# COOT: Cooperative Hierarchical Transformer for Video-Text Representation Learning

**Simon Ging**[1*], **Mohammadreza Zolfaghari**[1*], **Hamed Pirsiavash**[2], **Thomas Brox**[1]
[1]University of Freiburg, [2]University of Maryland Baltimore County
[1]{gings, zolfagha, brox}@cs.uni-freiburg.de, [2] hpirsiav@umbc.edu

## Abstract

Many real-world video-text tasks involve different levels of granularity, such as frames and words, clip and sentences or videos and paragraphs, each with distinct semantics. In this paper, we propose a **Coo**perative hierarchical **T**ransformer (**COOT**) to leverage this hierarchy information and model the interactions between different levels of granularity and different modalities. The method consists of three major components: an attention-aware feature aggregation layer, which leverages the local temporal context (intra-level, e.g., within a clip), a contextual transformer to learn the interactions between low-level and high-level semantics (inter-level, e.g. clip-video, sentence-paragraph), and a cross-modal cycle-consistency loss to connect video and text. The resulting method compares favorably to the state of the art on several benchmarks while having few parameters. All code is available open-source at https://github.com/gingsi/coot-videotext

## 1 Introduction

Representation learning based on both vision and language has many potential benefits: visual grounding[1–4]; visual learning with a more natural, almost self-supervised annotation process; and direct applicability to cross-modal tasks, such as video retrieval by text[5–9], video summarization [10], and automated indexing. This research direction has recently boomed [8, 11–17] also due to the success of self-attention in text analysis [18, 19] with its almost immediate applicability in the cross-modal context. Many different research foci are currently developing in this area, where some are concerned with large-scale pretraining to leverage the abundant data available [8, 11, 16, 17] to learn a joint embedding space, and others to bring in more explicit structure [20–22] or new losses [17, 23] into the learning process.

In this paper, we focus on long-range temporal dependencies and propose a hierarchical model that can exploit long-range temporal context both in videos and text when learning the joint cross-modal embedding. For instance, the action of “making tea” involves boiling water, pouring it into a cup, and then adding a tea bag. This action can take a long time and may have lots of details that distinguish a particular style of making tea from other styles. To capture the whole temporal context, we leverage the idea of a hierarchical model with losses that enforce the interaction within and between different hierarchy levels. The idea of such a hierarchy is generic and has been explored by several works [21, 22, 24] in the context of video-text learning. In addition, we use alignment losses from Zhang et al. [21] and extend our baseline model with a new feature aggregation method for the intra-level interactions between features and a new transformer-based module for inter-level interactions (between local and global semantics). We consider three different levels of hierarchy: frame/word, clip/sentence and video/paragraph, visualized by the three blocks in Figure 1.

[*] Equal contribution

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

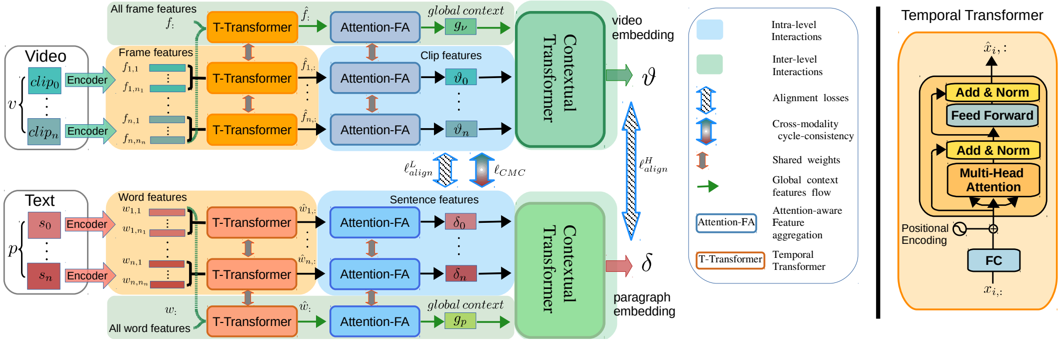


Figure 1: **Overview of COOT model** (best viewed in color). The model consist of two branches: one for video input (top) and one for text input (bottom). Given a video and a corresponding text, we encode them to frame-level/word-level features. Features belonging to each segment (clip/sentence) are fed to a standard temporal transformer (T-Transformer) followed by the proposed feature aggregation module (Attention-FA) to obtain clip/sentence-level features. Finally, a new contextual transformer produces the final video/paragraph embedding based on interactions between local context (clip/sentence features) and global context (all frames/words features). $\ell^L_{align}$, $\ell^H_{align}$, $\ell^g_{align}$ and $\ell_{CMC}$ enforce the model to align the representations at different levels.

To model *intra-level cooperation*, we introduce an attention-aware feature aggregation layer to focus on temporal interactions between low-level entities (Figure 1-Attention-FA).

This component replaces traditional sequence representation aggregation methods in transformers such as using a `[CLS]` token [11, 14, 15, 19] or mean pooling [25] with an attention-aware fusion. It leverages temporal context to encourage important entities to contribute more to the final representation of a sequence of frames or words.

For the *inter-level cooperation*, we introduce a contextual attention module, which enforces the network to highlight semantics relevant to the general context of the video and to suppress the irrelevant semantics. This is done by modeling the interaction between low-level (clips-sentences) and high-level entities (global contexts), as shown in Figure 1-green region.

In addition to this architectural contributions, we introduce a new *cross-modal cycle-consistency loss* to enforce interaction between modalities and encourage the semantic alignment between them in the learned common space. We show that enforcing two domains to produce consistent representations leads to substantially improved semantic alignment.

In summary, this paper contributes:

- a hierarchical transformer architecture with a new attention-aware feature aggregation layer and a new contextual attention module;
- a cross-modal cycle-consistency loss that encourages semantic alignment between vision and text features in the joint embedding space;
- state-of-the-art results on video-text retrieval.

## 2 Cooperative Hierarchical Transformer

Videos and text descriptions naturally involve different levels of granularity. Every paragraph contains multiple sentences, and each sentence is composed of several words. Similarly, videos have a hierarchical semantic structure, even if it is not as exactly defined as for text. Figure 1 illustrates the overview of the COOT model which consists of three levels: *1)* A temporal transformer that captures the relationships between frame/word features, *2)* attention-aware feature aggregation to produce clip/sentence features (Section 2.2) and *3)* a contextual transformer to produce final video and text embeddings (Sec. 2.3). We use alignment losses from Zhang et al. [21] to align representations at different granularity levels. In addition, we introduce a new cross-model cycle-consistency loss to connect video and text (Sec. 3). In this section, we briefly summarize the alignment losses from Zhang et al. [21] and the standard transformer.

## 2.1 Preliminaries

**Semantic Alignment Losses.** For the video-text alignment, Zhang et al. [21] leverage a contrastive loss to enforce the positive samples to stay in a close neighborhood and negative samples far apart [26–29]. Assuming the positive pair $\mathcal{P} = (x, y)$, two negative pairs $(x, y')$ and $(x', y)$ expressed as $\mathcal{N} = \{(x, y'), (x', y)\}$, and a margin $\alpha$, they define the following loss:

$$L(\mathcal{P}, \mathcal{N}, \alpha) = max(0, \alpha + D(x, y) - D(x', y)) + max(0, \alpha + D(x, y) - D(x, y')) \tag{1}$$

where $D(x, y) = 1 - x^{\intercal} y / (\|x\| \|y\|)$ is the cosine distance of two vectors.

To align representations at clip-sentence ($\vartheta_i^k$, $\delta_i^k$), video-paragraph ($\vartheta^k$, $\delta^k$) and global context ($g_v$, $g_p$) levels, Zhang et al. [21] use the following losses:

$$\begin{aligned}
\ell_{align}^{L} &= \sum_{k \in \mathcal{D}, i, k' \neq k, i' \neq i} L((\vartheta_i^k, \delta_i^k), \{(\vartheta_i^k, \delta_{i'}^{k'}), (\vartheta_{i'}^{k'}, \delta_i^k)\}, \beta) \\
\ell_{align}^{H} &= \sum_{k \in \mathcal{D}, k' \neq k} L((\vartheta^k, \delta^k), \{(\vartheta^k, \delta^{k'}), (\vartheta^{k'}, \delta^k)\}, \alpha) \\
\ell_{align}^{g} &= \sum_{k \in \mathcal{D}, k' \neq k} L((g_v^k, g_p^k), \{(g_v^k, g_p^{k'}), (g_v^{k'}, g_p^k)\}, \alpha_g)
\end{aligned} \tag{2}$$

Here, $\vartheta_i^k$ denotes the embedding for the $i$-th clip of the $k$-th video and similarly $\delta_i^k$ is the embedding of the $i$-th sentence of the $k$-th paragraph. $\alpha$, $\alpha_g$ and $\beta$ are constant margins, and $\mathcal{D}$ is a dataset of videos with corresponding text descriptions. Zhang et al. [21] employed an additional loss to model the clustering of low-level and high-level semantics in the joint embedding space:

$$\begin{aligned}
\ell_{cluster} = &\sum_{k \in \mathcal{D}, i, k' \neq k, i' \neq i} L((1, 1), \{(\vartheta_i^k, \vartheta_{i'}^{k'}), (\delta_{i'}^{k'}, \delta_i^k)\}, \gamma) \\
&\qquad + \sum_{k \in \mathcal{D}, k' \neq k} L((1, 1), \{(\vartheta^k, \vartheta^{k'}), (\delta^{k'}, \delta^k)\}, \eta)
\end{aligned} \tag{3}$$

where $\gamma$ and $\eta$ both are constant margins. The $(1, 1)$ pairs denote that positive samples are not changed. In short, the goal of this loss is to push apart embeddings for negative samples.

**Note.** Due to the symmetrical design of the video and text branches in our model, from now on, we explain only the video branch. For simplicity, we assume a single head in transformer formulations. All transformers use residual connections.

**Temporal Transformer.** We use standard attention-blocks [18] to learn frame and word representations, as shown in Fig 1-Right. We learn two temporal transformers (T-Transformer); one for the video branch and another one for the text branch. Both have the same architecture. All T-Transformers in each branch share their weights. This module draws the relationship between temporal features and yields improved representations as output. Given a video $v^k$, we first encode all its frames to obtain the frame-level features $\{f_{i,:}^k\}_{i=1}^n$, where $f_{i,:}^k$ are all frame-level features of the $i$-th clip for video $v^k$ (orange parts in Figure 1). We also consider all frame features ($f_:^k$) of a video as extra input for the global context computation (green parts in Figure 1). This yields $\{\hat{f}_{i,:}^k\}_{i=1}^n$ and $\hat{f}_:^k$.

## 2.2 Intra-Level Cooperation

Standard feature fusion methods consider each feature independently by average pooling or max pooling. Hence, they miss the relationship between features to highlight the relevant features. Recent transformers use a `[CLS]` token [11, 14, 15, 19] or average pooling [25] to obtain the aggregated features. For example, when a person is cooking, objects on the table are more relevant than objects on the wall or in the background. Therefore, we need to attend to specific features depending on the context. There have been some attempts in other domains to design a context-aware feature fusion method [30–33]. However, we introduce an attention-aware feature aggregation module (Attention-FA in Fig. 1) for video-text transformers.

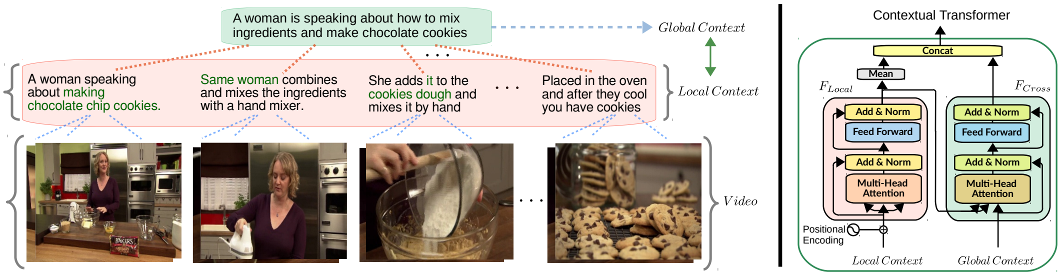

A woman is speaking about how to mix ingredients and make chocolate cookies
Global Context
A woman speaking about making chocolate chip cookies.
Same woman combines and mixes the ingredients with a hand mixer.
She adds it to the cookies dough and mixes it by hand
Placed in the oven and after they cool you have cookies
Local Context
Video
Contextual Transformer
Concat
Mean
$F_{Local}$
$F_{Cross}$
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Positional Encoding
Local Context
Global Context

Figure 2: **Contextual Transformer (CoT).** This module (right) encourages the model to optimize the representations with respect to interactions between local and global context. In the third sentence, to know the type of dough (*cookie*) the model should have information about the general context of the video (*making chocolate cookies*). Likewise, in the second sentence, to know that she is the *"same woman"*, the model must be aware of the person's identity throughout the video.

Suppose we have a sequence with $T$ feature vectors, denoted by $X = \{x_1, \ldots, x_T\}$ (e.g. $\hat{f}^k_{i,:} = \{\hat{f}^k_{i,1}, \ldots, \hat{f}^k_{i,T}\}$). We set **key** $K = X$ and utilize two learnable transformation weights $W_2$ and $W_1$ together with two biases $b_1$ and $b_2$. The attention matrix $A$ is computed as:

$$A = \text{softmax}(W_2 Q + b_2)^T, \qquad Q = GELU(W_1 K^T + b_1),\ K = X \tag{4}$$

We compute the final feature as $\hat{x} = \sum_{i=1}^{T} a_i \odot x_i$, where $\odot$ denotes element-wise multiplication and $a_i$ is the $i$-th attention vector of $A$ for the $i$-th feature. This module differs from attention [18] in two aspects: (1) we use only two learnable weights for query ($Q$) and key ($K$) and then aggregate the values based on calculated scores; (2) the query equals to transformed keys (K) and then we apply the activation function GELU [34, 35]. We feed $\{\hat{f}^k_{i,:}\}_{i=1}^n$ and $\hat{f}^k_{:}$ to this component and obtain the clip-level ($\{\vartheta^k_i\}_{i=1}^n$) features and the global context for the video ($g_\nu$).

## 2.3 Inter-Level Cooperation

By modeling the interactions between local and global context, the network learns to highlight semantics relevant to the general context of the video and to suppress the irrelevant ones: interactions between clip embeddings and the general context of the video; interactions between sentence embeddings and the general context of the text. As shown in Figure 2-Left, without knowing the global context, just from observing the frame in the third clip, there is no information about what type of *"dough"* is involved. Also the *"same woman"* in the second clip could not be related to the woman seen in the first clip.

Thus, we propose a Contextual Transformer (CoT) in Figure 2-Right to model the interactions between low-level and high-level semantics. More formally, we build the Contextual Transformer with two modules $F_{Local}$ and $F_{Global}$. We append the positional embedding to the inputs of $F_{Local}$. The goal of $F_{Local}$ is to model the short-term interactions between low-level semantics ($\{\vartheta^k_i\}_{i=1}^n$), whereas $F_{Global}$ models the interactions between local and global context ($g_\nu$) to highlight the important semantics.

Given local representations $\{\vartheta^k_i\}_{i=1}^n \in \mathbb{R}^{n \times d}$, where $n$ is the number of clips and $d$ indicates the feature dimension, $F_{Local}$ applies multi-head attention followed by a feed-forward layer and a normalization layer on top of both layers and produces embeddings $\{h_i\}_{i=1}^n$.

We compute **key (K)-value(V)** pairs based on these embeddings $\{h_i\}_{i=1}^n \in \mathbb{R}^{n \times d}$ and **query(Q)** based on the global context $g_v$. $F_{Global}$ produces the attention output as follows,

$$H_{attn} = \text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{d}})\mathbf{V}, \qquad Q = \mathcal{W}_q g_v,\ K = \mathcal{W}_k \{h_i\}_{i=1}^n,\ V = \mathcal{W}_v \{h_i\}_{i=1}^n \tag{5}$$

where $\mathcal{W}_q$, $\mathcal{W}_k$, and $\mathcal{W}_v$ are the embedding weights. $H_{attn}$ is a weighted sum of values (local semantics), where the weight of each value is calculated based on its interaction with the global context query $\mathbf{Q}$. $H_{attn}$ is further encoded by a feed-forward layer to produce the contextual embedding $H_{context}$. We calculate the mean of $\{h_i\}_{i=1}^n$ and concatenate it with $H_{context}$ to obtain the final video embedding $\vartheta^k = \text{concat}(mean(\{h_i\}_{i=1}^n), H_{context})$; see Figure 2.

# 3 Cross-Modal Cycle Consistency

We introduce a cross-modal cycle-consistency loss to enforce the semantic alignment between clips and sentences, as illustrated in Figure 3. It replaces the cross-modal attention units used in [8, 14]. A pair of clip and sentence will be identified as semantically aligned if they are nearest neighbors in the learned common spaces. Consider as input a sequence of clip embeddings $\{\vartheta_i\}_{i=1}^{n} = \{\vartheta_1, \ldots, \vartheta_n\}$ and sentence embeddings $\{\delta_i\}_{i=1}^{m} = \{\delta_1, \ldots, \delta_m\}$. As the sentences of a paragraph have a temporal order, given a sentence embedding $\delta_i$ on this sequence, we first find its soft nearest neighbor [36–38]

$$\bar{\vartheta}_{\delta_i} = \sum_{j=1}^{n} \alpha_j \vartheta_j \quad \text{where} \quad \alpha_j = \frac{\exp(-\|\delta_i - \vartheta_j\|^2)}{\sum_{k=1}^{n} \exp(-\|\delta_i - \vartheta_k\|^2)} \tag{6}$$

in the clip sequence $\{\vartheta_i\}_{i=1}^{n}$. $\alpha_j$ is the similarity score of clip $\vartheta_j$ to sentence $\delta_i$. We then cycle back from $\bar{\vartheta}_{\delta_i}$ to the sentence sequence $\{\delta_i\}_{i=1}^{m}$ and calculate the soft location

$$\mu = \sum_{j=1}^{m} \beta_j j \quad \text{where} \quad \beta_j = \frac{\exp(-\|\bar{\vartheta} - \delta_j\|^2)}{\sum_{k=1}^{m} \exp(-\|\bar{\vartheta} - \delta_k\|^2)}. \tag{7}$$

The sentence embedding $\delta_i$ is semantically cycle consistent if and only if it cycles back to the original location, i.e., $i = \mu$. We penalize deviations from cycle-consistency for sampled sets of clips and sentences, which encourages the model to learn semantically consistent representations.

Our objective is the distance between the source location $i$ and the soft destination location $\mu$.

$$\ell_{CMC} = \|i - \mu\|^2 \tag{8}$$

Computing nearest neighbors as soft nearest neighbors makes the loss differentiable [36–38]. We can use this loss in both supervised and self-supervised scenarios. In the self-supervised case, we split each video uniformly into several clips and each paragraph into sentences. Beside the cycle-consistency from text to video, we also calculate $\ell_{CMC}$ from video to text. Therefore, the final $\ell_{CMC}$ loss includes both cycles.

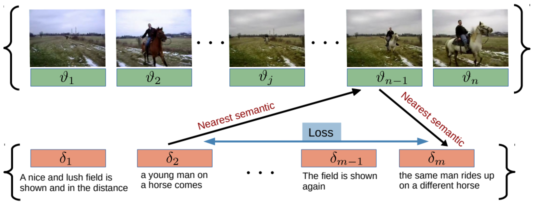


Figure 3: **Cross-Modality Cycle-Consistency.** Starting from a sentence $s_i$, we find its nearest neighbor in the clip sequence and again its neighbor in the sentence sequence. Deviations from the start index are penalized as alignment error.

The final training loss for the overall model is:

$$\ell_{final} = \ell_{align}^{L} + \ell_{align}^{H} + \ell_{align}^{g} + \ell_{cluster} + \lambda \ell_{CMC} \tag{9}$$

# 4 Experimental Setup

**Datasets.** We evaluate our method on the datasets **ActivityNet-captions** [39] and **Youcook2** [40]. ActivityNet-captions consists of 20k YouTube videos with an average length of 2 minutes, with 72k clip-sentence pairs. There are ~10k, ~5k and ~5k videos in train, val1 and val2, respectively. Youcook2 contains 2000 videos with a total number of 14k clips. This dataset is collected from YouTube and covers 89 types of recipes. There are ~9.6k clips for training and ~3.2k clips for validation. For each clip there is a manually annotated textual description.

**Evaluation Metrics.** We measure the performance on the retrieval task with standard retrieval metrics, i.e., recall at K (R@K e.g. R@1, R@5, R@10) and Median Rank (MR).

**Text encoding.** We feed paragraphs consisting of several sentences into a pretrained "BERT-Base, Uncased" model [19] and use the per-token outputs of the last 2 layers, resulting in 1536-d features.

**Video encoding.** For Activitynet-Captions, we use the 2048-d features provided by Zhang et al. [21] (at 3.8 FPS). For Youcook2, we test two approaches: (*A*) We follow Miech et al., 2019 [16] and concatenate 2D (Resnet-152 pretrained on ImageNet [41]) and 3D (ResNext-101 model [42] pretrained on Kinetics [43]) outputs to obtain 4096-d features at 3 FPS; (*B*) We use the video embedding network provided by Miech et al., 2020 [17] pretrained on video-text learning on the Howto100m dataset to obtain 512-d features at 0.6 FPS.

For each clip as well as for the entire video, we sample up to 80 frame features. If needed, we split the frames into 80 equal length intervals and uniformly sample a frame from each interval during training or take the center frame during validation.

Table 1: **Ablation study on ActivityNet-captions (val1).** We quantify the individual contributions of the attention-aware feature aggregation (AF), the Contextual Transformer (CoT), and the cross-modal cycle-consistency loss (CMC). HSE results are reproduced by us. Disabling CoT means removing the cross-attention layer between local and global context.

| Model | Pooling | CMC | CoT | Paragraph $\Longrightarrow$ Video | | | Video $\Longrightarrow$ Paragraph | | | Param (M) |
|---|---|---|---|---|---|---|---|---|---|---|
| | Lowlvl | | | R@1 | R@5 | R@50 | R@1 | R@5 | R@50 | |
| HSE | Max | ✗ | ✗ | 45.6±0.3 | 76.1±0.7 | 96.0±0.3 | 44.9±0.5 | 75.8±1.2 | 95.8±0.4 | 26.1 |
| HSE | Max | ✓ | ✗ | 46.6±0.4 | 78.1±0.3 | 97.3±0.1 | 46.4±0.3 | 77.6±0.3 | 97.1±0.3 | 26.1 |
| COOT | CLS | ✗ | ✗ | 49.4±1.4 | 77.7±1.3 | 95.7±0.2 | 49.7±1.9 | 77.8±0.9 | 95.8±0.3 | 4.9 |
| COOT | AVG | ✗ | ✗ | 52.6±0.6 | 80.6±0.4 | 97.0±0.2 | 52.1±0.4 | 80.8±0.2 | 97.0±0.2 | 4.9 |
| COOT | Max | ✗ | ✗ | 58.2±0.5 | 84.9±0.2 | 98.1±0.1 | 58.7±0.5 | 86.0±0.2 | 98.2±0.1 | 4.9 |
| COOT | AFA | ✗ | ✗ | 59.0±0.5 | 85.4±0.2 | 98.2±0.0 | 59.8±0.6 | 85.8±0.8 | 98.2±0.1 | 5.8 |
| COOT | Max | ✓ | ✓ | 59.4±0.9 | 86.1±0.6 | 98.3±0.0 | 60.5±0.1 | 87.1±0.2 | 98.5±0.1 | 6.7 |
| COOT | AFA | ✗ | ✓ | 59.8±1.1 | 86.3±0.3 | 98.5±0.1 | 60.1±0.1 | 87.1±0.4 | 98.5±0.1 | 7.6 |
| COOT | AFA | ✓ | ✗ | 59.5±0.5 | 85.5±0.4 | 98.1±0.0 | 60.5±0.7 | 86.2±0.5 | 98.2±0.1 | 5.8 |
| COOT | AFA | ✓ | ✓ | **60.8**±0.6 | **86.6**±0.4 | **98.6**±0.1 | **60.9**±0.3 | **87.4**±0.5 | **98.6**±0.0 | 7.6 |

**Training.** Similar to [21] we set all margins $\alpha = \alpha_g = \beta = \gamma = \mu = 0.2$. We use a mini-batch size of 64 video/paragraph pairs and sample all corresponding clips and sentences. All possible combinations of embeddings with non-matching indices in a batch are used as negative samples for the contrastive loss. To apply the cycle-consistency loss, we found that sampling 1 clip per video and 1 sentence per paragraph works best. The optimal loss weight $\lambda$ depends on architecture and dataset. As activation function, we found GELU [34] to perform best. We set the hidden size to 384 and use a pointwise linear layer to reduce the input feature dimension. We use one self-attention layer for the T-Transformer and one self-attention and one cross-attention layer for CoT. For further details on optimization and hyperparameters we refer the interested reader to the supplementary material.

**Video-Language Retrieval.** For video-text retrieval, the query is a paragraph and the task is to find the most relevant video from a database. Alternatively, the query can be a video and the task is to retrieve the most relevant paragraph. We follow the experimental protocol from Zhang et al. [21] to evaluate the models. We use the final embedding output of our model ($\vartheta^k, \delta^k$) to do the retrieval.

**Clip-sentence retrieval.** For Youcook2, we also evaluate the quality of our model when retrieving a short video clip given a single sentence. For this experiment, we use the intermediate low-level embeddings produced by our model ($\vartheta_i^k, \delta_i^k$) to do the retrieval.

## 5 Results

**Influence of each component.** We show results of a model ablation study in Table 1. First, to validate the general effectiveness of the proposed cross-modal cycle consistency loss (CMC), we apply it to the HSE architecture [21]. The $\ell_{CMC}$ loss provides a significant boost in performance for both HSE and COOT, which indicates that it will be beneficial if plugged into other video-text representation learning methods. Second, the Attention-FA module shows better performance (7.2% average improvement on R@1 for paragraph $\Longrightarrow$ video and video $\Longrightarrow$ paragraph tasks) than common average pooling. Third, we observe that integrating the Contextual Transformer into the overall model improves the performance. This confirms that interactions between local and global context help the model to highlight the relevant semantics (more in supp. material).

**Comparison to the state of the art.** Table 2 summarizes the results of paragraph to video and video to paragraph retrieval tasks on the ActivityNet-captions dataset. For a fair comparison, our model utilizes the same video features as HSE [21]. Our method significantly outperforms all previous methods across different evaluation metrics. COOT obtains on average 16.6% better R@1 in comparison to HSE [21] while having fewer parameters. We believe the major gain comes from our attention-aware feature aggregation component and the $\ell_{CMC}$ loss.

We further provide retrieval results on the Youcook2 [40] dataset in Table 3. We compare our model under two settings: (1) with features pretrained on classification (2) with features from a pretrained SOTA video-text model.

Table 2: **Video-paragraph retrieval results on AcitvityNet-captions dataset (val1).**

| | Paragraph ⟹ Video | | | | Video ⟹ Paragraph | | | |
|---|---|---|---|---|---|---|---|---|
| Method | R@1 | R@5 | R@50 | MR | R@1 | R@5 | R@50 | MR |
| LSTM-YT [52] | 0.0 | 4.0 | 24.0 | 102.0 | 0.0 | 7.0 | 38.0 | 98.0 |
| No Context [53] | 5.0 | 14.0 | 32.0 | 78.0 | 7.0 | 18.0 | 45.0 | 56.0 |
| DENSE [39] | 14.0 | 32.0 | 65.0 | 34.0 | 18.0 | 36.0 | 74.0 | 32.0 |
| VSE [54]( [5]) | 11.7 | 34.7 | 85.7 | 10 | - | - | - | - |
| FSE [21] | 18.2 | 44.8 | 89.1 | 7 | 16.7 | 43.1 | 88.4 | 7 |
| HSE [21] | 44.4±0.5 | 76.7±0.3 | 97.1±0.1 | 2 | 44.2±0.6 | 76.7±0.3 | 97.0±0.3 | 2 |
| COOT | **60.8**±0.6 | **86.6**±0.4 | **98.6**±0.1 | **1** | **60.9**±0.3 | **87.4**±0.5 | **98.6**±0.0 | **1** |

*Without HowTo100M pretrained features.* We use features ($A$) explained in Section 4 and train the COOT model on the YouCook2 dataset. Using the same training set, COOT outperforms Miech et al. [16] and HGLMM [44] on both paragraph-to-video and sentence-to-clip tasks. This supports our rationale that modeling interactions between different hierarchy levels is crucial for capturing long-term semantics.

*With HowTo100M pretrained features.* In Table 3, we compare our method with the recently proposed SOTA methods MIL-NCE [17], ActBERT [8], and Miech et al. [16], which utilize pretraining on the huge HowTo100M dataset. We use features ($B$) (Sec. 4) and train the model on the YouCook2 dataset. Note that the paragraph to video results of other methods are computed by us. Training our model with features of a model pretrained on the HowTo100M dataset clearly improves over training with features of a model pretrained on classification and over the state-of-the-art. We can see that our model outperforms MIL-NCE [17] 16.4% on R@1 score for paragraph-to-video task, which verifies that COOT benefits from hierarchy interactions. This shows that the contributions of this paper are complementary to works that focus on large-scale pretraining.

**Time complexity and number of parameters.** The COOT model has 10.6M, parameters which is 60% less than the HSE method (Table 1). Training is fast and takes less than 3 hours on two GTX1080Ti GPUs (without data I/O).

### 5.1 Video Captioning

To show that the learned representations contain meaningful information for other tasks than retrieval, we use the learned representations for video captioning building upon the captioning model MART [45]. The original method uses appearance (RGB) and optical flow features extracted from ResNet-200 [41] and BN-Inception [46], respectively.

We use the clip ($\vartheta_i^k$) and optionally the video ($\vartheta^k$) representation generated with our COOT model. In comparison to MART, we input about 100 times less features per video into the captioning model. We use the standard language evaluation metrics BLEU@3/4 [47], RougeL [48], METEOR [49], CIDEr-D [50] and R@4 [51] which measures the degree of n-gram repetition. Our results in Table 4 and Table 5 show that the MART method using our representations improves over using appearance and optical flow video features. Generated captions in Table 6 show that our video representations encapsulate richer information about the video while being more compact.

## 6 Related Work

**Image and Language.** Many self-supervised visual-language representation learning methods have focused on improving one representation with the help of the other [25, 44, 57–61]. These approaches learn joint image-text embeddings or map images and sentences into a common space. Recently, there has been a surging interest in utilizing Transformers for image-text representation learning [62–66]. ViLBERT [13] and VisualBERT [12] pretrain a BERT-like architecture on an image-text dataset and then transfer learned representations to different downstream tasks. LXMERT [14] uses additional pretraining tasks and an object-relationship component. In contrast to [13, 14], VL-BERT [15] does not utilize the task of sentence-image relationship prediction and additionally pretrains the model on text-only datasets.

Table 3: **Retrieval results on YouCook2 dataset.** Results with * are computed by us. $^{\triangle}$ we use features of a video-text model [17] pretrained on the HowTo100m dataset.

| | | Paragraph ⟹ Video | | | | Sentence ⟹ Clip | | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | TrainSet | R@1 | R@5 | R@10 | MR | R@1 | R@5 | R@10 | MR |
| Random | - | 0.21 | 1.09 | 2.19 | 229 | 0.03 | 0.15 | 0.3 | 1675 |
| Miech et al. [16] | HowTo100M | 43.1* | 68.6* | 79.1* | 2* | 6.1 | 17.3 | 24.8 | 46 |
| ActBERT [8] | HowTo100M | - | - | - | - | 9.6 | 26.7 | 38.0 | 19 |
| MIL-NCE [17] | HowTo100M | 61.9* | 89.4* | 98.9* | **1***| 15.1 | 38.0 | 51.2 | 10 |
| HGLMM [44] | YouCook2 | - | - | - | - | 4.6 | 14.3 | 21.6 | 75 |
| Miech et al. [16] | YouCook2 | 32.3* | 59.2* | 70.9* | 4* | 4.2 | 13.7 | 21.5 | 65 |
| COOT | YouCook2 | $50.4_{\pm 2.6}$ | $79.4_{\pm 0.6}$ | $87.4_{\pm 0.8}$ | $1.3_{\pm 0.6}$ | $5.9_{\pm 0.7}$ | $16.7_{\pm 0.6}$ | $24.8_{\pm 0.8}$ | $49.7_{\pm 2.9}$ |
| Miech et al. [16] | HowTo100M+<br>YouCook2 | 59.6* | 86.0* | 93.6* | **1***| 8.2 | 24.5 | 35.3 | 24 |
| COOT | HowTo100M$^{\triangle}$+<br>YouCook2 | $\mathbf{77.2}_{\pm 1.0}$ | $\mathbf{95.8}_{\pm 0.8}$ | $\mathbf{97.5}_{\pm 0.3}$ | $\mathbf{1.0}_{\pm 0.0}$ | $\mathbf{16.7}_{\pm 0.4}$ | $\mathbf{40.2}_{\pm 0.3}$ | $\mathbf{52.3}_{\pm 0.5}$ | $\mathbf{9.0}_{\pm 0.0}$ |

Table 4: **Captioning results on the YouCook2 dataset (val split).** Results with * are computed by us. $^{\triangle}$ we use features of a video-text model [17] pretrained on the HowTo100m dataset. "MART w/o re" denotes a MART variant without recurrence.

| Features | Method | TrainSet | B@3 | B@4 | RougeL | METEOR | CIDEr-D | R@4↓ |
|---|---|---|---|---|---|---|---|---|
| RGB+Flow | VTransformer [55] | YouCook2 | 13.08* | 7.62 | 32.18* | 15.65 | 32.26 | 7.83 |
| RGB+Flow | TransformerXL [56] | YouCook2 | 11.46* | 6.56 | 30.78* | 14.76 | 26.35 | 6.30 |
| RGB+Flow | MART [45] | YouCook2 | 12.83* | 8.00 | 31.97* | 15.90 | 35.74 | **4.39** |
| COOT clip | MART | YouCook2 | 14.17 | 8.69 | 33.01 | 16.11 | 38.28 | 8.07 |
| COOT video+clip | MART | YouCook2 | 15.75 | 9.44 | 34.32 | 18.17 | 46.06 | 6.30 |
| COOT clip | MART | H100M$^{\triangle}$+YC2 | 17.12 | 10.91 | 37.59 | 18.85 | 54.07 | 5.11 |
| COOT clip | MART w/o re. | H100M$^{\triangle}$+YC2 | 17.16 | 10.69 | 37.43 | 19.18 | 54.85 | 5.45 |
| COOT clip | VTransformer | H100M$^{\triangle}$+YC2 | 17.62 | 11.09 | 37.63 | 19.34 | 54.67 | 4.57 |
| COOT video+clip | VTransformer [55] | H100M$^{\triangle}$+YC2 | 17.79 | 11.05 | 37.51 | 19.79 | 55.57 | 5.69 |
| COOT video+clip | MART | H100M$^{\triangle}$+YC2 | **17.97** | **11.30** | **37.94** | **19.85** | **57.24** | 6.69 |

**Video and Language.** The multi-modal nature of video is a great source of self-supervision. Modalities such as audio, text and motion provide strong cues to learn richer spatio-temporal features [9, 11, 16, 17, 67–69]. Aytar et al. [70] leverage natural synchronization to learn rich representations across vision, sound, and language. VideoBERT [11] learns joint video-text representations based on predicting whether the linguistic sentence is temporally aligned with the visual sentence. These approaches [8, 11, 23] focus on self-supervised pretraining and require a large set of paired video clips and texts to learn a good representation model [17].

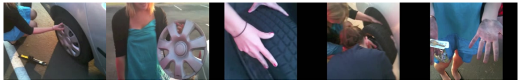
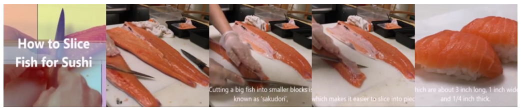
There has been growing interest in temporal localization of natural language in videos [2, 7, 71–73]. Moment localization identifies a time window given a text query [7, 71, 74]. Most related to our work are methods that focus on joint video-text embeddings and perform video-text retrieval or captioning [4–6, 9, 21, 52, 69, 75, 76]. Several works tried to utilize the temporal structure of video and text for the alignment task [4, 70, 77–79]. Miech et al. [68] proposed a mixture of experts approach to learn text-video representations. Likewise, CE [20] proposes a mixture-of-experts model to aggregate information from pretrained experts (e .g. object, action, audio) with a gating mechanism. In our work, we use a similar hierarchy as CMHSE [21, 22, 24] and extend their design by proposing three new components to learn the interactions between different levels of the hierarchy.

**Cycle-Consistency.** Cycle-Consistency uses transitivity as an objective for training [38, 80–82]. The assumption of cyclical structure has been used in various works [83–85]. Wang et al . [80] obtain the supervision for visual correspondence by tracking forward and backward. Shah et al. [81] enforce consistency between the generated and the original question in visual question answering. To prevent mode collapse, cycle-consistency is activated only after a certain number of training iterations [81]. TCC [38] employs cycle-consistency for temporal video alignment. In contrast to TCC, which works only in the video domain, we align video and text. To the best of our knowledge, this is the first work which introduces cycle-consistency to the video-text domain.

Table 5: **Captioning results on the ActivityNet-Captions dataset (ae-test split of MART [45]).** Results with * are computed by us. "MART w/o re" denotes a MART variant without recurrence.

| Features | Method | TrainSet | B@3 | B@4 | RougeL | METEOR | CIDEr-D | R@4↓ |
|---|---|---|---|---|---|---|---|---|
| RGB+Flow | VTransformer [55] | ActivityNet | 16.27* | 9.31 | 29.18* | 15.54 | 21.33 | 7.45 |
| RGB+Flow | TransformerXL [56] | ActivityNet | 16.71* | 10.25 | 30.53* | 14.91 | 21.71 | 8.79 |
| RGB+Flow | MART | ActivityNet | 16.43* | 9.78 | 30.63* | 15.57 | 22.16 | 5.44 |
| COOT video+clip | TransformerXL [56] | ActivityNet | 16.94 | 10.57 | 30.93 | 14.76 | 22.04 | 15.85 |
| COOT video+clip | VTransformer [55] | ActivityNet | 16.80 | 10.47 | 30.37 | 15.76 | 25.90 | 19.14 |
| COOT clip | MART w/o re. | ActivityNet | 15.41 | 9.37 | 28.66 | 15.61 | 22.05 | 12.03 |
| COOT video+clip | MART w/o re. | ActivityNet | 16.59 | 10.33 | 29.93 | 15.64 | 25.41 | 17.03 |
| COOT clip | MART | ActivityNet | 16.53 | 10.22 | 30.68 | 15.91 | 23.98 | **5.35** |
| COOT video+clip | MART | ActivityNet | **17.43** | **10.85** | **31.45** | **15.99** | **28.19** | 6.64 |

Table 6: **Captioning samples, more accurate (left) and less accurate (right) cases.** First row: ActivityNet (ae-test split), second row: YouCook2 (val split). Red/bold indicates content errors, blue/italic indicates repetitive patterns.

| | |
|---|---|
|  | |
| **MART**: A person is **driving the car**. **A boy** is **holding a bottle of wood**.<br>**COOT (Ours)**: A woman is seen kneeling down next to a car while others stand around her. The woman then pushes the tire **back and fourth**.<br>**GT**: A girl is shown trying to change a tire. She successfully removes the tire, then replaces it with a spare, showing off their dirty hands afterward. | **MART**: A man *is kneeling down on* a floor. He *is kneeling down on* the ground.<br>**COOT (Ours)**: A man is seen kneeling down on the ground and begins **putting shoes on**. The man continues to *put on the shoes* and ends by *putting his shoes on*.<br>**GT**: A person is seen bending over a floor placing tiles down over the plaster. The person continues laying tiles down and pushing down on the floor to make sure it's sturdy. |
| |  |
| **MART**: Heat up a pan and **cook until golden brown**. Add **onions** *to the pan*. Add **flour** salt and **pepper** *to the pan*. Add **rice** *to the pan* and stir.<br>**COOT (Ours)**: Boil the potatoes in water. Add chopped *potatoes* to the pan. Add butter and mash. Add some milk and *mash*.<br>**GT**: Boil some small pieces of potatoes in water. Mash the potato. Add some butter and salt and stir. Gradually add milk while stirring the potatoes. | **MART**: Cut the salmon into thin slices. *Cut the salmon into thin slices*. *Cut the salmon into thin slices*. *Cut the salmon into thin slices*.<br>**COOT (Ours)**: Cut the salmon in half. *Cut the salmon in half*. *Cut the salmon* into thin slices. *Cut the salmon into thin slices*.<br>**GT**: Slice the fish into smaller pieces. Chop the tail end off. Cut the fish at an angle. Cut the fish into thin pieces. |

## 7 Conclusions

We have presented a cooperative hierarchical transformer architecture for learning a joint video and text embedding space where similar semantics are aligned. The architecture is designed to encourage the use of long-range temporal context in a cross-level manner. Our approach uses two new components to model the interactions within and between hierarchy levels; an attention-aware feature aggregation module to model the interactions between frames and words, a contextual transformer to model the interactions between local contexts and global context. In addition, we have introduced a new cross-modal cycle-consistency loss which enforces the semantic alignment of clips and sentences. We have shown that both components contribute – jointly and individually – to an improved retrieval performance. As a result, our approach achieves state-of-the-art retrieval and captioning performance on two challenging datasets.

## Broader Impact

This work contributes fundamental research and does not present any foreseeable societal consequence. In the long run, this line of research can contribute to services on video search and video organisation.

## Acknowledgments

We thank Ehsan Adeli for helpful comments, Antoine Miech for providing details on their retrieval evaluation, and Facebook for providing us a GPU server with Tesla P100 processors for this research work.

## References

[1] Satwik Kottur, Ramakrishna Vedantam, José M. F. Moura, and Devi Parikh. Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. *CoRR*, abs/1511.07067, 2015.

[2] 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.

[3] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer, 2016.

[4] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013.

[5] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[6] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] Yi Yang Linchao Zhu. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

[9] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, page 19–27, New York, NY, USA, 2018. Association for Computing Machinery.

[10] B. A. Plummer, M. Brown, and S. Lazebnik. Enhancing video summarization via vision-language embedding. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1052–1060, 2017.

[11] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7464–7473, 2019.

[12] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[13] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[14] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[15] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[16] 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.

[17] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS 2017, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, Jun 2019. Association for Computational Linguistics.

[20] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *arXiv preprint arxiv:1907.13487*, 2019.

[21] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, pages 385–401, 2018.

[22] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1029–1038, 2016.

[23] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019.

[24] Jiwei Li, Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1106–1115, Beijing, China, Jul 2015. Association for Computational Linguistics.

[25] Donghyun Kim, Kuniaki Saito, Kate Saenko, Stan Sclaroff, and Bryan A. Plummer. Mule: Multimodal universal language embedding, 2019.

[26] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 2, pages 1735–1742, 2006.

[27] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. *CoRR*, abs/1703.07464, 2017.

[28] G. Kordopatis-Zilo, S. Papadopoulos, I. Patras, and Y. Kompatsiaris. Near-duplicate video retrieval with deep metric learning. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 347–356, 2017.

[29] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. *CoRR*, abs/1810.06951, 2018.

[30] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *CoRR*, abs/1706.06905, 2017.

[31] Mateusz Malinowski and Mario Fritz. Learnable pooling regions for image classification. In *ICLR workshop*, May 2013.

[32] Qian Chen, Zhen-Hua Ling, and Xiaodan Zhu. Enhancing sentence embedding with generalized pooling. *CoRR*, abs/1806.09828, 2018.

[33] Hayoung Eom and Heeyoul Choi. Alpha-pooling for convolutional neural networks. *CoRR*, abs/1811.03436, 2018.

[34] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[35] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[36] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic. Neighbourhood consensus networks. *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2018.

[37] Jacob Goldberger, Geoffrey E Hinton, Sam T. Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2005.

[38] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019.

[39] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[40] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2018.

[41] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[42] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[43] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[44] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4437–4446, 2015.

[45] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning. In *ACL*, 2020.

[46] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, Jul 2002. Association for Computational Linguistics.

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

[49] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, Jun 2014. Association for Computational Linguistics.

[50] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015.

[51] Yilei Xiong, Bo Dai, and Dahua Lin. Move forward and tell: A progressive generator of video descriptions. In *ECCV*, 2018.

[52] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[53] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[54] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., 2013.

[55] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer, 2018.

[56] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

[57] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[58] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.

[59] Gil Sadeh, Lior Fritz, Gabi Shalev, and Eduard Oks. Joint visual-textual embedding for multimodal style search. *CoRR*, abs/1906.06620, 2019.

[60] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. *CoRR*, abs/1511.06078, 2015.

[61] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1928–1937, 2017.

[62] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning, 2019.

[63] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data, 2020.

[64] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *arXiv preprint arXiv:1909.11059*, 2019.

[65] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers, 2020.

[66] Kazuki Miyazawa, Tatsuya Aoki, Takato Horii, and Takayuki Nagai. lambert: Language and action learning using multimodal bert. *arXiv preprint arXiv:2004.07093*, 2020.

[67] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[68] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. In *arXiv*, 2018.

[69] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. *CoRR*, abs/1505.01861, 2015.

[70] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. See, hear, and read: Deep aligned representations. *CoRR*, abs/1706.00932, 2017.

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

[72] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[73] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: temporal activity localization via language query. In *ICCV*, 2017.

[74] Mingfei Gao, Larry Davis, Richard Socher, and Caiming Xiong. Wslln:weakly supervised natural language localization networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[75] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval, 2020.

[76] Ran Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2346–2352. AAAI Press, 2015.

[77] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV - IEEE International Conference on Computer Vision*, pages 4462–4470, Santiago, Chile, dec 2015. IEEE.

[78] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1827–1835, 2015.

[79] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[80] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[81] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6649–6658, 2019.

[82] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.

[83] Y. Chen, Y. Lin, M. Yang, and J. Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1791–1800, 2019.

[84] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 820–828, Red Hook, NY, USA, 2016. Curran Associates Inc.

[85] Duyu Tang, Nan Duan, Zhao Yan, Zhirui Zhang, Yibo Sun, Shujie Liu, Yuanhua Lv, and Ming Zhou. Learning to collaborate for question answering and asking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1564–1574, New Orleans, Louisiana, jun 2018. Association for Computational Linguistics.

[86] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. *CoRR*, abs/1807.01774, 2018.

[87] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *CoRR*, abs/1706.02515, 2017.

[88] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[89] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, April 2020.

[90] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[91] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

# A Appendix

## A.1 Implementation Details

**Hyperparameters.** To select hyperparameters for our model, we used a combination of manual search and BOHB [86] to explore the hyperparameters space. In Table 7, we provide an overview of our hyperparameter search.

After testing different activation functions ReLU, SELU [87], ELU [88] and GELU [34] we found GELU to perform best. Increasing the capacity of the model by using a higher attention dimension can help improve the results, but comes at the cost of higher memory requirements and more difficult optimization.

Table 7: **Hyperparameters.** This table shows the hyperparameter ranges we considered and the final choices for our three best models (ActivityNet-captions, YouCook2-Resnet/Resnext features, Youcook2-Howto100m features.). ROP denotes the *Reduce on Plateau* Scheduler we used. Dimensions given in multiples (1x, 2x) refer to multiples of the Attention Dimension parameter. FF denotes Feed-Forward. AF is our Attention-aware Feature Aggregation module.

| Hyperparameter | Considered Range | | ActivityNet | Youcook2 | |
|---|---|---|---|---|---|
| | | | | Resnet/ResneXt | Howto100m |
| Optimizer | Adam, RAdam, SGD | | Adam | RAdam | RAdam |
| Learning rate | 1e-5 | 1e-2 | 1e-3 | 3.6e-4 | 9e-4 |
| Weight Decay | 0 | 1e-2 | 2e-5 | 2e-5 | 0 |
| Momentum | 0.5 | 0.99 | 0.9 | 0.56 | 0.56 |
| Adam Beta2 | 0.9 | 0.9999 | 0.999 | 0.98 | 0.98 |
| Adam Epsilon | 1e-10 | 1e-7 | 1e-8 | 1.5e-9 | 1.5e-9 |
| Warmup Epochs | 0 | 8 | 3 | 0 | 0 |
| ROP Patience | 2 | 10 | 2 | 5 | 5 |
| ROP Cooldown | 0 | 3 | 3 | 3 | 3 |
| Attention Layers | 1 | 3 | 1 | 1 | 1 |
| Attention Dimension | 256 | 1024 | 384 | 384 | 384 |
| Attention Heads | 1 | 8 | 8 | 8 | 8 |
| Attention FF Dimension | 1x | 2x | 1x | 1x | 1x |
| AF Dimension | 1x | 2x | 2x | 2x | 2x |
| AF Heads | 1 | 8 | 2 | 2 | 2 |
| Number of AF modules | 1 | 2 | 1 | 1 | 1 |
| Dropout | 0% | 10% | 2.5% | 1% | 5% |
| Gaussian Noise on Frame Features | 0 | 1 | 0 | 0.01 | 0 |

**Optimization.** We tried several optimizers such as Adam, RAdam [89] and SGD. If carefully configured, RAdam can improve over Adam.

We schedule the Learning Rate with a *Reduce on Plateau* approach: Whenever our validation metric does not improve for a certain number of epochs, we reduce the learning rate by a factor of 10. After no improvements for 15 epochs, we terminate the training process. As relevant metric we defined the sum of R@1 Retrieval Score for video-paragraph and paragraph-video retrieval on Activitynet-Captions and the sum of R@1 Retrieval Score for clip-sentence and sentence-clip retrieval on Youcook2. Careful tuning of the optimizer parameters, using an automated search method like BOHB [86] to search parts of the parameters space, was crucial to train the models properly.

**Strength of the cross-modal cycle-consistency loss.** For Activitynet we set $\lambda = 0.01$ and for Youcook2 we set $\lambda = 0.001$.

For weight initialization, we utilized Uniform, Normal and Truncated Normal distributions. The best results were obtained with initializing weights randomly from the Truncated Normal distribution with a standard deviation of $0.01$, redrawing all samples with more than 2 standard deviations.

To cope with the overfitting problem, the different regularization methods (Dropout, Weight Decay, CMC-loss, Gaussian Noise on Frame Features) need to be traded off carefully to obtain good results (see Table 7).

**Preprocessing.** For ActivityNet captions, we found it helpful to expand all clips to be at least 10 frames long. Expanding is done by iteratively adding frames to the start and end of the clip until we reach the desired length.

Table 8: **Text feature ablation study on ActivityNet-captions (val1).** We evaluate our choice of text encoding and show that Bert [19] outperforms GloVe [90] on both models and all metrics.

| Model | Text | Paragraph ⟹ Video | | | Video ⟹ Paragraph | | |
|---|---|---|---|---|---|---|---|
| | | R@1 | R@5 | R@50 | R@1 | R@5 | R@50 |
| HSE | GloVe | 45.7±0.3 | 76.1±0.7 | 96.0±0.3 | 44.9±0.5 | 75.8±1.2 | 95.8±0.4 |
| HSE | Bert | 47.0±1.1 | 77.0±1.5 | 96.1±0.4 | 46.9±0.8 | 77.2±1.1 | 95.9±0.6 |
| COOT | GloVe | 56.5±1.1 | 84.1±1.3 | 98.0±0.3 | 57.3±1.8 | 84.5±1.4 | 98.2±0.2 |
| COOT | Bert | **60.8**±0.6 | **86.6**±0.4 | **98.6**±0.1 | **60.9**±0.3 | **87.4**±0.5 | **98.6**±0.0 |

Table 9: **Loss function ablation study on ActivityNet-captions (val1).** We analyse performance of the COOT model while removing loss components with different base models. CoT denotes using global attention in the contextual transformer. AF is our Attention-aware Feature Aggregation module.

| # | Pooling | CMC | CoT | Alignment | | | Clustering | | Par. ⟹ Video | | Video ⟹ Par. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Lowlvl | | | High | Low | Ctx | High | Low | R@1 | R@5 | R@1 | R@5 |
| 1 | Avg | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | 30.4±3.2 | 58.4±4.5 | 29.9±3.3 | 58.7±4.5 |
| 2 | Avg | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | 49.7±0.7 | 79.0±0.6 | 48.6±0.5 | 79.1±0.9 |
| 3 | Avg | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | 49.2±0.7 | 78.9±0.2 | 48.6±0.6 | 78.9±0.6 |
| 4 | Avg | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | 50.6±1.1 | 79.8±0.8 | 50.8±1.0 | 79.8±0.8 |
| 5 | Avg | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | 51.5±0.7 | 80.2±0.4 | 52.0±0.8 | 80.5±0.3 |
| 6 | Avg | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | 52.6±0.6 | 80.6±0.4 | 52.1±0.4 | 80.8±0.2 |
| 7 | Avg | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | 27.4±2.1 | 55.3±2.4 | 27.3±1.6 | 56.0±2.3 |
| 8 | Avg | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | 54.1±0.8 | 82.0±0.1 | 54.7±0.2 | 82.1±0.1 |
| 9 | Avg | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 53.6±0.1 | 81.7±0.0 | 53.5±0.5 | 81.7±0.7 |
| 10 | Max | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | 47.9±0.7 | 76.9±0.1 | 48.3±0.2 | 77.5±0.6 |
| 11 | Max | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | 56.5±0.3 | 84.5±0.2 | 56.6±0.4 | 85.2±0.1 |
| 12 | Max | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | 54.4±0.9 | 83.3±0.9 | 55.4±1.4 | 84.0±0.8 |
| 13 | Max | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | 55.3±0.8 | 83.0±0.8 | 56.4±1.1 | 83.7±1.2 |
| 14 | Max | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | 56.1±0.2 | 83.3±0.2 | 57.0±0.3 | 83.9±0.5 |
| 15 | Max | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | 58.2±0.5 | 84.9±0.2 | 58.7±0.5 | 86.0±0.2 |
| 16 | Max | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | 46.3±1.0 | 76.2±0.9 | 47.7±0.9 | 77.2±0.7 |
| 17 | Max | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | 57.5±0.5 | 84.8±0.2 | 58.1±1.0 | 85.3±0.4 |
| 18 | Max | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 59.4±0.9 | 86.1±0.6 | 60.5±1.0 | 87.1±0.2 |
| 19 | AF | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | 47.1±0.7 | 76.7±0.6 | 47.6±0.2 | 77.4±0.2 |
| 20 | AF | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | 56.3±0.3 | 84.0±0.2 | 56.8±0.7 | 84.7±0.3 |
| 21 | AF | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | 55.2±0.2 | 83.3±0.1 | 55.8±0.5 | 83.6±0.2 |
| 22 | AF | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | 57.8±0.3 | 84.6±0.2 | 58.1±0.3 | 85.1±0.2 |
| 23 | AF | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | 58.8±0.4 | 85.3±0.4 | 59.1±0.6 | 85.8±0.4 |
| 24 | AF | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | 59.0±0.5 | 85.4±0.2 | 59.8±0.6 | 85.8±0.8 |
| 25 | AF | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | 47.8±0.5 | 76.4±0.4 | 47.8±0.2 | 77.5±0.5 |
| 26 | AF | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | 59.5±0.5 | 85.5±0.4 | 60.5±0.7 | 86.2±0.5 |
| 27 | AF | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | 53.9±0.7 | 82.6±0.6 | 53.8±0.6 | 83.0±0.5 |
| 28 | AF | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | 55.1±5.3 | 83.4±3.6 | 55.5±4.7 | 83.8±3.2 |
| 29 | AF | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | 58.5±1.1 | 85.2±0.5 | 58.5±0.7 | 85.5±0.7 |
| 30 | AF | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **60.8**±0.6 | **86.6**±0.4 | **60.9**±0.3 | **87.4**±0.5 |

**Retrieval.** We L2-normalize the output embeddings of our model so the squared elements sum to 1. Retrieval is done by cosine similarity, e.g. given video embedding $v$, we retrieve paragraph embedding

$$p = \max_{\hat{p}\in\mathcal{D}} v^\top \hat{p} \tag{10}$$

## A.2 Ablation Studies

In this section, we provide ablation studies on the importance of low-level supervision, different text encoders performance, impact of different alignment losses in our final training loss and analysis on sequence pooling.

Table 10: **Evaluation of different sequence pooling methods on ActivityNet-captions (val1).** We switch both the low level (frames, words) and high level (clips, sentences) pooling methods and observe the changes in performance. In experiments denoted with *, we used a different optimizer setting.

| Pooling | | CMC | CoT | Par. ⟹ Video | | Video ⟹ Par. | |
|---|---|---|---|---|---|---|---|
| Low | High | | | R@1 | R@5 | R@1 | R@5 |
| AF | AF x1 | ✓ | ✓ | 42.6±0.4 | 76.5±0.3 | 42.1±0.8 | 76.9±0.7 |
| AF | AF x2 | ✓ | ✓ | 42.8±0.1 | 76.0±0.7 | 42.8±0.1 | 76.4±0.6 |
| AF* | AF x1 | ✓ | ✓ | 48.7±1.0 | 82.2±0.6 | 50.1±0.4 | 82.6±0.4 |
| AF* | AF x2 | ✓ | ✓ | 50.5±0.4 | 82.3±0.4 | 51.4±1.4 | 82.9±0.6 |
| Max | Max | ✓ | ✓ | 40.9±0.7 | 75.3±0.1 | 42.2±0.5 | 76.2±0.6 |
| AF | Max | ✓ | ✓ | 43.3±0.9 | 76.3±1.0 | 42.5±0.6 | 77.2±1.2 |
| CLS | Avg | ✗ | ✗ | 49.4±1.4 | 77.7±1.3 | 49.7±1.9 | 77.8±0.9 |
| CLS | Avg | ✓ | ✓ | 49.7±0.5 | 79.4±0.2 | 51.2±0.1 | 79.6±0.1 |
| Avg | Avg | ✗ | ✗ | 52.6±0.6 | 80.6±0.4 | 52.1±0.4 | 80.8±0.2 |
| Avg | Avg | ✓ | ✓ | 53.6±0.1 | 81.7±0.0 | 53.5±0.5 | 81.7±0.7 |
| Max | Avg | ✗ | ✗ | 58.2±0.5 | 84.9±0.2 | 58.7±0.5 | 86.0±0.2 |
| Max | Avg | ✓ | ✓ | 59.4±0.9 | 86.1±0.6 | 60.5±1.0 | 87.1±0.2 |
| AF | Avg | ✓ | ✓ | **60.8**±0.6 | **86.6**±0.4 | **60.9**±0.3 | **87.4**±0.5 |

Table 11: **Evaluation of different averagepooling methods.** We modify our exact approach to averagepooling in the high-level and evaluate the results.

**ActivityNet-captions** dataset:

| Sum | Pad | Divide | Par. ⟹ Video | | Video ⟹ Par. | |
|---|---|---|---|---|---|---|
| | | | R@1 | R@5 | R@1 | R@5 |
| All | Max(Batch, 16) | All | 44.2±2.5 | 75.4±2.5 | 44.1±2.0 | 75.9±2.0 |
| All | Max(Batch, 16) | Nonzero | 44.1±0.7 | 76.2±0.9 | 44.7±1.1 | 76.9±0.9 |
| All | Batch | All | 48.3±0.2 | 76.8±0.8 | 47.9±0.8 | 77.7±0.7 |
| Nonzero | Batch | Nonzero | 42.0±0.5 | 76.3±0.3 | 41.6±0.6 | 77.0±0.7 |
| All | Batch | Nonzero | **60.8**±0.6 | **86.6**±0.4 | **60.9**±0.3 | **87.4**±0.5 |

**Youcook2** dataset:

| Sum | Pad | Divide | Par. ⟹ Video | | Sent. ⟹ Clip | |
|---|---|---|---|---|---|---|
| | | | R@1 | R@5 | R@1 | R@5 |
| All | Max(Batch, 16) | All | **77.6**±0.7 | **96.3**±0.4 | **17.5**±0.3 | **40.7**±0.1 |
| All | Max(Batch, 16) | Nonzero | 74.7±2.0 | 95.0±0.6 | 16.9±0.5 | 39.7±0.8 |
| All | Batch | All | 77.4±1.5 | 96.2±1.6 | 17.2±0.6 | 39.9±0.3 |
| Nonzero | Batch | Nonzero | 74.2±2.6 | 94.7±0.7 | 16.8±0.3 | 40.2±0.6 |
| All | Batch | Nonzero | 77.2±1.0 | 95.8±0.8 | 16.7±0.4 | 40.2±0.3 |

**Importance of low-level supervision.** In Fig. 4, we study the effect of adding uniform noise to the start and end frame index of each clip in ActivityNet-captions from the interval $[-N_f * P, +N_f * P]$. $N_f$ is the total number of video frames and $P$ is the noise percentage. We also perform a "full" noise experiment where we drop the temporal alignment labels of clips and sentences completely. We observe that increasing the noise from 0% to 40% consistently decreases the performance as labels get less reliable. For noise more than 40%, we do not observe significant changes in performance anymore. This is probably because at this noise level the labels become useless and are ignored. Still a good performance is obtained.

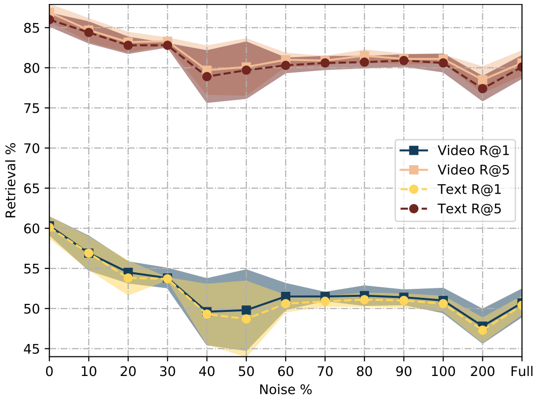
The study shows that the model is robust to noisy and missing low-level supervision and COOT is still able to capture useful dynamics between low-level and high-level semantics.

**Impact of Text Encoding.** We conduct ablation experiments to evaluate the importance of the text encoder for representation learning task. The ablation study results are shown in Table 8. We first evaluate the COOT model and the HSE [21] model with GloVe [90] features. We then replace

Table 12: **Video-paragraph retrieval results on AcitvityNet-captions dataset (val2).**

| Method | Par. ⟹ Video | | Video ⟹ Par. | |
|---|---|---|---|---|
| | R@1 | R@5 | R@1 | R@5 |
| FSE | 11.5 | 31.0 | 11.0 | 30.6 |
| HSE | 32.9 | 62.7 | 32.6 | 63.0 |
| COOT | **48.5** | **78.9** | **48.9** | **79.5** |

Figure 4: Noise vs Performance study on ActivityNet-captions dataset (val1)



GloVe features with features obtained from a pretrained Bert [19] model. Note that we feed an entire paragraph consisting of several sentences into Bert, leveraging high-level context. Our results show that replacing fixed word embeddings with the context-aware Bert features can significantly improve model performance over different architectures. Both models are relatively shallow (1 layer of attention / GRU respectively), which may be the reason why the deeper Bert model (13 layers) can help understand the text better.

**Impact of Alignment Losses.** We study the effect of alignment losses on the performance in Table 9. To give a more diverse picture, we evaluate the losses under different settings: We use three different low level pooling methods (Averagepool, Maxpool and Attention-aware Feature Aggregation) and selective disable the Cross-Modal Cycle Consistency loss and the global context attention in the Contextual Transformer.

We found that removing any or all of the three alignment losses significantly decreases performance. In addition, we observed that clustering losses have a positive impact on the performance of the model.

Note that we also tried clustering the global context and found it to be not helpful. It might be a too strong constraint on our low-level embedding network.

**Study on sequence pooling methods.** In Table 10 we replace our low-level (frames, words) and high-level (clips, sentences) pooling methods and evaluate the performance. Interestingly, we get the best results with our AF module on the low level, while averagepooling outperforms it on the high level. Removing our components (CMC, CoT) and replacing AF with maxpooling provides a considerably strong baseline compared to our full model.

The sequence length is higher on the low level (e.g. up to 80 frames) than on the high level (on average 3.6 clips per video). Additionally, there are stronger temporal relationships between semantics in the low level. The AF module can learn to capture these relationships and improve the features. However on the high-level, the semantics have more independent meanings which makes it much harder for AF to model the temporal relationships between them.

Note that to give a more fair comparison, we change the optimizer setting when adding AF on the high level, as denoted with *. We observe that concatenating the output of 2 AF modules on the high level improves the performance, suggesting that the two modules learn to attend to different features.

We also vary our approach on averagepooling on the high level and report results In Table 11. Working on variable length inputs, there are a number of design choices to make. We evaluate the following ones: *1)* Summing over the unmasked sequence elements (nonzero inputs) only or summing over both sequence and padding elements (zero inputs). *2)* Minimum padding to the maximum sequence length in the minibatch or to a length of at least 16. *3)* Obtaining the average by dividing the sum by the length of nonzero elements or by the length of all elements.

On **ActivityNet-captions** (split val1, average sequence length 3.6), we show our non-standard approach of including padding tokens in the sum but dividing by the length of non-padding tokens works well. Note that in all other reported experiments, we use this version of averagepooling.

On **Youcook2** (split val, average sequence length 7.6), we cannot reproduce this large gain in performance but the approach still works reasonably well. The good results when padding to a minimum length of 16 might be due to the average length being closer to 16 than in ActivityNet-captions.

### A.3 Retrieval on ActivityNet-captions (split val2)

We provide retrieval results for ActivityNet-Captions (val2 split) in Table 12.

### A.4 Qualitative Results

**ActivityNet-Captions.** To further check whether our COOT model can learn the semantic alignment of video and text well, we provide qualitative examples for the retrieval task on the ActivityNet-caption dataset (val1 split, 4917 video-paragraph pairs). Note that any spelling errors in the dataset are not corrected. As shown in Table 13 and Table 14, the model learns to semantically align the video and paragraph embeddings. Even for imperfect rankings, the model retrieves semantically similar items.

**YouCook2.** We also present a set of qualitative clip-to-sentence and sentence-to-clip retrieval examples for the YouCook2 dataset (val split, 3492 clip-sentence pairs, 457 video-paragraph pairs). Table 15 and Table 16 show several examples where we can reasonably retrieve similar semantics, even when the wrong object is recognized (Table 15-Right).

**t-SNE Visualization of Embeddings.** We project the video embeddings of Activitynet dataset to 2D space using t-SNE [91] and visualize each point with a sample frame from the video. As shown in Figure 5, the embeddings are clustered semantically around activities and videos with similar content are in close neighborhood.

### A.5 Captioning Results

To expand upon the qualitative captioning results, we provide evaluation on samples that are not cherry-picked for Youcook2 (val split) and ActivityNet (ae-val and ae-test split) in Tables 17, 18, 19.

Table 13: **Qualitative Results on Activitynet for Paragraph-to-Video Retrieval.** For each text query, we show some frames from the top three ranked videos together with the correct video. For clarification, we show video results with text. **Left:** The correct video has a high rank and all top results are very relevant to the query. **Right:** Even though the correct video is ranked low, the top videos are semantically similar to the text query.

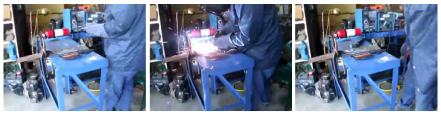
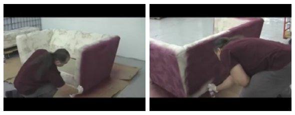
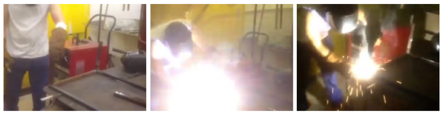
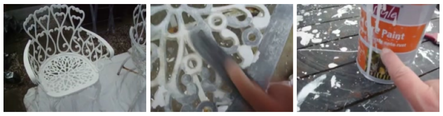
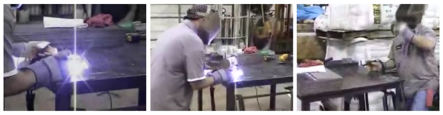
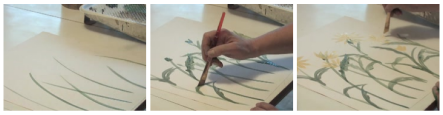
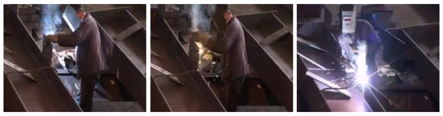
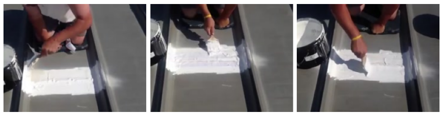
| *Query:* A man is standing inside a workshop. He leans over, welding a piece of metal. Sparks fly as he welds. | | *Query:* A person is kneeling down painting something on the ground. They smooth out the paint. They continue painting layers on top of the paint. | |
|---|---|---|---|
| Rank<br>*Score* | Retrieved Video | Rank<br>*Score* | Retrieved Video |
| **1**<br>*0.827* | <br>A man approaches a table with his soldiering gun.<br>The man soldiers on a piece of metal.<br>The man stops and walks away from his table. | **1**<br>*0.654* | <br>man is in a living room painting a couch with purle spray.<br>man paint the cushions of the couch on top of paperboard. |
| **2**<br>*0.821* | <br>A man is standing inside a workshop.<br>He leans over, welding a piece of metal.<br>Sparks fly as he welds. | **2**<br>*0.643* | <br>A white yard chair is being shown.<br>A man scrapes the paint off the chair with a scraper.<br>He shows off a bucket of paint, and uses the white paint to coat the chair. |
| **3**<br>*0.816* | <br>A man is welding something on a table.<br>He is moving his hand while he's welding.<br>He finishes welding and lifts up his mask. | **3**<br>*0.640* | <br>A woman is seated on a work table and holds a paint brush.<br>The woman paints a picture of long stems and leaves of a plant.<br>The woman paints flower petals onto the painting. |
| **4**<br>*0.783* | <br>A man in a brown shirt is standing in a room.<br>He is wearing a mask and welding something.<br>He stops welding and starts welding the back of it. | **48**<br>*0.438* | <br>A person is kneeling down painting something on the ground.<br>They smooth out the paint.<br>They continue painting layers on top of the paint. |

Table 14: **Retrieval Video to Paragraph on Activitynet.** Long paragraphs have been shortened, as indicated by "[...]". **Left:** The correct paragraph is identified with a considerable score margin to the 2nd place. **Right:** The top results are from the same activity as the input video (*dancing*).

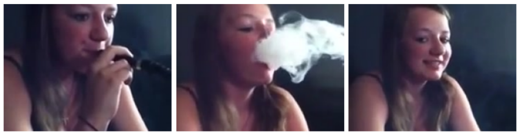
| *Query:*  | | *Query:* | |
|---|---|---|---|
| Rank<br>*Score* | Retrieved Text | Rank<br>*Score* | Retrieved Text |
| 1<br>*0.813* | **A woman is resting next to crashing water. She is smoking a pipe. She blows out a plume of smoke.** | **1**<br>*0.717* | A woman stands in front of a crowd of people on a public sidewalk and dances with a male dance partner in ballroom style dance. [. . . ] |
| **2**<br>*0.654* | A close up of a man's chin is shown followed by him smoking a hookah pipe. He takes the pipe out of his mouth and blows the smoke into the camera. | **2**<br>*0.678* | A woman in a leather dress and hat dances in a public station. A man joins her, dancing side to side in a flamenco style dance. They continue dancing as a small crowd gathers to watch. [. . . ] |
| **3**<br>*0.641* | A close up of tin foil is shown leading a woman taking a large hit out of a hookah hose. She continues smoking out of the hookah [. . . ] | **3**<br>*0.608* | A large group of people are seen standing around a city center waiting for people to arrive. Girls dancing are seen walking through the parade as other people watch on the side. [. . . ] |
| **4**<br>*0.601* | A woman is laying back in a chair getting her lip pierced. The piercer removes the tool and pulls on her lip. | 16<br>*0.496* | **People are dancing in a street. People are standing on the sidelines watching them. They continue dancing on a street.** |

Table 15: **Sentence-to-Clip Retrieval on Youcook2.** For clarification, we show clip results with corresponding text. **Left:** The model ranks the correct video at the top and even distinguishes it from other videos about the same activity. **Right:** The *slicing* task is correctly recognized, but the model is not able to understand which object is being chopped (*bamboo shots*).

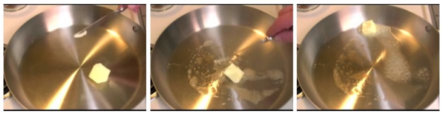
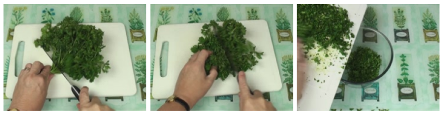
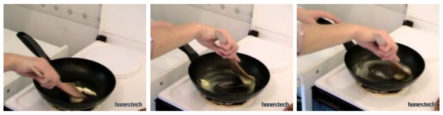
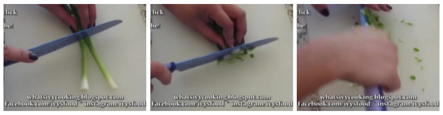
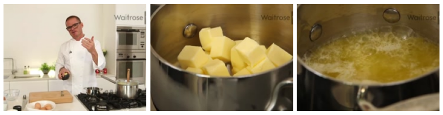
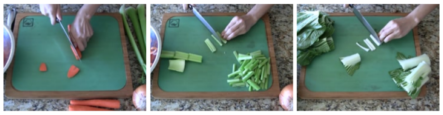
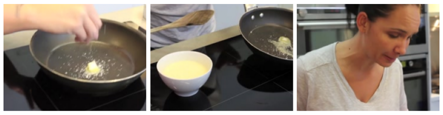
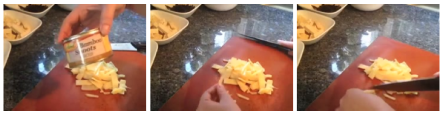
| *Query:* melt butter in the pan | | *Query:* slice the bamboo shoots into strips | |
|---|---|---|---|
| Rank<br>*Score* | Retrieved Clip | Rank<br>*Score* | Retrieved Clip |
| **1**<br>*0.642* | <br>melt butter in the pan | **1**<br>*0.621* | <br>finely chop a bundle of parsley and add to a bowl |
| **2**<br>*0.583* | <br>melt the butter in a pan | **2**<br>*0.610* | <br>finely chop green onions |
| **3**<br>*0.561* | <br>melt the butter in a pan | **3**<br>*0.609* | <br>cut garlic carrots celery onion and bok choi into thin slices |
| **4**<br>*0.553* | <br>heat the butter and some sea salt flakes in the pan | **168**<br>*0.326* | <br>slice the bamboo shoots into strips |

Table 16: **Clip-to-Sentence Retrieval on Youcook2 val set. Left:** The model gives high relative score to the relevant text but has problems visually distinguishing *apples* from *potatoes*. **Right:**: *Wine* is confused with *oil* and the herbs cannot be identified precisely to be *bay leaves* and *thyme*. Identical sentences can produce different results, since the Bert [19] text encoder takes paragraph context into account and therefore the model inputs differ.

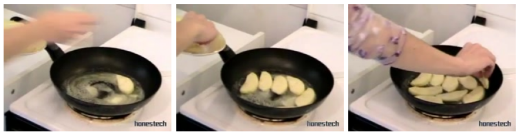
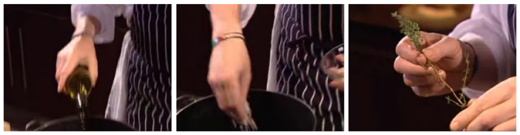
| *Query:*<br> | | *Query:*<br> | |
|---|---|---|---|
| Rank<br>*Score* | Retrieved Text | Rank<br>*Score* | Retrieved Text |
| **1**<br>*0.523* | place the potato wedges into a pan of hot oil | **1**<br>*0.705* | add oil and herbs to a pan |
| **2**<br>*0.514* | **cook the apple slices in the pan** | **2**<br>*0.622* | heat oil to 365 in a pan |
| **3**<br>*0.510* | remove the potatoes from the oil and place on paper towel | **3**<br>*0.603* | heat some oil in a pan |
| **4**<br>*0.497* | add oil to the pan and fry the hash browns | **4**<br>*0.579* | heat some oil in a pan |
| **5**<br>*0.495* | fry the potatos in oil | **5**<br>*0.575* | add oil to a pan |
| **6**<br>*0.480* | add the potatoes to the pan | **6**<br>*0.570* | heat some olive oil in a pan |
| **7**<br>*0.477* | heat the apple in a pan with some oil | **7**<br>*0.567* | heat some oil in a pan |
| **8**<br>*0.475* | pierce the knife inside the potatoes and find if the potatoes are cooked properly | **8**<br>*0.564* | heat oil in a pan |
| **9**<br>*0.474* | melt little butter and olive oil in a pan | **9**<br>*0.564* | heat some oil cumin seeds and coriander seeds in a pan |
| **10**<br>*0.470* | fry the potatoes in a deep fryer | **85**<br>*0.385* | **add white wine onions a bay leaf and thyme to the pot** |

Table 17: **Random Captioning samples on YouCook2 (val split).**

| | |
|---|---|
| <br>**MART**: Cook the bacon in a pan. Add chopped onions to the pan. Add chopped carrots. Add chopped tomatoes to the pan. Add the chicken to the pan.<br>**COOT (Ours)**: Fry the beef in a pan. Add onion and carrot to the pan. Add the chicken to the pan. Add the tomatoes and stir. Add the potatoes to the pan.<br>**GT**: Brown 400gm of sliced beef on a hot pan. Fry onions until golden then add garlic carrots and red pepper fry for 5 mins. Now add the beef 2 tbsp of flour 1 tsp of paprika 1 tbsp of tomato puree 2 bay leaves and 300ml beef stock. Add 200 gram canned tomato 100ml red wine sour cream and mix well let it simmer for 1 5 hour. Now add 400gm of baby potato and mix it let it cook for 30 more min. | <br>**MART**: Add flour to a bowl and whisk. Cut the chicken into pieces. Coat the chicken in flour. Coat the chicken in flour egg and breadcrumbs. Fry the chicken in a pan. Drizzle the sauce on top of the bread. Add sauce to the pizza. Bake the dish in the oven.<br>**COOT (Ours)**: Mix parmesan cheese black pepper and garlic powder. Cover the chicken in the bag. Coat the chicken in the flour. Coat the chicken in the egg and coat with flour. Place the chicken in a pan and fry it on a pan. Pour sauce on top of the chicken and top with mozzarella cheese. Sprinkle parmesan cheese on top. Bake the chicken in the oven.<br>**GT**: Mix bread crumbs and parmesan cheese. Pound the chicken. Rub salt and pepper onto the chicken. Rub flour onto the chicken dip it in egg and coat with breadcrumbs. Fry the chicken in a pan. Spread sauce over the chicken. Top the chicken with mozzarella cheese. Bake the chicken in the oven. |
| <br>**MART**: Add tomatoes and beef to a pot. Add water to the pan. Add tomato puree and salt. Add the beef and parsley to the soup. Add the beef to the pot. Add water to the soup and let it simmer. Add the soup to the soup.<br>**COOT (Ours)**: Add the tomatoes and onions to a food processor and blend them. Add the tomatoes and a bay leaf to the pot. Add the tomatoes and simmer. Remove the tomatoes from the pot and let it cook. Remove the tomatoes from the pot and let it cook. Strain the soup to a boil and let it boil. Turn on the heat and heat to a boil.<br>**GT**: Add tomato onion green chili and rice to a pan. Add water to the pan. Boil the ingredients and then turn down the heat. Strain the ingredients. Blend the ingredients. Add the water to the mixture and strain. Boil the soup. | <br>**MART**: Add flour to a bowl and whisk. Cut the chicken into pieces. Coat the chicken in flour. Coat the chicken in flour egg and breadcrumbs. Fry the chicken in a pan. Drizzle the sauce on top of the bread. Add sauce to the pizza. Bake the dish in the oven.<br>**COOT (Ours)**: Mix parmesan cheese black pepper and garlic powder. Cover the chicken in the bag. Coat the chicken in the flour. Coat the chicken in the egg and coat with flour. Place the chicken in a pan and fry it on a pan. Pour sauce on top of the chicken and top with mozzarella cheese. Sprinkle parmesan cheese on top. Bake the chicken in the oven.<br>**GT**: Mix bread crumbs and parmesan cheese. Pound the chicken. Rub salt and pepper onto the chicken. Rub flour onto the chicken dip it in egg and coat with breadcrumbs. Fry the chicken in a pan. Spread sauce over the chicken. Top the chicken with mozzarella cheese. Bake the chicken in the oven. |

Table 18: **Random Captioning samples on ActivityNet (ae-val split).**

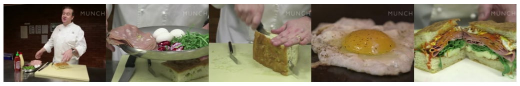
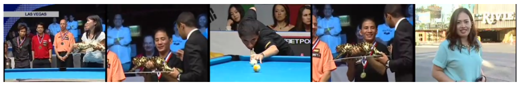
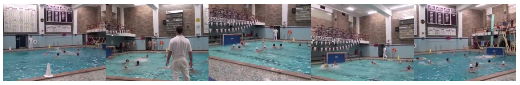
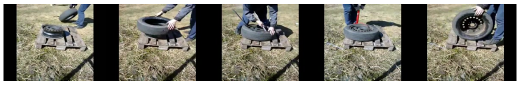
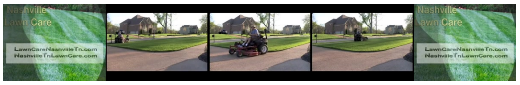
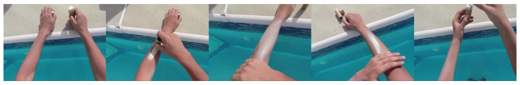
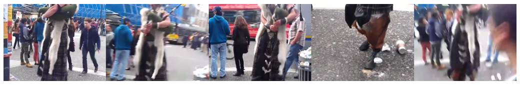
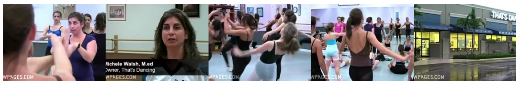
| | |
|---|---|
| <br>**MART**: A man is seen speaking to the camera and leads into him holding up various objects and presenting them to. He then cuts the knife and cuts the sandwich while still speaking to the camera. He then puts the sandwich into the pan and cuts it in half. He then puts the sandwich into the sandwich and puts it in the end.<br>**COOT (Ours)**: A chef demonstrates how to make a sandwich using bread , then he puts a knife in a kitchen and. Then , the man puts the bread on a bread and cuts it in half. After , the man puts the sandwich in the bread and put it in a plate. Next , the man cuts the bread and put on top of the sandwich .<br>**GT**: A man shows ingredients for a mortadella sandwich. The man cuts the bred in four pieces and puts mustard and then brown on the stove. Then, the man fries an egg and puts it on the bread as well the mortadella, green leaves, cheese and ketchup. After, the man cuts the sandwich in two and eat one. | <br>**MART**: A person is seen sitting in front of a large pile of grass and holding a stick. The person then puts the tire on the machine and begins putting the tire on.<br>**COOT (Ours)**: A person is seen using a tool on a machine and piecing together with the camera. The man continues to use the machine on the machine and ends by taking out more out of the machine.<br>**GT**: A person is seen walking in with a tire on a plank and painting the tire. The person then un does the tire and places the rubber tightly around the side. |
| <br>**MART**: A small group of people are seen swimming around a pool throwing a ball around to one another. The people continue playing with one another and end by throwing the ball back and fourth.<br>**COOT (Ours)**: A large group of people are seen swimming around a pool throwing a ball around to one another. The people continue playing with one another and ends with a large group of people watching on the sides.<br>**GT**: A video of water polo is shown in the gym. A few people watch and the ball goes back and forth. | <br>**MART**: A person is seen sitting in front of a large pile of grass and holding a stick. The person then puts the tire on the machine and begins putting the tire on.<br>**COOT (Ours)**: A person is seen using a tool on a machine and piecing together with the camera. The man continues to use the machine on the machine and ends by taking out more out of the machine.<br>**GT**: A person is seen walking in with a tire on a plank and painting the tire. The person then undoes the tire and places the rubber tightly around the side. |

Table 19: **Random Captioning samples on ActivityNet (ae-test split).**

| | |
|---|---|
|  Nashville Lawn Care<br>LawnCareNashvilleTn.com<br>NashvilleTnLawnCare.com<br>**MART**: A woman stands on front a house talking. The woman drives the lawn mower with a mower. The woman drives the lawn mower. The woman pushes the lawn mower along the grass. The woman talks to the camera.<br>**COOT (Ours)**: We see the title on the field , white and white text. We then see a man mowing his lawn. The man stops and talks to the camera. The man stops and turns around. We then see the grass again.<br>**GT**: The video begins with a picture of a lawn along with a company name and website. The video cuts to a man riding a lawnmower, cutting the grass in a nice neighborhood. When he begins, some kids are playing in the road. At one point, a car passes by. The video ends with the picture of the lawn showing the company name and website. | <br>**MART**: A group of women are dancing on a stage. They are dancing together in a room. They are dancing together.<br>**COOT (Ours)**: A large group of girls are seen standing together followed by a woman dancing and performing a dance routine. The woman continues speaking to the camera while more people are seen dancing around and leads into a group of. The group continues dancing with one another and ends with a woman speaking to the camera.<br>**GT**: Several girls are in a classroom dancing and doing ballet. The instructor then comes to talk briefly before continuing on coaching the girls. After,the exercises continue and the girls do leaps and jumps in the room before the outside of the dance studio is shown. |
| <br>**MART**: People are gathered around a street watching. They are holding flags in their hands. A man in a white shirt is standing next to a fence.<br>**COOT (Ours)**: A man plays bagpipes while people watch on the sidewalk. A person in a black shirt plays the bagpipes. A person in a white shirt walks past the person.<br>**GT**: A man on stilts is playing the bag pipes on a street. A bus passes on the street behind the man. A street sign on a pole is shown. |  Michele Walsh, M.ed<br>Owner, That's Dancing<br>WPAGES.COM<br>THAT'S DA<br>**MART**: A group of women are dancing on a stage. They are dancing together in a room. They are dancing together.<br>**COOT (Ours)**: A large group of girls are seen standing together followed by a woman dancing and performing a dance routine. The woman continues speaking to the camera while more people are seen dancing around and leads into a group of. The group continues dancing with one another and ends with a woman speaking to the camera.<br>**GT**: Several girls are in a classroom dancing and doing ballet. The instructor then comes to talk briefly before continuing on coaching the girls. After,the exercises continue and the girls do leaps and jumps in the room before the outside of the dance studio is shown. |

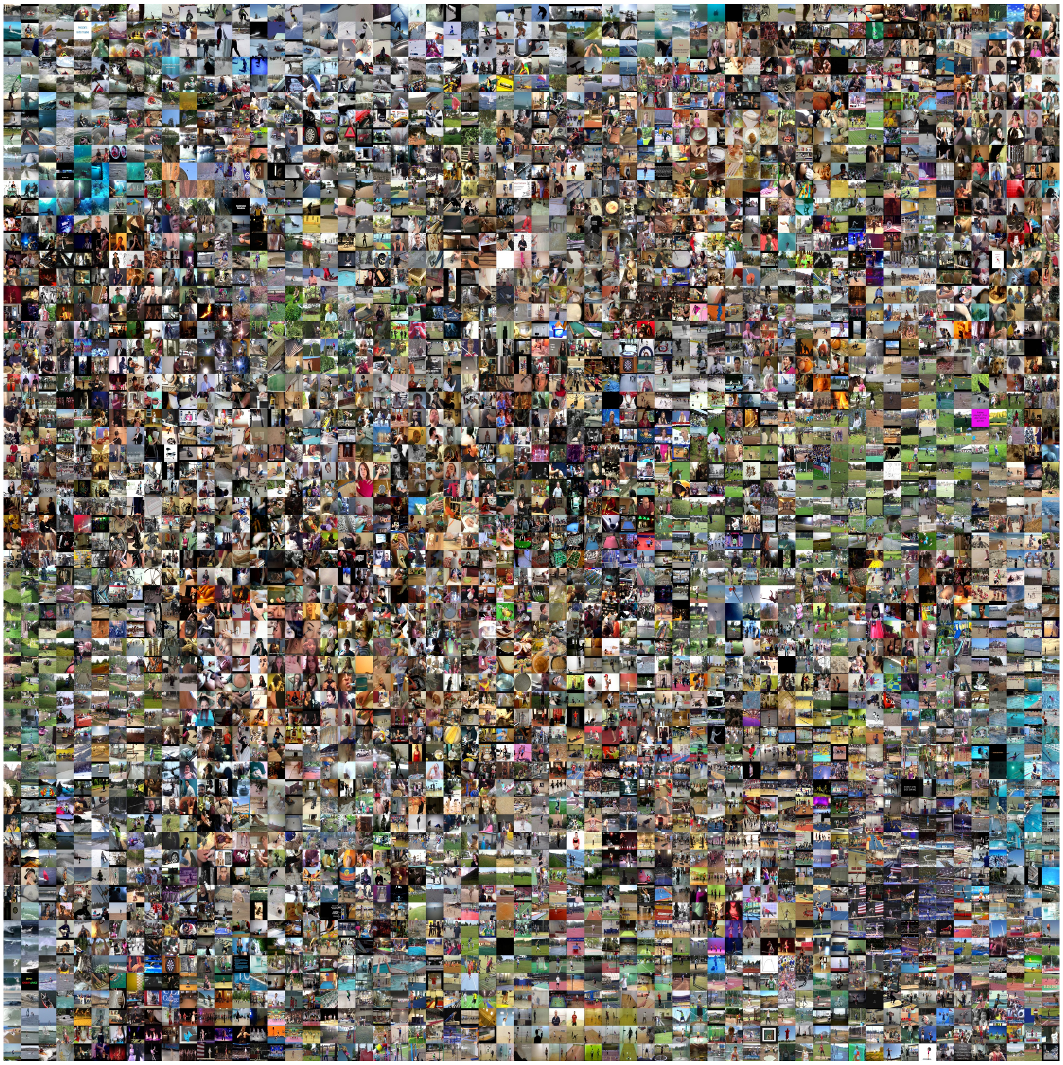


Figure 5: **Visualization of the video embedding space with t-SNE on ActivityNet-Captions.** We apply t-SNE to reduce the video embedding space to 2 dimensions and visualize videos by one sample frame.