Learning Temporal Sentence Grounding From Narrated EgoVideos

Kevin Flanagan
Kevin.Flanagan@bristol.ac.uk
Dima Damen
Dima.Damen@bristol.ac.uk
Michael Wray
Michael.Wray@bristol.ac.uk

School of Computer Science
University of Bristol
Bristol, UK

Abstract

The onset of long-form egocentric datasets such as Ego4D and EPIC-Kitchens presents a new challenge for the task of Temporal Sentence Grounding (TSG). Compared to traditional benchmarks on which this task is evaluated, these datasets offer finer-grained sentences to ground in notably longer videos. In this paper, we develop an approach for learning to ground sentences in these datasets using only narrations and their corresponding rough narration timestamps. We propose to artificially merge clips to train for temporal grounding in a contrastive manner using text-conditioning attention. This Clip Merging (Cli$\mathcal{M}$er) approach is shown to be effective when compared with a high performing TSG method—e.g. mean R@1 improves from 3.9 to 5.7 on Ego4D and from 10.7 to 13.0 on EPIC-Kitchens. Code and data splits available from: https://github.com/keflanagan/CliMer

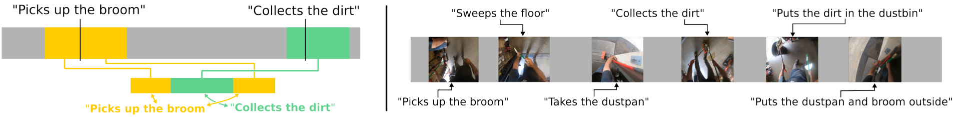


Figure 1: Left: We generate training examples from videos with rough timestamps of narrations by artificially merging clips to provide a contrastive signal. Right: At test time, Cli$\mathcal{M}$er can perform temporal grounding in long, dense videos.

1 Introduction

Suppose you have recorded a video of your birthday party using a wearable device, and you wish to find the exact moment you blew out the candles, or ripped open a present with reckless abandon. Searching through video data to retrieve the temporal extents of meaningful moments can be an arduous task. A method that can take in a sentence describing the intended moment and return that moment, as a short temporal segment, is known as Temporal Sentence Grounding [1, 14]. This problem is commonly explored within datasets with videos of a couple of minutes duration, typically grounding clips 10s of seconds long [1, 14, 21, 40].

	Total Vid Duration	Avg. Vid Duration	Avg. Mom. Duration	Annotations / Video	Total Annotations	Avg. Coverage ↓
ANet-Captions [21]	487.6h	2.0min	37.1s	4.9	72k	30.90%
Charades-STA [14]	57.1h	0.5min	8.1s	2.3	16k	27.00%
DiDeMo [1]	88.7h	0.5min	6.5s	3.9	41k	21.70%
TACoS [40]	10.1h	4.8min	27.9s	143.6	18k	9.70%
Ego4D [17]	234.9h	17.3min	2.0s	214.1	223k	**0.19%**
EPIC-Kitchens [12]	73.4h	8.9min	3.1s	134.1	67k	0.58%

Table 1: Temporal Sentence Grounding Datasets—comparative statistics. We compare previously used datasets (top) to the egocentric ones we use (bottom). Avg. Coverage showcases the ratio of the ground-truth moment duration to the video duration. Ego4D and EPIC-Kitchens have the shortest avg. moment duration with significantly lower coverage (< 1%).

In this work, we explore Temporal Sentence Grounding (TSG) for long-form egocentric video datasets where searches may correspond to segments of only a few seconds in videos that are up to an hour in length (See Table 1). Recent works in TSG have explored long-form videos [2, 19], yet still require full supervision in the form of start/end times of clips. In contrast, rough timestamps from video narrations have previously been used for learning other tasks such as recognition [35], detection [22, 31, 39], and spatio-temporal localisation [32]. In [31], Ma *et al*. found that narration timestamps roughly located at or near the relevant action require 6× less annotation effort than labelling with start/end times. The improved efficiency is crucial for event-rich datasets with long videos. This proposes the unique problem that we investigate within this paper: how do we train a successful model for TSG on long videos with only narrations and their rough timestamps as supervision.

Our proposed method, Cli$\mathcal{M}$er, artificially generates training examples from long-form narrated videos by merging clips together to form a merged segment, Figure 1 (left). The features from the merged segment can be conditioned with the narrations, thus providing hard boundaries for Cli$\mathcal{M}$er to learn from in a contrastive learning set-up. Because of this training setup, at inference time our method is able to perform TSG over long-form videos with a high density of annotated sentences, Figure 1 (right).

To summarise, (i) we propose Cli$\mathcal{M}$er for Temporal Sentence Grounding, only using narrations and their rough timestamps as supervision. Cli$\mathcal{M}$er merges clips as a supervisory signal to use for contrastive learning. (ii) We explore for the first time using Ego4D and EPIC-Kitchens for TSG providing train-test splits for this task. (iii) We show that Cli$\mathcal{M}$er outperforms the baseline VSLNet [50] on these two datasets, ablating our design decisions.

2 Related Work

TSG Datasets Table 1 shows a comparison between datasets previously used for Temporal Sentence Grounding and the two EgoVideo datasets we explore in this work, Ego4D [17] and EPIC-Kitchens [12]. Avg. Coverage (Avg. of Moment Duration/Video Duration) in particular displays the distinction between these datasets. Previous datasets aim to retrieve a moment ≥ 10% of the video length on average. Ego4D and EPIC-Kitchens have a significantly smaller coverage (0.19% and 0.58% respectively), due to their fine-grained sentences with much shorter average moment duration. Additionally, biases within previous TSG datasets ([14, 21]) have been explored in [37, 49]. Specifically, [49] finds that the evaluated methods "fail to utilize the video temporal relation or vision language interaction" and was shown to be even more catastrophic for weakly supervised approaches.

Our motivation to explore datasets with significantly smaller coverage is related to the seminal dataset of movie descriptions, MAD [42]. In MAD, movies up to 3 hours long were explored for temporal grounding. Movies offer different challenges including plot understanding and they typically contain scene boundaries. In this work, we focus on EgoVideos that are unedited with finer-grained narrations.

Fully Supervised Grounding Fully supervised approaches make use of exact start and end times for each sentence during training. Approaches are divided between proposal-based and proposal-free. Proposal-based methods [1, 3, 5, 14, 15, 20, 25, 27, 28, 38, 43, 45, 51] generate a set of candidate segments within the video and rank these for a given sentence. Proposal-free methods [6, 7, 8, 9, 18, 26, 29, 36, 41, 48, 50, 52] directly predict the start and end times. A small number of works have tackled the task of fully supervised long video grounding [2, 19]. These have taken the approach of using a secondary module to first pick out segments from the full video before applying models better suited to short video grounding within these segments.

Weakly Supervised Grounding Weakly supervised approaches [4, 10, 11, 24, 34, 44, 46, 47, 53, 54] do not have access to start and end times for each sentence during training. Instead, videos are paired with sentences which can be grounded somewhere within them. Recent approaches include CPL [54], which generates content-dependent proposals and mines hard negative samples from within the same video, and LCNet [47], which extracts a hierarchical feature representation for video and text and models the local correspondences between these. Similar to our work, [4] also creates merged videos, yet their merging approach assumes a dataset with high coverage by sampling a random clip from the video to act as the ground truth for a sentence. In our method, we merge clips based on the rough timestamp supervision and use hard negative mining to ensure a trainable signal for Ego4D and EPIC-Kitchens, which have much lower coverage. Other recent approaches [11, 46] have made use of "glance"/"point" annotations, which are similar to the rough timestamp we use, but both works make the assumption that the "glance"/"point" is within the ground truth segment which can be unrealistic given potential annotation noise, either human or automatic [33].

3 Method

We first introduce the task of Temporal Sentence Grounding (TSG) from Narration Timestamp Supervision in Sec. 3.1. Next, we provide an overview of the method in Sec. 3.2. Finally, we give details of sampling sentences and creating the merged segments in Sec. 3.3 followed by text conditioning in Sec. 3.4, as well as training losses and inference in Sec. 3.5.

3.1 Task Formulation: TSG from Narration Timestamps

Temporal Sentence Grounding is the problem of finding moments of a video that ground a sentence. Formally, given a sentence c_i and a video x[1], we wish to find the start and end times—given by t_i^s and t_i^e respectively—within x which ground the sentence c_i.

In the fully supervised setting, during both training and testing, methods use the tuple (x, c_i, t_i^s, t_i^e) to first train and later evaluate the model by comparing predicted times ($\hat{t}_i^s$ and $\hat{t}_i^e$) with the ground truth times. In this work, we explore a weaker form of supervision in which

[1] For simplicity we drop the video index as Temporal Sentence Grounding assumes oracle knowledge of the video that the sentence corresponds to.

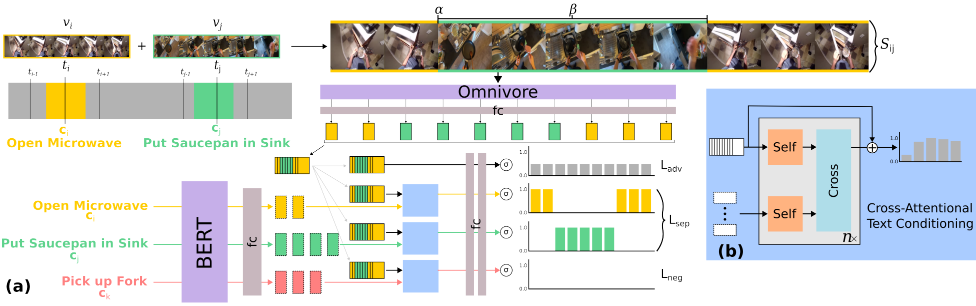


Figure 2: Overview of Cli$\mathcal{M}$er. (a) We generate a merged segment (S_{ij}) from two clips (v_i/v_j) of one long video to contrast against sentences $c_i/c_j/c_k$. The video features are conditioned using extracted text features and trained with 3 losses $L_{adv}, L_{sep}, L_{neg}$. (b) Cross-Attentional Text Conditioning uses n blocks of self- and cross-attention to condition the text.

only rough timestamps are available for each sentence during training. These represent a narration's single timestamp t_i that roughly corresponds to the sentence as labelled by a narrator. Thus, in this setting, models are trained using (x, c_i, t_i) but are still evaluated on their ability to ground the sentence to its annotated start/end times. Naturally, this leads to a more difficult training regime, yet has the benefit of requiring much less annotation time [31].

3.2 Cli$\mathcal{M}$er Overview

An overview of the approach can be seen in Figure 2, where from an untrimmed video two sentences c_i (yellow) and c_j (green) are sampled. We additionally sample a third sentence from the same video, c_k (red), as a negative sentence in order to train the model to distinguish video segments that do not contain the corresponding grounding. As discussed in the previous section, during training we only have access to single rough timestamps from the narrations as supervision, *e.g.* t_i/t_j, for sentence c_i/c_j. We first generate rough clips, v_i and v_j around t_i and t_j.

Next, we create a new input S_{ij} for training, which we term the merged segment, by merging the clips v_i and v_j. This is beneficial for two reasons: firstly, the artificial boundaries between clips act as a supervision signal; and secondly, contrasting multiple sentences ensures that the method learns to discriminate and ground individual sentences. As we artificially perform the merging, we use the merging signal as supervision for learning to ground.

We condition each sentence against S_{ij}, using cross-attentional text conditioning (Figure 2 (b)) and train the model with three losses which contrast the merged segment with the three sentences and regularise the predictions. We detail our method next.

3.3 Sampling Sentences and Merging Segments

We now explain how to sample the sentences, from a given video, and create a merged segment on which to train. We avoid neighbouring clips, by ensuring *e.g.* $|i-j| \geq 3$. Additionally, we check that all three sentences c_i, c_j, and c_k are semantically distinct, by ensuring the main action (the verb) and/or the main object (the noun) are distinct. *E.g.* given the sentence "*The person closes the bin*", the sentence "*The person opens the bin*" is considered distinct due to the main verb being different. However, "*The person shuts the trash can*" is considered indistinct due to the main verb (closes/shuts) and noun (bin/trash can) both being semantically equivalent.

Next, we segment the rough clips v_i and v_j. To compensate for the lack of temporal bounds, we use the timestamps of surrounding narrations, t_{i+1} and t_{i-1} (resp. t_{j+1} and t_{j-1}), as upper and lower bounds for the clips. Note that there are many possible ways of initialising these bounds (*e.g.* in [23]) which we explore further in Table 1 of the supplementary material.

To merge the clips, we randomly sample two variables α, β such that $0 \leq \alpha \leq 1$ is the position at which we position the first clip and $0 < \beta \leq (1-\alpha)$ is its duration. From these, we define a merging template y as a rectangular function, so that $y(t) = 1$ if $\alpha < t < \alpha + \beta$, and $y(t) = 0$ otherwise. This template merges the two clips v_i and v_j such that:

$$S_{ij}(t) = \begin{cases} v_i((t-\alpha)/\beta) & y(t) = 1 \\ v_j(t/(1-\beta)) & y(t) = 0 \quad \& \quad t < \alpha \\ v_j((t-\beta)/(1-\beta)) & y(t) = 0 \quad \& \quad t > \alpha \end{cases} \tag{1}$$

where $0 \leq t \leq 1$ represents the time along the normalised length of any clip and $v_i(t)$ represents the clip v_i at time t. The template y also defines the supervisory signal (y_i and y_j) for both sentences. For the sentence c_i, $y_i = y$ is a binary vector set to 1 for all locations where S_{ij} contains parts of the clip v_i and 0 otherwise. Analogously $y_j = \mathbb{1}_{|y|} - y$, where $\mathbb{1}_{|y|}$ is a vector of 1s of length $|y|$.

Note that the above definition assumes continuous time t. Instead, we pre-extract visual video features F at a fixed temporal rate from the video. This rate is then used to discretise the merging operation noted above.

3.4 Text Conditioning

We condition the visual features F_{ij} of the merged segment S_{ij} on the sentences c_i and c_j, by matching to the corresponding sentence and contrasting from the other sentences. A projection layer, g, is applied to extracted word features to project the sentence embeddings into the same space as the visual feature embeddings, resulting in final text features $\hat{c}_i$ with $L \times d$ dimensions where L is the length of the sentence:

$$\hat{c}_i = [g(f(w_{i,1})), \ldots, g(f(w_{i,L}))] \tag{2}$$

where $w_{i,l}$ represents the lth word in the sentence c_i and f represents the text feature extractor. To condition the visual features on $\hat{c}_i$, we propose a combination of both self-attention and cross-attention within successive transformer layers to learn correlations both within and between the modalities, see Figure 2(b). We follow a similar approach to [30] where first visual and text features undergo self-attention, before being concatenated and passed through a transformer layer using cross-attention:

$$P = \mathcal{A}^s_v(F_{ij}) \quad ; \qquad Q_i = \mathcal{A}^s_t(\hat{c}_i) \tag{3}$$

where $P = (p_1, \ldots, p_M)$ and $Q_i = (q_{i,1}, \ldots, q_{i,L})$ are the outputs of the self-attention layers and A^s represents a self-attention layer.

$$H_i^l = \mathcal{A}^c(H_i) \tag{4}$$

where $H_i = [p_1, \ldots, p_m, \ldots, p_M, q_{i,1}, \ldots, q_{i,l}, \ldots, q_{i,L}]$ (the concatenated video and textual outputs from the previous layer) and $\mathcal{A}^C$ is a cross-attention transformer layer in which keys and values are passed as inputs to the opposing modality. The combination of self-attention and cross-attention layers, *i.e.* equations 3 and 4, are repeated n times. We then include a residual connection from the input visual features. This is followed by two fully connected layers and a sigmoid function that estimates the probability of each visual input feature being the grounding for the sentence c_i:

$$\hat{Y}_i = \mathcal{M}(F_{ij}, c_i) = \sigma\big(e(H_i^l) + F_{ij}\big) \tag{5}$$

where $\mathcal{M}$ defines the model Cli$\mathcal{M}$er, e represents two fully connected layers and $\hat{Y}_i$ is the vector of estimated probabilities that the visual features ground the sentence c_i.

3.5 Training and Inference

During training, we randomly sample clips for merging within each video in the dataset. We restrict the size of the merged clip S_{ij} to a fixed size of M features for batching purposes. We next describe the losses used to train Cli$\mathcal{M}$er.

The main contrasting loss, denoted collectively as $\mathcal{L}_{sep}$, is applied between the model's predicted probabilities $\hat{Y}$ and the merged clips supervisory signals y_i, y_j.

$$\mathcal{L}_{sep} = (1-\beta)\cdot \text{BCE}(\hat{Y}_i, y_i) + \beta \cdot \text{BCE}(\hat{Y}_j, y_j) \tag{6}$$

where BCE is the Binary Cross Entropy Loss, balanced by the proportion of the other sentence in the merged segment S_{ij} using β from Sec. 3.3 and $\hat{Y}_i$ is the vector of estimated grounding probabilities from Eq 5.

$\mathcal{L}_{neg}$ is applied to the negative sentence c_k to prevent the method overfitting to the sampled boundaries within the merged segment S_{ij}. This is then applied across video features that have been conditioned on c_k where resulting probabilities should be 0:

$$\mathcal{L}_{neg} = \text{BCE}(\mathcal{M}(S_{ij}, c_k), 0 \cdot \mathbb{1}_{|S_{ij}|}) \tag{7}$$

A further regularisation loss, which we term the adversarial loss $\mathcal{L}_{adv}$, is used on the non-conditioned visual features (i.e. without any text conditioning). The adversarial loss enforces that Cli$\mathcal{M}$er uses both the textual query and the visual frames to retrieve moments. This follows related works which find that temporal sentence grounding models can perform well with no visual inputs [37, 49]. Using this loss, visual features with no text are conditioned to produce a vector with maximal uncertainty (i.e. $0.5 \cdot \mathbb{1}_{|S_{ij}|}$)).

$$\mathcal{L}_{adv} = \text{BCE}(\mathcal{M}(S_{ij}), 0.5 \cdot \mathbb{1}_{|S_{ij}|}) \tag{8}$$

The final loss to train the model is given as:

$$\mathcal{L} = \mathcal{L}_{sep} + \lambda_{neg} \cdot \mathcal{L}_{neg} + \lambda_{adv} \cdot \mathcal{L}_{adv} \tag{9}$$

where $\lambda_{\{neg,adv\}}$ are weights used to balance the different losses.

At inference, the model considers a single sentence and is applied to the full untrimmed length of a single video, no longer restricting the number of features. The model outputs the probability of grounding the sentence across the full extent of the video, not merged segments. To predict the grounding, we normalise the predictions between 0 and 1 and consider a threshold ε to convert predictions into a grounding such that $\forall \hat{t} : \hat{t}_i^s \le \hat{t} \le \hat{t}_i^e \iff \hat{Y}_i(\hat{t}) > \varepsilon$. Predicted groundings $(\hat{t}_i^s, \hat{t}_i^e)$ are ranked by each one's maximum prediction value and returned as a ranked list. We then calculate the temporal IoU of the highest ranked grounding against the ground truth (t_i^s, t_i^e) to report performance.

4 Experiments

Metrics We use the standard metric for TSG, Recall@K (R@K) with Intersection Over Union (IoU) at threshold θ (IoU=θ). As in [50], we evaluate $\theta \in \{0.1, 0.3, 0.5\}$ and K=1. We also report Mean Recall@1 (mR@1) metric introduced in [2] as the average across θ.

Baselines We select VSLNet [50] and its implementation in [17] as our baseline. For fair comparison, we use the same Omnivore visual features and BERT text features for training this baseline as we use for Cli$\mathcal{M}$er. This is one of two baselines used for the NLQ task in Ego4D and has shown high performance on grounding tasks in previous datasets. We train VSLNet using artificially generated start and end times based on the rough timestamps. We also include a random baseline which samples a random segment of any length from the video to demonstrate the challenge of the test sets.

Datasets We use two datasets: Ego4D [17] and EPIC-Kitchens-100 [12]. Both datasets have published a mapping from open vocabulary to closed vocabulary, using manual and automatic clustering. For each dataset, we consider all verbs/nouns within the same 'closed vocabulary' cluster to be semantically equivalent (i.e. pan and saucepan are both contained within the pan cluster so are considered semantically equivalent). These are used to measure the semantic equivalence of sentences.

Ego4D: We consider a subset of videos from [17] covering various scenarios: {*cooking, carpenter, scooter mechanic, car mechanic, gardener, farmer*}. Annotations from the hand-object interaction challenge are used as ground truth in the val/test splits, as these videos contain human-labelled start and end times with corresponding sentences. Due to the nature of the Ego4D videos, there are often a number of repeated or semantically similar sentences within the videos. Currently, TSG approaches assume there is only one positive grounding moment for each sentence. To align with the grounding task, any repeated or semantically indistinct sentences were excluded from the val/test sets. This resulted in a dataset with 462/96/223 videos and 197k/1.9k/4.4k sentences in train/val/test. When using the val/test sets, only full videos are used, no clip merging takes place.

EPIC-Kitchens-100: We use all 495 videos in the EPIC-Kitchens-100 train set to make up our train/val/test splits. We use the narration annotations for training and the manually labelled start-end times for evaluation. We similarly remove semantically equivalent sentences, and the resulting dataset has 332/50/110 videos and 45k/1.8k/3.8k sentences in train/val/test respectively.

Features BERT [13] is used to generate features for the sentences, trained on BookCorpus [55] and English Wikipedia, generating 768-length features for each word. Visual features are obtained from the video head of the Omnivore [16] model (SwinL for Ego4D, SwinB for EPIC-Kitchens) trained on Kinetics-400 and ImageNet-1K jointly and are of length 1536 for Ego4D and 1024 for EPIC-Kitchens. Each visual feature represents 32 frames of the video, and these are generated with a stride of 16 frames. As Ego4D videos have an FPS of 30, the feature rate is ~1.87 per second. EPIC-Kitchens videos are down-sampled to 30 FPS to match this.

Model The textual features and visual features are first projected to the common size of 2048 dimensions. For Ego4D, the number of blocks of self and cross-attention before the final self-attention block is set to $n = 2$. For EPIC-Kitchens, we switch the text attenuation module with a simpler one. In this case, the visual features undergo two self-attention layers before a Hadamard product is used to attenuate them with mean-pooled text features instead

	IoU=0.1	IoU=0.3	IoU=0.5	mR
Random Baseline	0.47	0.09	0.02	0.19
VSLNet [50]	7.32	3.16	1.35	3.94
CliMer	**9.68**	**5.03**	**2.24**	**5.65**

Table 2: Results for R@1 against baselines on Ego4D [17]

	IoU=0.1	IoU=0.3	IoU=0.5	mR
Random Baseline	0.78	0.13	0.03	0.31
VSLNet [50]	19.32	8.76	3.90	10.66
CliMer	**22.20**	**11.57**	**5.25**	**13.01**

Table 3: Results for R@1 against baselines on EPIC-Kitchens [12]

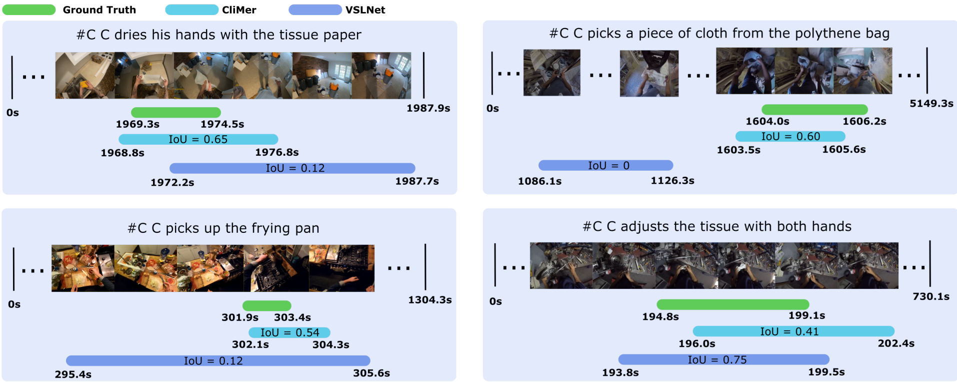


Figure 3: Examples from Ego4D comparing CliMer and VSLNet with ground truth.

of a cross attention layer. We ablate this design choice in Table 7 and show it is more suited to the simpler sentences within EPIC-Kitchens.

All attention layers contain 6 heads. The number of visual features during training is set to $M = 20$. We use the Adam optimiser with a learning rate of 10^{-5} and batch size of 32. We use a dropout value of 0.3 and a value of 0.5 for the text feature projection. The model is trained for 100 epochs and the epoch with the best validation result is selected to report performance on the test set (*i.e.* we do not optimise on the test set). We set the weights $\lambda_{neg} = \lambda_{adv} = 1$ and ablate these losses. The prediction threshold, ε is empirically chosen as 0.8 for Ego4D and 0.6 for EPIC-Kitchens (see Figure 4).

4.1 Results

We first show results of CliMer on Ego4D in Table 2. The random baseline shows the challenge of this dataset. Although VSLNet has been adapted for long videos as described in [17], CliMer outperforms it across all metrics.

Table 3 includes the results on EPIC-Kitchens. Random performance continues to be low for EPIC-Kitchens, and both methods perform considerably better than it. Once again, CliMer achieves higher performance across all metrics. We believe the contrastive learning enabled by our clip merging method allows the model to learn to locate corresponding video segments more effectively when given an input sentence.

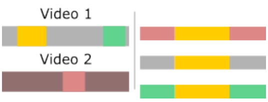
Figure 3 shows qualitative examples comparing CliMer with VSLNet [50]. VSLNet often over-predicts start/end times leading to poor IoU scores when compared to the ground truth. An extreme example of this can be seen in the top right predicting a ~40s clip for a ~2s ground truth segment. We find that CliMer is able to match the ground truth more closely, despite the fine-grained nature of the sentences. In the bottom right example, CliMer continues to predict the part of the video where the tissue is visible but is no longer being adjusted.

c_j	$\mathcal{L}_{neg}$	$\mathcal{L}_{adv}$	IoU=0.1	IoU=0.3	IoU=0.5	mR
✓	✓	-	9.57	4.46	2.13	5.39
✓	-	✓	8.61	3.62	1.58	4.60
-	✓	✓	8.88	4.35	1.90	5.04
✓	✓	✓	**9.68**	**5.03**	**2.24**	**5.65**

Table 4: Impact of removing each loss during training on Cli$\mathcal{M}$er.

λ_{neg}	λ_{adv}	IoU=0.1	IoU=0.3	IoU=0.5	mR
0.5	1	8.93	4.12	1.95	5.00
1	0.5	8.88	4.49	2.08	5.15
1	0	9.57	4.46	2.13	5.39
0	1	8.61	3.62	1.58	4.60
1	1	**9.68**	**5.03**	**2.24**	**5.65**

Table 5: Effect of changing the loss weights on Cli$\mathcal{M}$er.



Merged Segment	Hard Negatives	IoU=0.1	IoU=0.3	IoU=0.5	mR
✓	-	7.28	3.09	1.08	3.82
-	✓	7.23	4.42	2.20	4.62
✓	✓	**9.68**	**5.03**	**2.24**	**5.65**

Table 6: Performance on Ego4D with ablations on merging clips to produce the supervision signal and sampling clips from the same video as hard negatives. (Left inset) We show how each row corresponds to the merged segment creation.

4.2 Ablation Studies

Losses In Table 4 we showcase that each loss contributes to the model's performance, with performance decreasing when any are omitted. Specifically, removing the loss of the second sentence (c_j) from $\mathcal{L}_{sep}$ removes the contrastive learning aspect of the model, leading to performance decreases, albeit still being able to achieve reasonable performance via grounding the single sentence only. Removing $\mathcal{L}_{adv}$ has the least impact, being a regularisation loss. Removing $\mathcal{L}_{neg}$ leads to the largest performance drop, as the model otherwise assumes that all sentences can be grounded resulting in an increase of false positives.

We also adjust the weights of the negative and adversarial losses (λ_{neg} and λ_{adv} from Eq. 9) in the loss function to examine the effect this has on model performance for Ego4D. The results in Table 5 show that decreasing either weights results in reduced performance across all metrics. This demonstrates the importance of each loss having equal weighting to the main separation loss, providing complementary signals during training.

Clip Merging We investigate the impact of merging segments. We replace merging with using a single clip with surrounding background. The background is selected to make up between 25% and 75% of the video at random, retaining the value of $M = 20$ for the number of features. The losses are retained as before however the weighting of loss $\mathcal{L}_{sep}$ is doubled to compensate for the lack of contribution from c_j.

From results in Table 6 (row 2) without merging, performance drops. The decrease at IoU=0.1 in particular shows that the model without merging struggles to learn the general alignment between sentences and video, likely due to the lack of contrastive learning and the decreased visual diversity found in continuous segments compared to merged segments. The model trained without merging frequently misses the ground truth moment entirely. The results show that clip merging increases robustness to changes in clip positions and lengths in full videos.

	Ego4D				EPIC-Kitchens			
	IoU=0.1	IoU=0.3	IoU=0.5	mR	IoU=0.1	IoU=0.3	IoU=0.5	mR
Hadamard Prod. Cross Att.	9.29	4.30	2.04	5.21	**22.20**	**11.57**	5.25	**13.01**
Learned Cross Att.	**9.68**	**5.03**	**2.24**	**5.65**	19.79	10.58	**5.46**	11.94

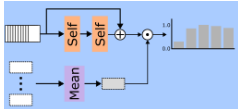


Table 7: Ablation comparing the type of text attenuation used: Cross-attention as in Figure 2(b) or using a Hadamard Product (shown right).

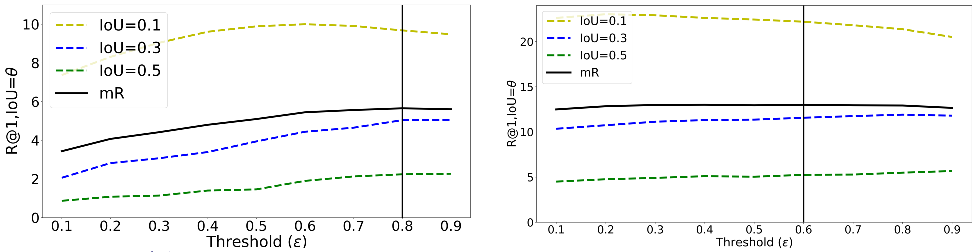


Figure 4: Cli$\mathcal{M}$er performance as the prediction threshold, ε, is varied on Ego4D (Left) and EPIC-Kitchens (Right). The vertical lines indicate our thresholds of 0.8 and 0.6.

Hard Negative Sampling Table 6 also compares clips sampled as hard negatives from the same video against when they are sampled from random videos. A significant performance drop is shown in the latter case. Hard negatives force the model to distinguish between the actions occurring within the videos as the visual environments of the two clips are similar.

Text Attenuation Table 7 evaluates the choice of text conditioning module, comparing using Cross Attention (see Section 3.4) with the Hadamard product. The simpler module which we use for EPIC-Kitchens works best due to the smaller dataset and shorter sentences (avg. 3.0 words) vs. Ego4D (avg. 8.4 words).

Prediction Threshold Figure 4 displays the effect of varying the value of the prediction threshold ε. The threshold ε has more impact on Ego4D, however the figure demonstrates that the model is reasonably robust to the choice of $\varepsilon > 0.5$.

5 Conclusion

In this work we have explored Temporal Sentence Grounding for long-form egocentric datasets Ego4D and EPIC-Kitchens. To overcome the need for full supervision, we propose to merge clips from the same video from rough narration timestamps. Our method, Cli$\mathcal{M}$er, trains using contrastive learning with text-conditioning over sentences. Results demonstrate the ability of Cli$\mathcal{M}$er to tackle long videos at test time, grounding fine-grained sentences in egocentric videos which can be over an hour in length.

Acknowledgments. K Flanagan is supported by UKRI (Grant ref EP/S022937/1) CDT in Interactive AI & Qinetiq Ltd via studentship CON11954. D Damen is supported by EPSRC Fellowship UMPIRE (EP/T004991/1) & EPSRC Program Grant Visual AI (EP/T028572/1).

References

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

[2] Wayner Barrios, Mattia Soldan, Fabian Caba Heilbron, Alberto M. Ceballos-Arroyo, and Bernard Ghanem. Localizing moments in long video via multimodal guidance. *ICCV*, 2023.

[3] Meng Cao, Long Chen, Mike Zheng Shou, Can Zhang, and Yuexian Zou. On pursuit of designing multi-modal transformer for video grounding. In *EMNLP*, 2021.

[4] Jiaming Chen, Weixin Luo, Wei Zhang, and Lin Ma. Explore inter-contrast between videos via composition for weakly supervised temporal sentence grounding. In *AAAI*, 2022.

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

[6] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. Rethinking the bottom-up framework for query-based video localization. In *AAAI*, 2020.

[7] Shaoxiang Chen and Yu-Gang Jiang. Hierarchical visual-textual graph for temporal activity localization via language. In *ECCV*, 2020.

[8] Shaoxiang Chen, Wenhao Jiang, Wei Liu, and Yu-Gang Jiang. Learning modality interaction for temporal sentence localization and event captioning in videos. In *ECCV*, 2020.

[9] Yi-Wen Chen, Yi-Hsuan Tsai, and Ming-Hsuan Yang. End-to-end multi-modal video temporal grounding. *NeurIPS*, 2021.

[10] Zhenfang Chen, Lin Ma, Wenhan Luo, Peng Tang, and Kwan-Yee K Wong. Look closer to ground better: Weakly-supervised temporal grounding of sentence in video. *CoRR abs/2001.09308*, 2020.

[11] Ran Cui, Tianwen Qian, Pai Peng, Elena Daskalaki, Jingjing Chen, Xiaowei Guo, Huyang Sun, and Yu-Gang Jiang. Video moment retrieval from text queries via single frame annotation. In *SIGIR*, 2022.

[12] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *IJCV*, 2022.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

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

[15] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. Mac: Mining activity concepts for language-based temporal localization. In *WACV*, 2019.

[16] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022.

[17] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *CVPR*, 2022.

[18] Jiachang Hao, Haifeng Sun, Pengfei Ren, Jingyu Wang, Qi Qi, and Jianxin Liao. Query-aware video encoder for video moment retrieval. *Neurocomputing*, 2022.

[19] Zhijian Hou, Wanjun Zhong, Lei Ji, Difei Gao, Kun Yan, Wing-Kwong Chan, Chong-Wah Ngo, Zheng Shou, and Nan Duan. Cone: An efficient coarse-to-fine alignment framework for long video temporal grounding. *ECCVW*, 2022.

[20] Bin Jiang, Xin Huang, Chao Yang, and Junsong Yuan. Cross-modal video moment retrieval with spatial and language-temporal attention. In *ICMR*, pages 217–225, 2019.

[21] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017.

[22] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. In *CVPR*, 2021.

[23] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *NeurIPS*, 2022.

[24] Zhijie Lin, Zhou Zhao, Zhu Zhang, Qi Wang, and Huasheng Liu. Weakly-supervised video moment retrieval via semantic completion network. In *AAAI*, 2020.

[25] Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. Context-aware biaffine localizing network for temporal sentence grounding. In *CVPR*, 2021.

[26] Daizong Liu, Xiaoye Qu, Xing Di, Yu Cheng, Zichuan Xu, and Pan Zhou. Memory-guided semantic learning network for temporal sentence grounding. In *AAAI*, 2022.

[27] Daizong Liu, Xiaoye Qu, Pan Zhou, and Yang Liu. Exploring motion and appearance information for temporal sentence grounding. In *AAAI*, 2022.

[28] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. In *MM*, 2018.

[29] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. Debug: A dense bottom-up grounding approach for natural language video localization. In *EMNLP-IJCNLP*, 2019.

[30] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 2019.

[31] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *ECCV*, 2020.

[32] Pascal Mettes, Jan C Van Gemert, and Cees GM Snoek. Spot on: Action localization from pointly-supervised proposals. In *ECCV*, 2016.

[33] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[34] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, 2019.

[35] Davide Moltisanti, Sanja Fidler, and Dima Damen. Action recognition from single timestamp supervision in untrimmed videos. In *CVPR*, 2019.

[36] Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, Hao Zhang, and Wei Lu. Interventional video grounding with dual contrastive learning. In *CVPR*, 2021.

[37] Mayu Otani et al. Uncovering hidden challenges in query-based video moment retrieval. In *BMVC*, 2020.

[38] Xiaoye Qu, Pengwei Tang, Zhikang Zou, Yu Cheng, Jianfeng Dong, Pan Zhou, and Zichuan Xu. Fine-grained iterative attention network for temporal language localization in videos. In *MM*, 2020.

[39] Rahul Rahaman, Dipika Singhania, Alexandre Thiery, and Angela Yao. A generalized and robust framework for timestamp supervision in temporal action segmentation. In *ECCV*, 2022.

[40] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *TACL*, 2013.

[41] Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *WACV*, 2020.

[42] 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 *CVPR*, 2022.

[43] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *AAAI*, 2020.

[44] Yuechen Wang, Jiajun Deng, Wengang Zhou, and Houqiang Li. Weakly supervised temporal adjacent network for language grounding. *TMM*, 2021.

[45] 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 *AAAI*, 2019.

[46] Zhe Xu, Kun Wei, Xu Yang, and Cheng Deng. Point-supervised video temporal grounding. *TMM*, 2022.

[47] Wenfei Yang, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Local correspondence network for weakly supervised temporal sentence grounding. *TIP*, 2021.

[48] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*, 2019.

[49] Yitian Yuan, Xiaohan Lan, Xin Wang, Long Chen, Zhi Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In *HuMA*, 2021.

[50] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Natural language video localization: A revisit in span-based question answering framework. *TPAMI*, 2021.

[51] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, 2019.

[52] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *ECCV*, 2020.

[53] Minghang Zheng, Yanjie Huang, Qingchao Chen, and Yang Liu. Weakly supervised video moment localization with contrastive negative sample mining. In *AAAI*, 2022.

[54] Minghang Zheng, Yanjie Huang, Qingchao Chen, Yuxin Peng, and Yang Liu. Weakly supervised temporal sentence grounding with gaussian-based contrastive proposal learning. In *ICCV*, 2022.

[55] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, pages 19–27, 2015.