C(m_{i,j}^{(v)}, q) = \frac{1}{Z} \sum_{k=-i}^{j} \|f_k^{(v)} - f^{(q)}\|^2, \qquad (2)$$

where Z=j-i+1 is the number of clips in the moment. To prevent the degeneracy of always returning single-clip moments, we enforce that moments have at least two clips, *i.e.*, i < j. We used shorter-length clips and observed that this requirement does not degrade performance in practice.

Our alignment cost has two advantages over previous ones. First, our cost is separable with respect to the video clips, *i.e.*, our cost is expressed as a sum of terms over clips, allowing for finer clip alignment. Second, the video clips are indexable since the terms in the cost are Euclidean distances, which can be computed efficiently. We discuss the advantages of both properties and relate to prior work next. **Discussion.** In prior work [10, 16], the language feature is compared to an aggregated feature over the video moment,

$$C_{\text{agg}}\left(m_{i,j}^{(v)},q\right) = \Phi\left(\Psi\left(f_i^{(v)},\cdots,f_j^{(v)}\right),f^{(q)}\right), \quad (3)$$

where  $\Psi$  aggregates the clip features  $f_i^{(v)}, \cdots, f_j^{(v)}$  into an embedded feature for the candidate moment and  $\Phi$  compares the aggregated video moment and language features. In MCN [16], squared-Euclidean distance  $(\Phi)$  is used to compare aggregated video moment  $(\Psi)$  and language features. In CTRL [10], aggregated video moment features  $(\Psi)$  and language features pass through a neural network

 $(\Phi)$ . One drawback of these formulations is that the language feature is compared to an aggregated feature over the entire moment and does not have the ability to align to the individual clips in the moment.

In recent work [4, 5, 10, 23, 24, 35], a joint model over language and video features is used to return the alignment cost,

$$C_{\text{joint}}\left(m_{i,j}^{(v)},q\right) = \Phi\left(\left(f_{i}^{(v)},\cdots,f_{j}^{(v)}\right),f^{(q)}\right), \quad (4)$$

where  $\Phi$  is a neural network. These neural networks perform early fusion and incorporate an attention mechanism into the model. While these approaches have achieved early success for single video moment retrieval, they currently cannot perform efficient indexing and retrieval at large scale (e.g., over millions of untrimmed and unsegmented videos) due to their reliance on a neural network for comparing video and language features, i.e., it would be too expensive to compute at test time for a large video corpus.

Model details. Figure 2a illustrates our model. Clip features  $f^{(v)}$  are computed and compared to language features  $f^{(q)}$  for query q using squared-Euclidean distance and then averaged. For each clip feature  $f_k^{(v)}$ , we concatenate visual features computed over the temporal extent of the clip with a context feature and (optionally) temporal endpoints for the moment, which are then passed through a multilayer perceptron (MLP). As in MCN [16], for context features, we average pool clip features over the entire video. The language feature  $f^{(q)}$  is computed as in MCN [16], where the output of the last hidden layer of an LSTM with word embedding features for each query word as inputs passes through a linear mapping. We use pre-computed features for the visual and word-embedding features, so our model parameters comprise the MLP, LSTM, and hidden-layer linear mapping. Note that our CAL model has the same number of parameters as MCN, which allows for direct comparison of the two approaches.

**Training.** We seek to have our CAL model rank correctly aligned video and language query training examples better than misaligned examples. To achieve this goal, we define a ranking loss for our training objective. Let  $\mathcal{P} = \left\{ \begin{pmatrix} m_{i,j}^{(v)}, q \\ k \end{pmatrix}_k^N \right\}_{k=1}^N$  be a training set of N aligned video moment and natural language query pairs. For a positive training example  $p \in \mathcal{P}$ , we define an intra-video negative set  $\mathcal{N}_{\text{intra}}^{(p)}$  consisting of video moments in the training example video not aligned to the language query training example. Similarly, we define an inter-video negative set  $\mathcal{N}_{\text{inter}}^{(p)}$  consisting of video moments from completely different videos in the training set. We define a set  $\Gamma$  where each member is the triple  $(p, n, n') \in \Gamma$  such that  $p \sim \mathcal{P}$ ,  $n \sim \mathcal{N}_{\text{intra}}^{(p)}$ , and  $n' \sim \mathcal{N}_{\text{inter}}^{(p)}$ . We optimize a training loss  $\mathcal{L}_{\theta}$  for model parameters  $\theta$ , where the loss is a sum of ranking losses over

intra- and inter-video negatives for all sampled triples,

$$\mathcal{L}_{\theta} = \sum_{(p,n,n')\in\Gamma} \mathcal{L}^{R} \Big( \tilde{\mathcal{C}}_{p}, \tilde{\mathcal{C}}_{n} \Big) + \lambda \mathcal{L}^{R} \Big( \tilde{\mathcal{C}}_{p}, \tilde{\mathcal{C}}_{n'} \Big), \quad (5)$$

where  $\mathcal{L}^R(x,y) = \max\left(0,x-y+b\right)$  is a ranking loss,  $\tilde{\mathcal{C}}_p = \mathcal{C}\left(m_{i,j}^{(v)},q\right)$  is the alignment Cost (2) for positive training example  $p = \left(m_{i,j}^{(v)},q\right)$  (similarly  $\tilde{\mathcal{C}}_n$  and  $\tilde{\mathcal{C}}_{n'}$  for intra- and inter-negative training examples n and n', respectively), and b and  $\lambda$  are margin and weighting hyperparameters, respectively. We set b = 0.1 and  $\lambda = 0.4$  using cross validation. We optimize Loss (5) using stochastic gradient descent with momentum by uniform sampling over positive and intra-/inter-negative triples.

### 3.1. Efficient retrieval with re-ranking

For inference, one can evaluate Cost (2) exhaustively over all possible moments in all videos. While this routine was used in MCN [16] to localize moments in a single video, this exhaustive strategy does not efficiently scale to localizing moments in a large video corpus. To achieve efficient retrieval while maintaining recall accuracy, we propose a two-stage approach consisting of an efficient retrieval stage followed by a more expensive re-ranking stage.

Our CAL model allows for efficient indexing and retrieval of video moments for a natural language query since it relies on comparing video and language features with a sum of Euclidean distances. This is important for our application as we may potentially want to search through a large corpus comprising millions of untrimmed, unsegmented videos. As noted in our earlier discussion, approaches that align video and language features with neural networks currently do not extend to large-scale indexing applications, which is a key difference from our approach.

Our strategy for implementing the efficient retrieval stage with our approach is to index video clip features  $f^{(v)}$ for each video v. At query time, the system retrieves moments in a greedy fashion by retrieving top clips corresponding to the language feature  $f^{(q)}$  for query q. For the re-ranking stage, we score and re-rank the set of moments containing the retrieved clips with the more expensive Cost (2). To boost recall during re-ranking, we retrain our CAL model using the top-retrieved moments from the retrieval stage. This strategy is illustrated in Figure 2b. While this efficient retrieval with re-ranking strategy is not guaranteed to retrieve the best moment in terms of Cost (2), we are able to effectively return the correct moment in practice (see Section 4). Moreover, our approach allows for retrieval of any moment from any video, which is in contrast to proposal-based methods [9] that discard clips from

While MCN [16] can also index features corresponding to video moments, our approach offers an advantage

Dataset	Clip	Max moment	Stride	Avg. video	Re	call
	length	length	length	length	IoU=0.5	IoU=0.7
DiDeMo [16]	2.5 secs.	6 clips	5 secs.	29 secs.	100.00	100.00
Charades-STA [10]	3 secs.	8 clips	3-6 secs.	31 secs.	99.62	88.79
ActivityNet-captions [22]	5 secs.	26 clips	5-40 secs.	120 secs.	89.26	80.24

Table 1: Dataset settings and statistics. Right – oracle upper bound. See text for details.

with respect to the index size. For our clip-alignment approach, only N clips are indexed for a video. For MCN, all possible-length moments must be indexed as the model relies on aggregated features over the moments. Assuming maximum moment length of K clips results in an index of size  $NK - \frac{1}{2}K(K-1)$  for a video. For the datasets considered in this paper, this results in  $6\times-12\times$  increase in the index size. This increase is expected to get even worse when longer, more complex moments need to be considered, thus increasing the value of K.

# 4. Experiments

In this section, we show qualitative and quantitative results on our proposed task of retrieving relevant moments from a large corpus of videos for a natural language query. We start by showing results on our proposed video corpus moment retrieval task in an exhaustive setting (Section 4.1). Next, we show results using efficient retrieval with re-ranking (Section 4.2). We show additional results in the supplemental.

# 4.1. Video corpus moment retrieval

Our first experiment consists of exhaustively evaluating a method over an entire video corpus. More specifically, given a language query, we evaluate the alignment cost exhaustively over all possible moments in all videos. We describe in detail our evaluation setup and results.

Datasets. We evaluate on three datasets that have natural language sentences aligned in time to videos and have been proposed for the single video moment retrieval task: DiDeMo [16], Charades-STA [10], and ActivityNetcaptions [22]. These datasets have a large number of temporally aligned natural language sentences with large (open) vocabulary. Moreover, the videos depict general scenes and are not constrained to a specific scene type. DiDeMo consists of unedited video footage from Flickr with sentences aligned to unique moments in the video (i.e., the sentences are referring). There are 10642 videos and 41206 sentences in the dataset and we use the published splits over videos (train-8511, val-1094, test-1037). Note that moment start and end points are aligned to five-second intervals and that the maximum annotated video length is 30 seconds. Charades-STA builds on the Charades dataset [33] consisting of unedited videos of humans acting from scripts. There are 6670 videos and 16124 sentences in the dataset and we use the published splits over videos (train–5336, test–1334). The videos are typically longer in length than the ones in DiDeMo and sentences from the scripts are aligned in time and may not be referring. ActivityNet-captions builds on the ActivityNet dataset [15] consisting of YouTube video footage. There are 14926 videos and 71942 sentences in the dataset and we use the published splits over videos (train–10009, val–4917). Videos are typically longer in length than DiDeMo and Charades-STA and may be edited; the sentences may not be referring.

We adapt the DiDeMo, Charades-STA, and ActivityNetcaptions datasets used for single video moment retrieval to our video corpus moment retrieval task. Specifically, a method must correctly identify both the video and the moment within the video corresponding to a ground truth natural language query.

**Evaluation criteria.** We adopt the criteria proposed in TALL [10], where average recall at K (R@K) is reported over all language queries. We measure recall for a particular language query by determining whether one of the top K-scoring retrieved moments sufficiently overlaps with the ground truth annotation (recall will be 0 or 1). A retrieved moment sufficiently overlaps with a ground truth annotation if the ratio of the temporal intersection over union (IoU) exceeds a specified threshold. We average the recall values across all language queries to obtain the average recall at K. We report R@K over all retrieved moments from the video corpus for  $K \in \{1, 10, 100\}$  and IoU  $\in \{0.5, 0.7\}$ . In addition, we report the median rank for the correct retrieval. While the annotations are not exhaustive (i.e., a given natural language query may appear in a video but not be annotated), reporting over different values of K allows us to take into account the missing annotations. Finally, note that DiDeMo [16] has multiple annotations for each sentence corresponding to different human judgements. We account for the multiple annotations by requiring that a correct detection must overlap with at least two of the human judgements with the specified IoU, which can be satisfied for all sentences in the val and test sets.

**Implementation details.** To obtain candidate moments in a video, we need to specify the clip length, maximum number of clips in a moment, and how frequently to extract clips in a video (temporal stride). Table 7 shows the set-

	DiDeMo [16] (test)			(	Charades-S	TA [10] (te	st)	ActivityNet-captions [22] (val)				
	K=1	K=10	K=100	$MR\downarrow$	K=1	K=10	K=100	$MR\downarrow$	K=1	K=10	K=100	$MR\downarrow$
IoU=0.5												
Chance	0.00	0.10	1.99	4233	0.01	0.09	1.09	6393	0.00	0.02	0.18	46718
Moment prior	0.02	0.22	2.34	2527	0.02	0.17	1.63	4906	0.01	0.05	0.47	32597
TEF-only	0.05	0.32	2.58	2426	0.04	0.34	2.87	3809	0.01	0.05	0.70	24447
MCN	0.36	2.15	12.47	1057	0.08	0.52	2.96	6540	0.02	0.18	1.26	24658
Ours	0.74	3.90	16.51	831	0.15	0.75	4.39	5486	0.01	0.21	1.58	16150
MCN (TEF)	0.88	5.16	26.23	340	0.13	0.96	6.05	3221	0.12	0.75	4.54	7850
Ours (TEF)	0.97	6.15	28.06	325	0.23	1.39	7.03	2960	0.21	1.32	6.82	5200
IoU=0.7												
Chance	0.00	0.02	0.64	13434	0.00	0.03	0.39	17070	0.00	0.01	0.06	130371
Moment prior	0.02	0.17	1.99	3234	0.01	0.05	0.56	11699	0.00	0.03	0.26	82488
TEF-only	0.03	0.27	2.12	3209	0.01	0.16	1.57	8737	0.01	0.03	0.39	57919
MCN	0.28	1.55	9.03	1423	0.04	0.31	1.75	10262	0.01	0.09	0.70	40474
Ours	0.58	2.81	12.79	1148	0.06	0.42	2.78	8627	0.01	0.10	0.90	26652
MCN (TEF)	0.58	4.12	21.03	500	0.08	0.63	4.24	5567	0.07	0.48	3.04	17101
Ours (TEF)	0.66	4.69	22.89	449	0.12	1.00	4.91	4970	0.12	0.89	4.79	11596

Table 2: Video corpus retrieval quantitative results (exhaustive setting). We show average recall for top K retrievals and median retrieval rank (MR, lower is better) on DiDeMo [16], Charades-STA [10], and ActivityNet-captions [22] datasets for different baselines and our model. Top section - IoU=0.5, bottom section - IoU=0.7. More details in text.

tings for the video clip length, maximum moment length, and temporal stride used for the evaluated datasets. We set the values for each dataset to maximize an oracle detector where a sequence of (non-overlapping) clips are aligned with the ground truth moments, while minimizing computational cost. We set the temporal stride to 5 seconds for all moments in DiDeMo and proportionally to the moment length d in the other datasets computed as  $0.3 \times d$  (rounded to the nearest clip boundary) as longer-length moments do not need fine temporal stride. Given the settings in Table 7, the number of candidate moments for each dataset are: DiDeMo - 21,777, Charades-STA - 49,465, ActivityNetcaptions - 460,265. For approaches, such as MCN, that index all possible moments, these numbers would be the index sizes for the evaluated datasets. We report the performance of the oracle detector in Table 7. While the oracle's returned endpoints align to clip boundaries and do not have the ability to exactly align to ground truth endpoints, we note that the oracle detector still achieves high performance. Also note that humans may not generally agree on temporal endpoints [1]. For all approaches, we evaluate their alignment cost for every moment in a video and perform nonminimum suppression with temporal IoU threshold chosen empirically for each dataset (DiDeMo – 1.0, Charades-STA -0.6, ActivityNet-captions -0.5).

Our model uses ResNet-152 features [14] computed over the video clips. We computed ResNet *pool*5 features over video frames extracted at 5 fps and max-pooled the features over the clips. Empirically, we observed max pooling outperformed average pooling. We used Glove word-

embedding features [27] for the words in the language query. For temporal endpoints, we normalized the start and end points relative to the video length as in MCN [16] to obtain temporal endpoint features (TEFs). For stochastic gradient descent, we set momentum to 0.95 and used a schedule of lowering an initial learning rate of 0.05 by a factor of 0.1 every 30 epochs; training stopped at 108 epochs. We formed mini-batches with 128 positive/negative examples. We selected intra-negatives such that their overlap with the ground-truth moment is lower than a given IoU value. For DiDeMo, we used IoU=1 since the ground truth is aligned to five-second intervals: for Charades-STA and ActivityNetcaptions we used IoU=0.35. Similarly, inter-negatives were selected from the same temporal location as the groundtruth moment, whenever possible, in another video selected at random from the entire dataset.

**Baselines.** We compare our CAL model to the MCN baseline [16] run exhaustively over all moments in the corpus in addition to chance and moment frequency prior baselines. For chance, we return moments across all videos sampled from a uniform distribution. We compute the moment frequency prior as in Hendricks *et al.* [16] for each dataset by discretizing the range of video-length-normalized start and end points and histograming the training ground truth moments. We output the probability for each video's moment; ties across different videos are broken by sampling a uniform distribution. We train the MCN model using the same procedure as for single video moment retrieval [16].

**Results.** Quantitative results are shown in Table 2. First, we observe that our CAL model without TEF is on par or out-



Figure 3: **Video corpus retrieval qualitative results.** We show top temporally localized moment retrievals for different natural language queries across all videos in DiDeMo [16] and Charades-STA [10]. Ground truth annotations appear as a green line below a video, best viewed in color.

performs MCN across all datasets on all criteria; for some criteria there is greater than twofold increase in accuracy. These results indicate the effectiveness of our approach on visual and language cues alone without temporal endpoints. When we include TEF, accuracy for CAL improves and outperforms all baselines across all datasets, validating the effectiveness of our approach on our newly proposed task. In particular, we obtain an 8%-85% and 11%-47% boost over MCN with TEF for average recall and median rank, respectively. We note that the performance is low for all methods as annotations are not exhaustive and there are many more candidate moments to search over than in the single video retrieval task, illustrating the great difficulty of the video corpus retrieval task.

Qualitative results are shown in Figure 3. Notice how we are able to retrieve relevant moments for the different language queries. For example, the queries "the person is eating a sandwich" and "person drink out of the glass" retrieves well-localized moments depicting people eating or drinking, including single ground truth annotated moments for the queries. The query "person pour sauce on first piece of meat in skillet" shows example failures of our system. While the top retrieval is correct, the other retrievals depict different parts of the language query, such as "sauce", "meat', and "pour", but not the entire query.

What is the effect of TEF on the datasets? We analyze the effect of incorporating the temporal endpoint features (TEFs) into the model. For our analysis, we train the MCN model using only the language features and TEF on all the datasets, *i.e.*, the model does not see appearance features from the video. During testing, we run the model exhaustively over all moments in the corpus. Results are shown in Table 2 ("TEF-only"). We observe that the TEF-only baseline is competitive and on par or out-performs the moment frequency and chance baselines. Moreover, on the single video moment retrieval task (*c.f.*, supplemental), we observe that the TEF-only baseline is a competitive baseline for each dataset and out-performs many of the other

baselines that use video appearance features. The fact that the TEF-only baseline performs so well indicates that there is a strong bias in the datasets as only knowing the language query and the relative position of the moment in the video can allow for reasonably high accuracy. This fact was also observed in early datasets for visual question answering (VQA) [12] and suggests future work to mitigate such dataset bias.

#### 4.2. Efficient retrieval with re-ranking

For our second experiment, we evaluate the efficient retrieval and re-ranking system described in Section 3. In the *retrieval stage*, we retrieve the top 200 moments using a given method. In the *re-ranking stage*, we then re-rank the top-retrieved moments using a given method.

**Evaluation criteria.** Similar to the exhaustive retrieval setting (Section 4.1), we report average recall at  $K \in \{1, 10, 100\}$  on the DiDeMo and Charades-STA datasets for the video corpus moment retrieval task. Note that we do not report median rank as only the top retrieved moments are considered.

Baselines and ablations. We use MCN or CAL for the retrieval stage, followed by MCN or CAL with TEF for the re-ranking stage. We also consider using MEE [26] for the retrieval stage as it performs well on the LSMDC benchmark [30] and outperforms other recent methods [7] on MSR-VTT [36] (c.f., supplemental). We used the publicly available implementation of MEE to retrieve videos from the corpus and turned off flow, face, and audio features in MEE for fair comparison. We tried MEE pre-trained on LSMDC and MSR-VTT, which performed near chance on our task; retraining MEE on the target datasets performed best. During retrieval with MEE, we maintain comparable number of moments for the re-ranking stage by retrieving the top videos such that there are 200 available moments within the retrieved videos. Finally, we consider the approximate retrieval setting where we retrieve the top 200 clips given a language query and consider moments around

Retrieval	Re-ranking	DiI	DeMo [16]	(test)	Chara	Charades-STA [10] (test)			
stage	stage	K=1	K = 10	K=100	K=1	K=10	K=100		
IoU=0.5									
MEE	MCN (TEF)	0.53	3.00	6.52	0.12	0.67	1.75		
MCN	MCN (TEF)	0.92	4.83	17.50	0.19	1.04	4.09		
CAL	MCN (TEF)	0.98	5.94	22.83	0.23	1.41	5.89		
CAL	CAL (TEF)	1.07	6.45	22.60	0.30	1.63	6.03		
CAL	CAL (TEF,re-train)	1.29	6.71	22.51	0.48	1.91	6.85		
Approx. CAL	CAL (TEF,re-train)	1.27	6.39	15.82	0.46	1.24	2.93		
IoU=0.7									
MEE	MCN (TEF)	0.46	2.64	6.37	0.09	0.53	1.63		
MCN	MCN (TEF)	0.64	3.67	13.12	0.13	0.71	2.58		
CAL	MCN (TEF)	0.69	4.63	17.89	0.12	0.89	3.78		
CAL	CAL (TEF)	0.72	4.86	17.60	0.15	1.15	3.71		
CAL	CAL (TEF,re-train)	0.85	4.95	17.73	0.35	1.32	4.49		
Approx. CAL	CAL (TEF,re-train)	0.80	4.95	11.59	0.35	1.05	2.55		

Table 3: **Efficient retrieval with re-ranking quantitative results.** We show average recall for top K retrievals on DiDeMo [16] and Charades-STA [10] datasets for different baselines and our model. Top section - IoU=0.5, bottom section - IoU=0.7. More details in text.

the retrieved clips for the re-ranking stage (Approx. CAL).

**Re-training the re-ranking stage.** For re-training, we take the top retrieved moments from the retrieval stage and sample inter-video negatives from the retrieved moments (instead of over all possible videos). We sample inter-video negatives using an exponential distribution over the moment's rank from the retrieval stage. Finally, we fine tune the re-ranking model initializing with the original model's parameters.

**Results.** We report quantitative results in Table 3. Our CAL for retrieval followed by CAL with TEF and re-training for re-ranking performs best across all criteria on Charades-STA and for  $K \in \{1, 10\}$  on DiDeMo (we are on par for K = 100), demonstrating the effectiveness of our approach for retrieval with re-ranking. Moreover, our twostage approach outperforms the exhaustive approaches in Table 2 for  $K \in \{1, 10\}$ . Note that CAL for retrieval is on par or outperforms MCN and MEE using the same method for the re-ranking stage across all criteria on both datasets. We also tried retrieving 200 videos with MEE followed by MCN with TEF for re-ranking and found that our approach outperforms this baseline on all criteria for Charades-STA and for  $K \in \{1, 10\}$  on DiDeMo (we are on par for K=100). However, note that this baseline has access to significantly  $(21 \times -33 \times)$  more moments for the re-ranking stage, which aids in improving recall. Finally, for approximate retrieval, we maintain similar recall as our best model for  $K \in \{1, 10\}$  on both datasets, demonstrating its effectiveness in an efficient retrieval setting.

**Run time and index size.** We report run time and the retrieval index size for a video corpus containing 1M videos

	Run time (s)	Index size (GB)
MCN	144.7	63.3
CAL	24.6	7.45
MCN / MCN (TEF)	144.7 / 0.4	63.3
CAL / CAL (TEF)	24.6 / 0.3	7.45
Approx. CAL / CAL (TEF)	1.0 / 0.3	7.45

Table 4: **Run time and index size.** Top – exhaustive retrieval; bottom – efficient retrieval with re-ranking.

each containing 20 clips with max moment length of 14 clips for the different methods in Table 4. We observe that CAL is  $5 \times$  faster and has  $8 \times$  smaller index size than MCN. Finally, Approx. CAL with approximate nearest neighbor search [19] has fastest run-time ( $111 \times$  speed up) and smallest index size, demonstrating its efficiency.

#### 5. Conclusion

We have shown a simple yet effective approach for aligning video clips to natural language queries for retrieving moments in untrimmed, unsegmented videos. Our approach allows for efficient indexing and retrieval of video moments on our newly proposed task of search through large video collections. We have quantitatively evaluated on three benchmark datasets extended to our task and shown the effectiveness of our approach over prior work on our proposed task in terms of accuracy and search index size. Our work opens up the possibility of effectively searching video at large scale with natural language interfaces.

**Acknowledgement.** This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2017-3405. We thank Lisa Anne Hendricks, and other members of the IVUL at KAUST.

# References

- H. Alwassel, F. C. Heilbron, V. Escorcia, and B. Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, 2018. 6, 12
- [2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017. 10, 11
- [3] P. Bojanowski, R. Lagugie, E. Grave, F. R. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 2
- [4] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua. Temporally grounding natural sentence in video. In *EMNLP*, 2018. 1, 2, 4, 12, 14
- [5] J. Chen, L. Ma, X. Chen, Z. Jie, and J. Luo. Localizing natural language in videos. In *AAAI*, 2019. 2, 4
- [6] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, April 2017. 10
- [7] J. Dong, X. Li, C. Xu, S. Ji, and X. Wang. Dual dense encoding for zero-example video retrieval. *CoRR*, abs/1809.06181, 2018. 2, 7, 10, 12
- [8] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In ECCV, 2016. 1
- [9] J. Gao, K. Chen, and R. Nevatia. CTAP: Complementary temporal action proposal generation. In ECCV, 2018. 1, 4
- [10] J. Gao, C. Sun, Z. Yang, and R. Nevatia. TALL: Temporal activity localization via language query. In *ICCV*, 2017. 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13
- [11] T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:744–755, 2014.
- [12] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *International Journal of Computer Vision*, 2018. 7
- [13] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, 1998. 1
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016. 6, 10, 11, 12
- [15] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In CVPR, 2015. 5
- [16] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14

- [17] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 1, 2
- [18] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:117–128, 2011. 1
- [19] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. CoRR, abs/1702.08734, 2017. 8
- [20] A. Khoreva, A. Rohrbach, and B. Schiele. Video object segmentation with language referring expressions. In ACCV, 2018. 2
- [21] S. Kornblith, J. Shlens, and Q. V. Le. Do better ImageNet models transfer better? *CoRR*, abs/1805.08974, 2018. 10
- [22] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 1, 2, 5, 6
- [23] B. Liu, S. Yeung, E. Chou, D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Temporal modular networks for retrieving complex compositional activities in videos. In *ECCV*, 2018. 1, 2, 4, 12, 14
- [24] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua. Attentive moment retrieval in videos. In SIGIR, 2018. 1, 2, 4, 12, 13, 14
- [25] A. Miech, J.-B. Alayrac, P. Bojanowski, I. Laptev, and J. Sivic. Learning from video and text via large-scale discriminative clustering. In *ICCV*, 2017. 2
- [26] A. Miech, I. Laptev, and J. Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *CoRR*, abs/1804.02516, 2018. 2, 7, 10, 12
- [27] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 6, 10, 11, 12
- [28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018. 10, 11
- [29] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguis*tics (TACL), 1:25–36, 2013. 2
- [30] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In CVPR, 2015. 2, 7
- [31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015. 10
- [32] D. Shao, Y. Xiong, Y. Zhao, Q. Huang, Y. Qiao, and D. Lin. Find and focus: Retrieve and localize video events with natural language queries. In ECCV, 2018. 2
- [33] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 5
- [34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 10, 11, 12
- [35] H. Xu, K. He, B. A. Plummer, L. Sigal, S. Sclaroff, and K. Saenko. Multilevel language and vision integration for text-to-clip retrieval. In AAAI, 2019. 2, 4, 12, 13

- [36] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In CVPR, 2016. 2, 7, 10, 12
- [37] Y. Yu, J. Kim, and G. Kim. A joint sequence fusion model for video question answering and retrieval. In ECCV, 2018.
- [38] Y. Yu, H. Ko, J. Choi, and G. Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In CVPR, 2017. 2
- [39] B. Zhang, H. Hu, and F. Sha. Cross-modal and hierarchical modeling of video and text. In ECCV, 2018. 2

## **Appendix**

We complement our work with the following:

- A video summarizing our work. We show additional qualitative results generated by our Clip Alignment with Language model (CAL). The codec of the video is H264 MPEG-4 AVC with resolution 1280 × 738 and frame rate of 60 fps. We recommend viewing the video with VLC media player <sup>1</sup>.
- Details about our CAL architecture (Section 6).
- Feature ablations (Section 7).
- Results of Mixture of Embedding Experts [26] (MEE) (Section 8).
- Single video retrieval results (Section 9).

#### 6. Model architecture details

Our Clip Alignment with Language model architecture was built on top of the insights of MCN [16]. In particular, the multilayer perceptron (MLP) of the visual stream is formed by two linear layers with a ReLU non-linearity in between. The number of hidden units is 500 and 100, respectively. Note that the size of the embedding space corresponds to the number of units in the second linear layer which matches those in the linear layer of the text query stream. The LSTM layer contains 1000 hidden units and follows the recurrent equation presented by Donahue *et al.* [6]. Note that we did not observe significant improvements in performance by increasing the depth and capacity of our architecture.

Figure 4 illustrates our CAL architecture with the tensor shapes across the model. In this particular instantiation, we assume the feature vector produced by the CNN is of size 2048. Note that we tile the output of the *Average pool clip features*, and the *temporal endpoints* to match the number of clips in a given moment (for example, we tile to length two to match the two clips in the example illustrated in Figure 4) before the *Concat* operation.

We train our architecture in an end-to-end fashion using supervised learning by updating all the parameters except for the CNN weights (*c.f.*, training paragraph of Section 3 in the main submission).

#### 7. Feature ablations

In this section, we show quantitative results of different visual and language features on our proposed task of retrieving relevant moments from a large corpus of videos for a natural language query. Our experiments consist of evaluating the alignment cost exhaustively over all possible moments in all videos on DiDeMo (val set).

Which video features perform best? Table 5 (top part) shows an ablation over video features. We use our CAL model as the base model with Glove language features; we do not use temporal endpoint features. We evaluate the VGG features [34], along with ResNet-152 features [14]. We use the same process outlined in Section 4.1 to compute the ResNet and VGG features (*c.f.* paragraph implementation details in main submission). Similar to [16], we extracted *fc7* features from VGG-16 pre-trained on Imagenet [31].

We observe that ResNet-152 features perform best across all evaluation criteria. The strength of ResNet features for this task is consistent with the finding that ResNet without fine tuning is a good stand-alone base feature for image recognition tasks [21].

Which language features perform best? Table 5 (bottom part) shows an ablation over language features. We use our CAL model as the base model with ResNet video features; we do not use temporal endpoint features. We evaluate Glove [27], FastText [2], and ELMO [28] word embedding features using their publicly available code. We observe that Glove and FastText word embedding features perform best. Since Glove performs similarly to FastText, we used Glove for all experiments in our work.

# 8. Mixture of Embedding Experts for text-tovideo retrieval

In this section, we provide more details about the Mixture of Embedding Experts baseline (MEE) used in section 4.2 (*c.f.*, main submission) [26]. This model belongs to the family of methods of video clip retrieval with natural language described in Section 2 in the main submission. In a nutshell, given a natural language query, MEE retrieves the most similar trimmed video clip that aligns with the given query. This approach, by itself, falls short of addressing our proposed task of retrieving relevant moments from a large corpus of untrimmed, unsegmented videos. Thus, we paired MEE with a model for localizing moments in a single video in a two-stage fashion to fulfill the requirement of our task.

We chose MEE over other methods as it performs well

https://www.videolan.org/vlc/index.html

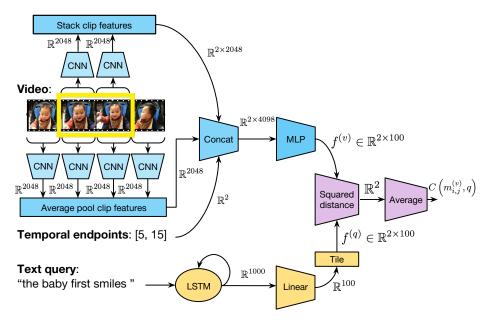


Figure 4: Our model architecture with tensor shapes. See text for details.

Modality	Feature	IoU=0.5				IoU=0.7				
		K=1	K=10	K=100	$MR\downarrow$	K=1	K=10	K=100	$MR\downarrow$	
Visual	VGG [34]	0.44	2.68	11.28	1356	0.30	1.89	8.62	1944	
visuai	ResNet-152 [14]	0.71	4.00	16.48	882	0.48	2.92	12.94	1214	
Languaga	ELMo [28]	0.49	3.03	13.84	1077	0.33	2.13	10.41	1559	
Language	FastText [2]	0.73	3.64	15.44	959	0.57	2.51	12.02	1315	
	Glove [27]	0.71	4.00	16.48	882	0.48	2.92	12.94	1214	

Table 5: Video (top) and language (bottom) feature ablation. DiDeMo (val) results on our CAL model. See text for details.

on LSMDC benchmark and outperforms other recent methods [7] on MSR-VTT [36]. Table 6 "(MSR-VTT)" shows the quantitative comparison in the text to video retrieval task on the MSR-VTT corpus. We used the publicly available implementation of MEE to retrieve videos from the corpus and turned off flow, face and audio features in MEE for fair comparison.

For our two-stage retrieval baseline of moments from a video corpus for a natural language query, we tested MEE models pre-trained on MSR-VTT and the corresponding dataset. Table 6 "(DiDeMo, Charades-STA)" summarizes the results of these experiments. We observed that MEE pre-trained on MSR-VTT performed near chance; while re-training MEE on the target datasets performed the best. Thus, we used the latter setup in the rest of the experiments in our main submission.

# 9. Single video moment retrieval

In this section, we show quantitative results of our CAL architecture on the existing task of retrieving a moment from a single video given a natural language query.

**Datasets.** We evaluate on two datasets that have natural language sentences aligned in time to videos and have been proposed for the single video moment retrieval task: DiDeMo [16], Charades-STA [10]. These datasets have a large number of temporally aligned natural language setnecnes with large (open) vocabulary. For more details about these datasets, refer to Section 4.1 (*c.f.*, main submission).

**Evaluation criteria.** We adopt the criteria proposed in [10], where average recall at K (R@K) is reported over all language queries. We measure recall for a particular language query by determining whether one of the top K-scoring retrieved moments sufficiently overlaps with the ground truth annotation (recall will be 0 or 1). A retrieved moment sufficiently overlaps with a ground truth annotation if the ra-

Dataset	Method	R@1	R@10	R@100	MR↓
	Chance	0.10	0.50	1.00	500
MSR-VTT [36]	Dual Encoding [7]	7.70	22.00	31.80	32
	MEE [26]	11.90	34.00	48.10	11
	Chance	0.10	0.48	0.95	519
DiDeMo [16]	MEE (trained on MSR-VTT)	0.10	0.40	0.94	532
	MEE (trained on DiDeMo)	0.88	3.65	6.63	186
	Chance	0.08	0.38	0.75	667
Charades-STA [10]	MEE (trained on MSR-VTT)	0.10	0.38	0.83	664
	MEE (trained on Charades-STA)	0.48	1.78	3.08	442

Table 6: **Text to video retrieval results.** We show Recall@K on MSR-VTT [36], DiDeMo [16], and Charades-STA [10] with RGB features. See text for more details.

Dataset	Clip	Max moment	Stride	Avg. video	Recall		
	length	length	length	length	IoU=0.5	IoU=0.7	
DiDeMo [16]	5 secs.	6 clips	5 secs.	29 secs.	100.00	100.00	
Charades-STA [10]	3 secs.	8 clips	3-6 secs.	31 secs.	99.62	88.79	

Table 7: **Dataset settings and statistics.** Right – oracle upper bound. See text for details.

tio of the temporal intersection over union (IoU) exceeds a specified threshold. We average the recall values across all language queries to obtain the average recall at K. We report R@K for  $K \in \{1,5\}$  and  $IoU \in \{0.5,0.7\}$ . Note that DiDeMo [16] has multiple annotations for each sentence corresponding to different human judgements. We account for the multiple annotations by requiring that a correct detection must overlap with at least two of the human judgements with the specified IoU, which can be satisfied for all sentences in the val and test sets. For completeness, we also report results and compare to prior work on the original criteria proposed by DiDeMo [16] at the end of this section.

Implementation details. To obtain candidate moments in a video, we need to specify the clip length, maximum number of clips in a moment, and how frequently to extract clips in a video (temporal stride). We used the same setup used in the main submission (c.f. Table 1). For convenience, Table 7 shows the settings again for the video clip length, maximum moment length, and temporal stride used for the evaluated datasets. We set the values for each dataset to maximize an oracle detector where a sequence of (non-overlapping) clips are aligned with the ground truth moments, while minimizing computational cost. We report the performance of the oracle detector in Table 7. While the oracle's returned endpoints align to clip boundaries and do not have the ability to exactly align to ground truth endpoints, we note that the oracle detector still achieves high performance. Also note that humans may not generally agree on temporal endpoints [1]. For all approaches, we evaluate the alignment

cost for every moment in a video and suppress moments that are near lower-cost moments, *i.e.*, those that have temporal IoU greater than a given value chosen empirically (DiDeMo -1.0, Charades-STA -0.6).

We evaluate the VGG features [34], along with ResNet-152 features [14]. We use the same process outlined in Section 2 (*c.f.* supplemental). We used Glove word-embedding features [27] for the words in the language query. For temporal endpoints, we normalized the start and end points relative to the video length as in MCN [16] to obtain temporal endpoint features (TEFs).

**Baselines.** We compare our CAL model to several baselines: MCN [16], CTRL [10], ACRN [24], TGN [4], TMN [23], Xu *et al.* [35]. We also compute a moment frequency prior as in Hendricks *et al.* [16] for each dataset by discretizing the range of video-length-normalized start and end points and histograming the training ground truth moments. For the DiDeMo dataset, we reached out to the authors to get their raw outputs so we could compare directly on our DiDeMo criteria; we report obtained results in Table 8.

**Results.** Table 8 shows quantitative results over the two datasets. We separate the results into four categories – frequency-prior and language-only baselines, "non-indexable" baselines, approaches that are "indexable", and "indexable" approaches that incorporate temporal endpoint features (TEFs). We list the base features that are used for each method. Finally, for MCN and our CAL model, we evaluate over all datasets.

	DiDeMo [16] (test)				Charades-STA [10] (test)			
	IoU:	=0.5	IoU=0.7		IoU=0.5		loU:	=0.7
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Frequency Prior	26.11	88.51	21.89	70.53	23.47	66.94	11.72	42.98
TEF-only (Glove,TEF)	26.90	87.85	21.65	70.33	37.59	76.70	20.93	50.72
ACRN [24] (VGG,Glove)	28.45	61.43	16.76	41.13	_	-	_	-
CTRL [10] (C3D,Skip-thought)	-	-	-	-	21.42	59.11	7.15	26.91
[35] (C3D, word2vec)	-	-	-	-	35.60	79.40	15.80	45.40
MCN [16] (VGG-mean,Glove)	27.01	64.55	16.51	46.65	_	-	-	-
Ours (VGG-mean,Glove)	34.21	69.12	24.30	51.67	-	-	-	-
MCN [16] (ResNet,Glove)	31.26	71.35	21.29	55.82	34.06	71.65	15.18	45.08
Ours (ResNet,Glove)	36.53	71.96	25.58	54.84	32.91	72.40	15.53	48.06
MCN [16] (ResNet,Glove,TEF)	40.85	90.52	30.50	76.94	44.48	83.63	24.65	56.92
Ours (ResNet,Glove,TEF)	41.79	89.92	30.90	77.19	44.90	83.24	24.37	57.15

Table 8: **Single video moment retrieval quantitative results.** We show average recall on DiDeMo [16] and Charades-STA [10] with RGB and temporal endpoint features. See text for more details.

We observe that with VGG features, our CAL model out-performs ACRN and MCN on all criteria. ResNet features further improve performance. Adding TEF further improves performance, but closes the gap between CAL and MCN across all datasets, with a slight edge toward CAL over most criteria. The TEF allows the models to leverage the strength of the frequency prior and suggests that further work is needed on improving the joint video-language representation. On Charades-STA, the frequency prior baseline out-performs CTRL on all criteria, while CAL with and without TEF out-performs both baselines. MCN and CAL achieve results on par with [35] without TEF feature or an attention mechanism. Moreover, CAL aided by TEF outperforms this baseline over all criteria with absolute gap of +3.8-11.8%. To sum up CAL is on par or out-performs the state-of-the-art for the single video moment retrieval task.

What is the effect of TEF on the datasets? As stated in Section 4.1 (*c.f.*, submission), we observe that the TEF-only baseline is a competitive baseline for each dataset. For our analysis, we train the MCN model using only the language features and TEF on all the datasets, *i.e.* the model does not see appearance features from the video. Results are shown in Table 8 ("TEF-only"). The TEF-only baseline gives a significant boost over the moment frequency prior baseline on the Charades-STA dataset, while performing similarly to the moment frequency prior on DiDeMo. Moreover, the TEF-only baseline out-performs many of the other baselines that use video appearance features in the single video moment retrieval task.

**Results with original DiDeMo criteria.** We also show quantitative results on the original DiDeMo [16] evaluation criteria for single video moment retrieval in Table 9. The

Rank@k criterion represents that among the top-k predictions is possible to find one that overlaps with a ground truth moment with an IoU of 1.0. Note that DiDeMo [16] has multiple temporal segments for each sentence corresponding to different human judgements. Instead of consolidating all the segments of a text query into a single temporal segment, [16] uses a consensus strategy that takes into account outliers in the annotations as follows,

$$\operatorname{Rank}(P, A) = \min_{A' \in \binom{A}{3}} \frac{1}{3} \sum_{a \in A'} \operatorname{Rank}(P, a), \tag{6}$$

where P corresponds to the ordered set of predictions, A is the set of all annotations associated to a given text query, and A' is the set of all triads of annotations in A. Similarly, the mIoU criterion measures the tightness of the top-1 prediction with the ground-truth moments in terms of temporal intersection over union,

$$\mathsf{mIoU}(p,A) = \max_{A' \in \binom{A}{3}} \frac{1}{3} \sum_{a \in A'} \mathsf{IoU}(p,a), \tag{7}$$

where p corresponds to the top-1 prediction in the ordered set of predictions P. The previous criteria are computed for each text query. Thus the overall performance is the average among all the queries in the subset of interest. In practice, we use the public evaluation code released by [16].

We observe that our CAL model outperforms MCN and ACRN in the most stringent criteria (mIoU and Rank@1) with VGG and Glove features. Along the same lines of the previous results, ResNet features further improve performance of MCN and CAL. TEF, moreover, provides additional performance gains, but closes the gap between CAL

Method		Validation	set		Test set	
	mIoU	Rank@1	Rank@5	mIoU	Rank@1	Rank@5
TEF-only (Glove, TEF)	25.88	18.94	71.39	25.85	19.17	69.00
TMN [23] (VGG,Glove)	30.14	18.71	72.97	-	-	-
ACRN [24] (VGG,Glove)	-	-	-	27.22	13.03	39.27
TGN [4] (VGG,Glove)	-	-	-	38.62	24.28	71.43
MCN [16] (VGG,Glove)	27.44	15.65	55.07	28.00	16.06	55.15
Ours (VGG,Glove)	31.82	18.96	53.28	30.91	18.37	50.55
MCN [16] (ResNet, Glove)	28.36	16.36	56.01	28.96	16.97	55.41
Ours (ResNet,Glove)	34.68	21.08	55.39	33.49	20.23	53.82
MCN [16] (ResNet,Glove,TEF)	38.78	25.67	79.65	37.79	25.68	77.51
Ours (ResNet,Glove,TEF)	39.54	26.24	80.20	38.46	25.99	77.96

Table 9: **Single video moment retrieval results on DiDeMo** with its original evaluation criterion. Note that TGN [4] may not have used the original DiDeMo evaluation setup (they used  $R@\{1,5\}$  IoU=1). We include their numbers for completeness. See text for more details.

and MCN. We note that the difference between both models is not statistically significant across different random initializations with the same hyper-parameters. Interestingly, TEF allows MCN and Clip Alignment with Language to leverage the strength of the frequency prior and achieve performance competitive or better in most of the cases than TMN and TGN. This result suggests exploring prior attention-based models (*e.g.*, TMN and TGN) to clarify that their additional model complexity is not indirectly learning the frequency prior of the dataset.