Temporal Alignment Networks for Long-term Video

Tengda Han[1] Weidi Xie[1,2] Andrew Zisserman[1]
[1]Visual Geometry Group, University of Oxford [2]Shanghai Jiao Tong University
https://www.robots.ox.ac.uk/~vgg/research/tan/

Abstract

The objective of this paper is a temporal alignment network that ingests long term video sequences, and associated text sentences, in order to: (1) determine if a sentence is alignable with the video; and (2) if it is alignable, then determine its alignment. The challenge is to train such networks from large-scale datasets, such as HowTo100M, where the associated text sentences have significant noise, and are only weakly aligned when relevant.

Apart from proposing the alignment network, we also make four contributions: (i) we describe a novel co-training method that enables to denoise and train on raw instructional videos without using manual annotation, despite the considerable noise; (ii) to benchmark the alignment performance, we manually curate a 10-hour subset of HowTo100M, totalling 80 videos, with sparse temporal descriptions. Our proposed model, trained on HowTo100M, outperforms strong baselines (CLIP, MIL-NCE) on this alignment dataset by a significant margin; (iii) we apply the trained model in the zero-shot settings to multiple downstream video understanding tasks and achieve state-of-the-art results, including text-video retrieval on YouCook2, and weakly supervised video action segmentation on Breakfast-Action. (iv) we use the automatically-aligned HowTo100M annotations for end-to-end finetuning of the backbone model, and obtain improved performance on downstream action recognition tasks.

1. Introduction

The recent CLIP and ALIGN papers [30, 53] have demonstrated that a combination of large scale paired image-caption data, and a simple noise contrastive learning loss can be used to learn powerful image-text embeddings from scratch. The image-caption data can be crawled from the internet at scale, for example from image alt-text, and the resulting embeddings demonstrate strong "zero-shot" generalization abilities. In the video domain, there also exists large-scale sources of text supervision, *e.g.* narrated instructional videos such as the HowTo100M [47] dataset, where demonstrators explain their actions while performing a complex task. The narrations are unconstrained and can be combinatorially complex, including information on "what", "where" and "when", such as the actions, the objects, human-object interactions, *etc*.

However, these instructional videos pose additional fundamental challenges over the image-caption scenario due to the temporal *alignment* problem (illustrated in Figure 1): (i) the demonstrator often makes statements that are unrelated to the visual signal, such as describing food taste or explaining the consequence of actions. These texts are *not visually alignable*. (ii) the demonstrator might explain their action before or after performing it, and their statements often *do not follow the same order* as their actions, resulting in the text and visual entities being asynchronous. These texts are *not temporally aligned* to the visual signal. Additionally, unlike spatial segmentation in images, where objects boundaries are often formed by a discontinuity between regions with strong gradients, temporal actions in videos are often continuous, making it difficult to clearly define the start and end points for the temporal interval. Last but not the least, there is additional noise coming from the imperfect Automatic Speech Recognition (ASR) systems on the spoken narrations. Note that the image-caption data does not face these problems since captions are provided by human annotators for that image; although they may be incomplete, there is no temporal alignment issue.

The extent of these alignment challenges is significant [46, 47]. In 10 hours of instructional videos (sourced from HowTo100M) that we annotated for this work, only 30% of the narration sentences are visually alignable, and only 15% are naturally well-aligned. This means that the demonstrator is describing their action synchronously with the video only 15% of the time. If the alignment issues are resolved then the benefits of learning from such narrated instruction videos can potentially be substantial: with the extra time axis alignment, models can be trained to deal with fine-grained tasks, and predict temporal action localization and segmentation.

In this paper, we tackle the sentence-to-video temporal alignment problem, and propose a Temporal Alignment Network (TAN) that ingests a video sequence and its associated narrative sentences, attends to a large temporal con-

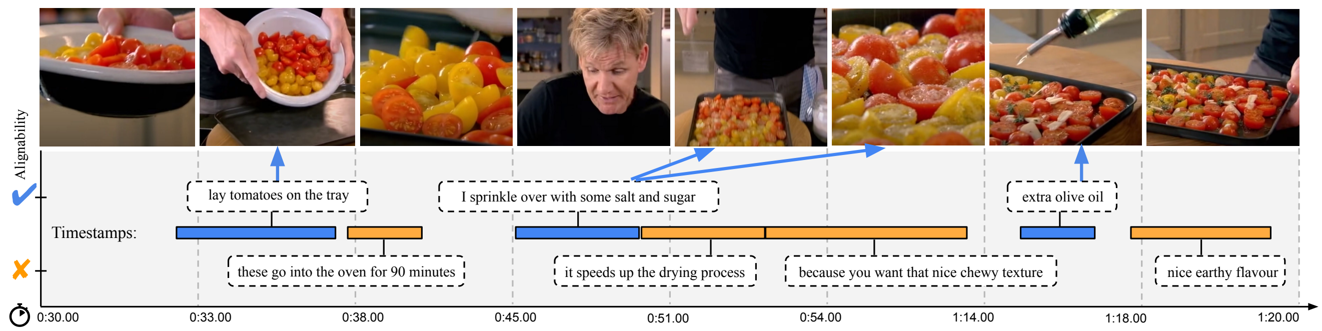

Figure 1: An example of visual-textual mis-alignment in a raw instructional video. The presenter's narration can be not visually relevant at all, *e.g.* describing a flavor; or asynchronous with visual content by a time difference. The ✓ and ✗ indicate visually alignable and non-alignable text, respectively (by human judgement). The colored bar shows the start-end timestamp of narration. Example from https://www.youtube.com/watch?v=M8OGXmLTTiI?t=30.

text in both, and is able to: (1) determine if a sentence is alignable with the video; and (2) if it is alignable, then determine its temporal alignment. Given all the challenges described above, training such a network on raw instructional videos, *e.g.*, HowTo100M, is clearly a non-trivial task. To this end, we propose a novel method for denoising, by co-training TAN with an auxiliary dual encoder network. By design, these two networks use complementary architectures: TAN iteratively attends to temporal context from both visual and textual modalities, establishing accurate alignment for sentences that are alignable; while the dual encoder processes visual and textual modalities independently, which enables it to spot unalignable sentences at ease, *e.g.*, sentences that emit low alignment score to all frames within the video. The output from these two networks can be treated as two different views for alignment, and their *mutual agreements* are adopted for co-training.

In addition to introducing the model and training methodology, we make the following contributions: (1) We manually annotate an 80-video subset of HowTo100M, named **HTM-Align**, by assigning the visually related sentences to their corresponding timestamps and annotating visually unrelated ones. This aligned subset is used to evaluate the model's performance and is released publicly; (2) We train the model on the HowTo100M dataset, and demonstrate a significant improvement in alignment over prior work (MIL-NCE approach of [46] in particular); (3) We apply the trained model in both the zero-shot and fine-tuned settings to multiple downstream video tasks and achieve state of the art results on both settings. This includes text-video retrieval on YouCook2 [75] and weakly supervised video action segmentation on Breakfast-Action [34]; (4) We use the automatically-aligned HowTo100M annotations to finetune the *backbone model*, and observe improved performance on downstream action classification tasks.

2. Related Work

Joint Visual-Textual Learning has a long history in computer vision. As examples, early work from Mori et al. [49] explored the connection between image and words in paired text documents, and [68] learnt a joint image-text embedding for the case of class name annotations. Recent works like CLIP [53] and ALIGN [30] show that large-scale paired image-caption data combined with a simple noise contrastive learning loss is able to learn a powerful visual representation. In video domains, this is also true, as shown by MIL-NCE [46], ALBEF [41], and VideoClip [71].

Visual-Textual Retrieval learns a joint embedding space for both vision and language, either using a dual encoder [2, 19, 23, 24, 30, 33, 47, 51–53], where visual and textual inputs are independently encoded, or a joint encoder, constructed with multimodal Transformers [13, 39, 43, 44, 62, 63, 74], where vision and text inputs are fed into the cross-modal attention to compute the similarity. Despite being more accurate, the incurred computation of the joint encoder limits its use for large-scale retrieval systems. In [45], the authors propose to speed up the process by only using the joint encoder for re-ranking. In this work, we also use both joint and dual encoders, but for a different purpose – to exploit their complementary information for co-training.

Visual-Textual Alignment aims to temporally assign words or sentences to the corresponding video segments. A similar task is weakly-supervised action segmentation that tries to delineate the video segments corresponding to a given action list [5, 6, 9, 18, 29, 36, 40, 54, 79]. In transcript alignment [15, 56, 57, 64, 78], where instead of an action list, scripts describing a series of events in the video are given, the goal is to assign each of the script texts to the appropriate segment (shot) of the video. More closely related to our goal are methods that seek a global alignment between sequences with soft Dynamic Time Warping (DTW) [16]. The recent Drop-DTW [20] proposes to handle outliers in the sequences by allowing the alignment process to automatically skip certain steps. This is similar to our aim of identifiying non-alignable sentences. However, since in HowTo100M the order of the alignable sentences does not follow the original order of the subtitles, this rules out the use of DTW-type approaches.

Co-training and Self-training are common techniques for unsupervised and weakly supervised learning. Co-training [4] builds two models to learn the different views of the data, while using one to expand the training set for the other. It has recently been used for representation learning [26, 65]. Self-training refers to the process of training on pseudo-labels generated from a model's own predictions. It has been used for image classification [1, 7, 8, 69], object detection [14], and machine translation [27]. Our work is related to this line of research, where the TAN and the auxiliary network self-correct the noisy annotations, such that both networks can gradually improve their performance by training on cleaner data.

Supervised Action Segmentation & Detection have been extensively studied on numerous video datasets, *e.g.* Breakfast-Action [34], YouCook2 [75], Charades [59], ActivityNet [28], EPIC-Kitchens [17]. For segmentation, the goal is to densely classify each time point of the video into one of the pre-defined action categories [3, 5, 12, 21, 22, 34, 37, 38, 55, 60]. Research has focused on designing effective modules to capture dependencies between different video chunks [21, 37, 38, 60]. For detection, the goal is to localize the sparsely distributed action segments, *i.e.* annotation is non-contiguous. In general, there are two-stage approaches that consist of a separate action proposal stage and a classification stage [11, 42, 58, 70, 73], and one-stage approaches that combine both [50, 72].

3. Method

We start by describing the problem scenario in Sec 3.1, followed by the architecture for our proposed alignment network in Sec 3.2. In Sec 3.3, we describe a naïve training procedure on raw instructional video, with the text-video correspondence provided by YouTube ASR, despite the considerable noise. In Sec 3.4, we present the co-training method, that exploits the *mutual agreement* between the alignment network and an auxiliary dual encoder, and is able to simultaneously denoise and learn from the noisy narrated instructional videos.

3.1. Problem Scenario

Given an untrimmed instructional video $\mathcal{X} = \{\mathcal{I}, \mathcal{S}\}$, where $\mathcal{I} = \{I_1, I_2, \dots, I_T\}$ refers to the corresponding video with T frames, and $\mathcal{S} = \{S_1, \dots, S_K\}$ denotes the K given sentences (ordered by time). For each sentence, we also have their timestamps obtained from YouTube ASR (*e.g.* $[t_k^{\text{start}}, t_k^{\text{end}}]$ for the k-th sentence). In this paper, our goal is to train a temporal alignment network on a video dataset of instructional videos, which takes the videos and sentences as inputs, and outputs a textual-visual similarity matrix ($\hat{\mathbb{A}}$), as well as an alignability score for each sentence:

$$\{\hat{y}, \hat{\mathbb{A}}\} = \Phi(\mathcal{X}; \Theta), \quad \hat{\mathbb{A}} \in \mathbb{R}^{K \times T} \tag{1}$$

where $\hat{y} \in \mathbb{R}^{K \times 2}$ refers to binary scores for all sentences, indicating whether the sentence is alignable. $\hat{\mathbb{A}}$ denotes the similarity matrix between frames and the given sentences, where for any alignable sentence it should emit a higher score with its corresponding video timestamps than others, and Θ are the parameters of the model.

3.2. Temporal Alignment Network (TAN)

As shown in Figure 2 (left), the alignment network takes a video sequence and its associated narration / text sentences as input, and attends to the long temporal contexts in both, in order to: (i) determine if a sentence is alignable with the video ($\hat{y}$), and (ii) output the alignment matrix ($\hat{\mathbb{A}}$). Next, we describe the alignment network, consisting of a visual-textual backbone, Multimodal Transformer, and alignability prediction module.

Visual-Textual Backbone. Given a long instructional video (*e.g.* 64s) with its associated sentences, we first extract the visual and textual features with pre-trained networks. Specifically, based on MIL-NCE [46], we use their pre-trained S3D-G backbone to extract video features, and a 2-layer MLP with the word2vec embeddings [48] to extract sentence features.

$$v = f(\mathcal{I}) \in \mathbb{R}^{T \times C} \qquad s = g(\mathcal{S}) \in \mathbb{R}^{K \times C} \tag{2}$$

v, s refer to the computed video and text features respectively, and each is of dimension C, in general, $T \gg K$.

Multimodal Transformer. This module jointly processes the visual-textual features (v, s) with a multi-layer Transformer Encoder, which iteratively attends to both modalities to establish the text-to-video correspondence:

$$[\hat{v}; \hat{s}] = \Phi_{\text{MT}}([v + \text{TE}; s]) \tag{3}$$

where Φ_{MT} refers to the Multimodal Transformer Encoder, TE denotes the learnable temporal embedding, $\hat{v} \in \mathbb{R}^{T \times C}$ and $\hat{s} \in \mathbb{R}^{K \times C}$ are the output visual and textual embeddings from the Multimodal Transformer, and the "[;]" symbol denotes concatenation. The alignment matrix $\hat{\mathbb{A}} \in \mathbb{R}^{K \times T}$ is computed via cosine similarity:

$$\hat{\mathbb{A}}_{[i,j]} = \frac{\hat{s}_i \cdot \hat{v}_j}{\|\hat{s}_i\| \|\hat{v}_j\|} \tag{4}$$

Alignability Prediction Module. Apart from estimating the alignment matrix, another main functionality of the alignment network is to infer whether a particular sentence is alignable or not. This is achieved by training a single linear layer ($\phi(\cdot)$) on the textual features, as shown in Figure 2 (left):

$$\hat{y} = \phi_{\text{align}}(\hat{s}) \tag{5}$$

where $\hat{y} \in \mathbb{R}^{K \times 2}$ refers to the binary predictions for all sentences, deciding if the sentence is alignable or not.

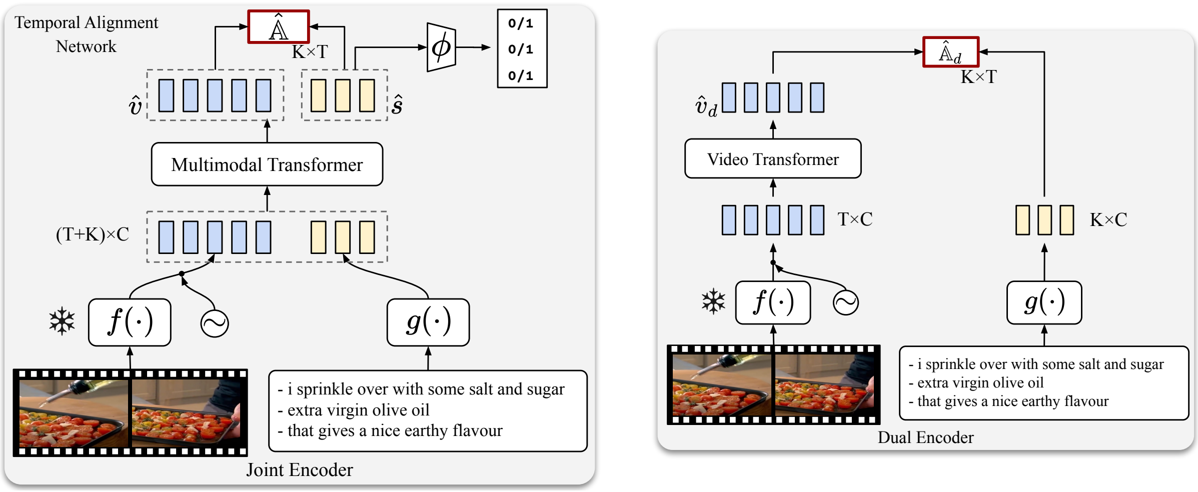

Figure 2: *Left*: The Temporal Alignment Network (TAN) takes an untrimmed long video as input, and first extracts the visual and textual features by a pre-trained 3D ConvNet ($f(.)$) and a pre-trained text module ($g(.)$). The visual features and textual features are concatenated and passed into a Multimodal Transformer Encoder, a.k.a. *joint encoder*, where the attention can capture the interaction between the visual and textual modalities. A linear head ϕ classifies the alignability of the output text embedding. *Right*: To train the TAN on noisy instructional videos, we build an auxiliary *dual encoder*, which takes the same visual and textual features as input, but only use a Video Transformer Encoder to process the video data with self-attention. For both TAN and the dual encoder, similarity matrices $\hat{\mathbb{A}}$, $\hat{\mathbb{A}}_d$ are computed between the output text features and the output visual features respectively, which are used at the co-training stage, as introduced in Section 3.4.

3.3. Training

In this section, we describe a naïve training procedure for the alignment network with contrastive learning, on the instructional videos with YouTube ASR timestamps. Note that, at this stage, all the sentences have their corresponding video timestamps, and are treated as alignable. Hence, the alignability prediction module can not be trained here.

Temporal Correspondence. For a video with K sentences, we directly convert its YouTube ASR results into 1D binary masks, with 1's at the timestamps where the sentence is being spoken by the demonstrator, *i.e.*, $\mathcal{Y} = \{m_1, \dots, m_K\}$, where $m_i \in \mathbb{R}^{1\times T}$. The objective is therefore to jointly optimize the visual-textual embedding, such that the similarity score between the sentence and its corresponding visual frames is maximised. The training objective is constructed as:

$$\mathcal{L}_{\text{TC}} = -\sum_{k=1}^{K} \log \frac{\sum_{i\in\mathcal{P}_k} \exp\left(\hat{\mathbb{A}}_{[k,i]}/\tau\right)}{\sum_{i\in\mathcal{P}_k} \exp\left(\hat{\mathbb{A}}_{[k,i]}/\tau\right) + \sum_{j\in\mathcal{N}_k} \exp\left(\hat{\mathbb{A}}_{[k,j]}/\tau\right)} \tag{6}$$

where $\mathcal{P}_k \in \{m_k = 1\}$, $\mathcal{N}_k \in \{m_k = 0\}$ refer to the sets consisting of positive and negative pairs, respectively. $\mathcal{L}_{\text{TC}}$ resembles a variant of the InfoNCE loss [66].

Discussion. Given the groundtruth annotation for alignment, optimizing $\mathcal{L}_{\text{TC}}$ would be trivial. However, on raw instructional videos where the provided YouTube ASR timestamps are highly unreliable with an extremely high noise ratio, naïvely optimising $\mathcal{L}_{\text{TC}}$ leads to sub-optimal results, as will be demonstrated in Section 5.2.

In general, the noise sources from the raw instructional videos can be mainly categorised into three types, as shown in Figure 1: *First*, the majority of the given sentences are actually not correlated to the video content (unalignable), *e.g.* greeting, chatting; *Second*, there is an alignment offset, in that the temporal interval of the spoken sentence rarely aligns with the video segments it refers to; *Third*, the demonstrator often makes statements that do not follow the same order as their action, which rules out the use of DTW-type approaches.

3.4. Co-Training

In this section, we propose a novel *co-training* method to both denoise the instructional videos and train the alignment network. Specifically, we introduce a dual encoder (Section 3.4.1), which can be seen as a collaborator to the alignment network. This procedure is detailed below.

3.4.1 Dual Encoder

As shown in Figure 2 (right), the dual encoder independently processes the visual features with a Transformer Encoder [67]. It is designed to be complementary to the alignment network: for example, the dual encoder is fast and lightweight, which enables training on a large number of visual-text pairs, however, it only allows both modalities to communicate at the end, hence is unable to capture the textual contexts, and it is more sensitive to detect unaligned texts; while the proposed TAN, consisting of a Multimodal Transformer, always has access to both modalities, and can learn to establish visual-textual correspondence within the network. Despite being beneficial for the temporal alignment task, the TAN is slow and computationally demanding, limiting its ability for contrasting with large-scale and diverse negative visual-textual pairs. Formally, for the dual encoder, we have:

$$\hat{v}^d = \Phi_{\text{D}}(v + \text{TE}) \in \mathbb{R}^{T\times C} \tag{7}$$

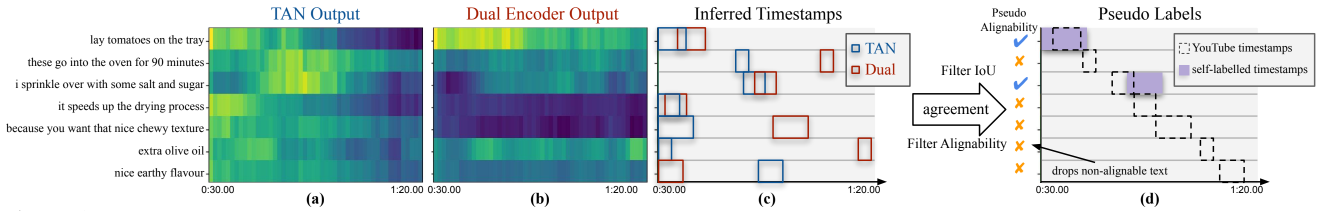

Figure 3: Illustration of denoising by mutual agreement. The video sample is the same as in Figure 1. **(a)**: The alignment matrix $\mathbb{A}$ from the TAN after Stage-1 training. **(b)**: The alignment matrix $\mathbb{A}_d$ from the dual encoder after Stage-1 training. **(c)**: The most alignable timestamps are inferred from both alignment matrices. **(d)**: By filtering the IoU of pseudo-timestamps and filtering the alignable/non-alignable text denoted by ✓ / ✗, the model dynamically chooses aligned temporal segments to train, and ignores non-alignable ones. For this example, the self-labelling process corrects the timestamps of the 1st and 3rd sentences, and marks the rest of the sentences as non-alignable. It roughly matches the human judgement of the alignment as shown in Figure 1. The alignment matrix values shown here are computed from the trained model-A in Table 1.

where Φ_{D} refers to the Video Transformer, and TE is the learnable temporal embedding for supplying the temporal ordering. A textual-visual cosine similarity matrix $\hat{\mathbb{A}}_d \in \mathbb{R}^{K\times T}$ from the dual encoder is computed as:

$$\hat{\mathbb{A}}_{d[i,j]} = \frac{s_i \cdot \hat{v}_j^d}{\|s_i\|\|\hat{v}_j^d\|} \tag{8}$$

3.4.2 Denoising by Mutual Agreement

To denoise the YouTube ASR annotations, we generate pseudo-labels (both the alignability and the timestamps) by verifying the *mutual agreement* between the output alignment matrices from the alignment network and dual encoder, *i.e.* $\hat{\mathbb{A}}$ and $\hat{\mathbb{A}}_d$. The verification process is executed in three steps:

(a) Infer Timestamps. During training, for each sentence, we use the two output alignment matrices $\hat{\mathbb{A}}, \hat{\mathbb{A}}_d \in \mathbb{R}^{K\times T}$ (Figure 3-a,b) to infer the most plausible aligned timestamps. To avoid outlier points, for the k-th sentence, we scan its corresponding similarity row by averaging the scores within a temporal window, this window is of the same size as its original YouTube timestamp label, *i.e* sentence by the demonstrator.

Afterwards, we pick the most confident prediction by taking the argmax. Note that, such operation ends up with a single temporal window with the same duration as the YouTube timestamp. That is to say, we only shift the temporal position of the original YouTube label to its most confident prediction. At this step, for the k-th sentence, we obtain two 'shifted' timestamps $\hat{m}_k$ and $\hat{m}_k^d$, one from the alignment network, the other from the dual encoder respectively, as shown in Figure 3-c.

(b) Alignment overlap using IoU. Given the inferred timestamps for sentence k, we compute an Intersection-over-Union (IoU) score to measure the agreement between the shifted timestamps:

$$\text{IoU-score}_k = \frac{\hat{m}_k \cap \hat{m}_k^d}{\hat{m}_k \cup \hat{m}_k^d} \tag{9}$$

A high IoU score indicates the sentence is very likely to be aligned with the inferred timestamps. For a batch, we filter the sentences with a positive IoU score, and update their timestamps by the union of their inferred timestamps $\hat{m}_k \cup \hat{m}_k^d$. Empirically, we find this operation roughly updates the timestamps for about 30% of the sentences. For the sentences with zero IoU score, we keep their YouTube timestamps unchanged. Such an operation ends up with a set of updated timestamps $\{\hat{m}'_1, \dots, \hat{m}'_K\}$ for all sentences.

In addition, to reflect each sentence's alignability, we can compute an average cosine similarity score falling into the new temporal segment. Formally, for the k-th sentence,

$$\epsilon_k = \frac{1}{\sum \hat{m}'_k} \sum \hat{m}'_k \cdot (\hat{\mathbb{A}} + \hat{\mathbb{A}}_d)_{[k,:]} \tag{10}$$

where ϵ_k refers to the alignment score. To put it simply, if a sentence has positive IoU-score, we compute its align-score within the union of inferred timestamps; if it has zero IoU-score, we compute its align-score within its original YouTube timestamps.

(c) Filter Alignability. To filter the alignability scores, *i.e.*, $\{\epsilon_1, \dots, \epsilon_K\}$, we introduce a hyper-parameter $\alpha \in [0, 1]$, within a sample batch, we treat the sentences with the top $100\alpha\%$ of align-score as positive, and the bottom $100(1-\alpha)\%$ sentences as negative. This gives binary pseudo-labels for alignability, denoted as y_{pseudo}. The alignability prediction module can thus be trained for binary classification with a cross-entropy loss (as shown in Figure 2), *i.e.*, $\hat{\mathcal{L}}_{\text{Alignability}} = \text{CE}(\hat{y}, y_{\text{pseudo}})$.

Intuitively, this is to say, a sentence is treated as being alignable if both the alignment network and the dual encoder agree the sentence has a high similarity with its corresponding time stamps. Also the $\mathcal{L}_{\text{TC}}$ (Equation 6) is only trained for the top $100\alpha\%$ of the sentences. In our experiments, we sweep $\alpha \in \{0.25, 0.5, 0.75\}$.

3.4.3 Training Cycle

To summarize, the training can be divided into two stages. At the first stage (S1: Initialization), both the alignment network and dual encoder are trained with $\mathcal{L}_{\text{TC}}$ using the given YouTube timestamps as labels. Once warmed up, the new pseudo-labels will be generated from the mutual agreement between alignment network and dual encoder *on the fly*, and starts the second stage training (S2: Co-Training), with $\mathcal{L}_{\text{total}} = \hat{\mathcal{L}}_{\text{TC}} + \hat{\mathcal{L}}_{\text{Alignability}}$. Note that it is not necessary to iterate S1 and S2, because in S2 the quality of pseudo-labels can be improved along the training with an

EMA mechanism (introduced next). By default in our experiments, we train S1 for 50k iterations, and train S2 for another 50k iterations. It accounts for 8 epochs on HTM-370K per stage.

3.4.4 Self-labelling with EMA

Naïvely using the *mutual agreement* between the alignment network and dual encoder for co-training can lead to trivial solutions, where the alignment network and the dual encoder learn to "collaborate" with each other and assign high similarity scores to certain fixed timestamps. We avoid this 'collapse problem' by keeping an Exponential Moving Average (EMA) of the model similar to BYOL [25]. The EMA branch is only slowly updated and used to generate the agreements for denoising as introduced in Section 3.4.2. The main branch is trained with the updated timestamps and alignability. We use the same momentum coefficient as that from BYOL in our experiments (0.99). By default, all evaluations use the main branch.

4. Experiments

In this paper, we train the proposed temporal alignment network on a subset of the HowTo100M dataset [47]. To start, we first describe the data preparation process, and present the annotated visual-textual aligned subset of HowTo100M (named **HTM-Align**) for evaluation. Then we describe the implementation details and ablation studies for the alignment task.

4.1. Data Preparation

HowTo100M is a large-scale instructional video dataset crawled from YouTube, consisting of around 1.2M videos and their generated text transcripts from speech (ASR). The start-end timestamps of each sentence are provided by ASR, but they are often *not semantically aligned* with the visual scene (Figure 1).

4.1.1 HTM-370K (Training)

We mostly use a subset of the original HowTo100M for training, with 370K videos from the 'Food & Entertaining' categories, consisting of 32% of the videos of the entire HowTo100M dataset. Apart from the mis-alignment issue, we also find three other issues in the subtitles: incorrect language translation, duplicated text, and incomplete sentence fractions. As dataset pre-processing, we conduct an automatic curation with open-sourced BERT-based model. The full details of automatic curation can be found in the Appendix.

After automatically processing and filtering out low-quality subtitles, we end up with a subset of 370K instructional videos, thus the name **HTM-370K**. Note that all the cleaning steps are automatic, using models trained with self-supervised learning. We attribute the pre-processing of HowTo100M as a small contribution, and we will make all cleaned video IDs and subtitles publicly available.

4.1.2 HTM-Align (Evaluation)

We randomly pick 80 videos from the HTM-370K as a hold-out testing set for evaluation purpose. These videos range from 2 to 16 minutes, totalling 10 hours. We manually label the alignability for each sentence, *i.e.* binary annotation. For those alignable ones, we further align them to the video segments with start-end timestamps. In total, 49K sentences are manually examined, with 13K of them being manually aligned. On average each video contains 61 sentences, and 17 of them are visually aligned.

Unlike the existing YouCook2 benchmark, where annotators only rephrase fixed recipe steps as the action description, **HTM-Align** includes random instructional videos without a fixed recipe, and are adopted from the demonstrators' narration with minor modification, hence containing large diversity on both videos and texts. The details of the annotation and examples can be found in the Appendix.

4.2. Implementation Details

During training, we adopt a pre-trained S3D (released by [46]) as the video encoder. Specifically, the S3D network outputs a single feature vector (1024D) for every 16 frames, when the videos are decoded with 16fps, this accounts for 1 feature per second without temporal overlap. For the text encoder, we use Bag-of-word (BoW) based on Word2Vec embeddings. By default, in each video we randomly sample a temporal window of 64 seconds (which is 64 continuous visual features, we also tried 32s and 128s in ablation study), and the corresponding subtitles within this window. We train the model with AdamW optimizer and 10^{-4} learning rate, with a batch size of 64 videos. Full implementation details are in the Appendix.

5. Alignment Results

In this section, we report the experimental results for our proposed temporal visual-textual alignment task on the **HTM-Align** dataset. In detail, during inference, given the video with a sequence of sentences by the demonstrator, we take the alignment matrix from our alignment network, $\hat{\mathbb{A}} \in \mathbb{R}^{K \times T}$, with K, T indicating the number of sentences and video timestamps respectively.

5.1. Metrics

We measure two metrics for the alignment task, Recall@1 and ROC-AUC value. The **Recall@1** metric is a 'pointing game' as introduced in [79]. Specifically, for a considered sentence, if its maximally matched video timestamp falls into the groundtruth segment, it is counted as being successfully recalled. The recall scores are then averaged across all the text segments. The alignability prediction is a binary classification problem as introduced in section 3.1, we use ROC curve and report the **Area-Under-the-Curve** value (ROC-AUC).

	Basic Setting			Training Stages		Stage2 Settings	Aligned-HTM	
model	dataset	length (# sec)	# tfm layers	S1:Init	S2:Self	Threshold α	R@1↑	ROC-AUC↑
CLIP (ViT-B/32) [53]	YFCC-400M	–	–	–	–	–	16.8	71.7*
MIL-NCE [46]	HTM-Full (uncurated)	–	–	–	–	–	31.3	73.1*
A	HTM-370K	64	6-6	✓	✗	–	45.8	73.0*
B	HTM-370K	64	3-3	✓	✗	–	42.3	72.6*
C	HTM-370K	64	6-6	✓	✓	0.25	42.5	79.7
D	HTM-370K	64	6-6	✓	✓	0.5	**49.4**	82.4
E	HTM-370K	64	6-6	✓	✓	0.75	48.8	82.2
F	HTM-370K	32	6-6	✓	✓	0.5	41.1	77.5
G	HTM-370K	128	6-6	✓	✓	0.5	48.4	81.8
H	HTM-Full	64	6-6	✓	✓	0.5	49.2	**82.6**

Table 1: **Alignment results on the HTM-Align dataset.** *: since the model does not have a binary classifier for alignability, for each sentence, we take its maximum logits over time as the alignability measurement to compute ROC-AUC. For the '# tfm layers' column, we show the number of transformer encoder layers we use for the TAN and the dual encoder.

5.2. Ablation Study

In this section, we investigate the effects of multiple design choices and discuss the results.

Comparing with baseline. In Table 1, the first two rows are the baselines from CLIP (ViT-B/32) [53] and MIL-NCE [46]. Specifically, we compute the alignment similarity matrix using their textual and visual encoders, normalize the score following their pretrained paradigms, and compute the R@1 on top of the alignment matrix. Note that for ROC-AUC, since CLIP and MIL-NCE do not have a specific binary classifier, for each text, we directly use its maximum similarity score (across the time axis) as an indicator of alignability. First, CLIP [53] is performing significantly worse than others on this alignment task. A possible reason is that CLIP has only been trained on images, thus lacks video dynamics. MIL-NCE is a strong baseline which has short-term temporal modelling (up to 3.2s) and was trained end-to-end on HowTo100M. In our model-A, we take the pre-extracted visual and textual feature from MIL-NCE, and train the transformers on the HTM-370K dataset to learn a longer temporal context (*e.g.* 64s) for the alignment task. Our result (model-A 45.8 R@1 vs MIL-NCE 31.3 R@1) shows that longer temporal context is useful for this alignment task.

Effect of Transformer Depth. For both the alignment network and dual encoder, we use 6-layer transformers by default, as a balance between performance and training cost. In model-B we also tried using 3-layer transformer and found it performs worse than 6-layer transformer (model-B vs A). Using more than 6 layers takes more memory and sacrifices batch size.

Effect of Co-Training. In the model-{D,E}, we apply the Stage-2 training (co-training) based on the model-A. We observe that co-training brings a clear performance gain for the alignment task (model-{D,E} vs. model-A, 3-4% boost on R@1), confirming the effectiveness of the denoising procedure. Note that model-C does not perform well due to the choice of alignability threshold α, explained next.

Effect of Alignability Thresholds. For the choice of alignability threshold α (as introduced in Section 3.4), which reflects a balance of data noise and diversity in the co-training procedure, our model-{C,D,E} show $\alpha = 0.5$ and $\alpha = 0.75$ work similarly well for alignment metrics and $\alpha = 0.5$ is slightly better for the R@1 metric. However $\alpha = 0.25$ leads to much worse performance. We conjecture that a low value of α limits the diversity while training $\mathcal{L}_{\text{TC}}$ (*i.e.* $\mathcal{L}_{\text{TC}}$ learns from only 25% of the sentences).

Effect of Training Data. In model-H, we train the co-training stage on the automatically curated HTM-Full dataset, which includes all other non-cooking categories from HowTo100M comparing with HTM-370K. Comparing model-H with D on the alignment task, adding out-of-domain videos does not harm the alignment performance on our curated subset.

Effect of Input Video Length. In Table 1, we vary the length of the input video to show if our alignment network benefits from the longer video context. Indeed, the alignment network gets better performance when increasing the input video length from 32 to 64 seconds (model-D vs model-F). We conjecture that sampling longer input video introduces more *alignable* sentences, helps to reduce the temporal ambiguity for other sentences. However, further increasing the input video length to 128 seconds gives a similar alignment performance (model-G vs model-D), we conjecture this is due the reduced batch size in training, and the far-away visual context (*i.e.* 2 minutes or further) is less relevant for aligning the sentence.

6. Downstream Tasks

Apart from evaluating the alignment task on **HTM-Align**, we also test our alignment network on other downstream tasks. Specifically, we evaluate on temporal action alignment (using the alignment network) and text-based video retrieval (using the dual encoder due to speed considerations [45]). We also evaluate linear action classification on the backbone feature to show the effect of auto-aligned dataset. See the Appendix for full details.

Method	Trained on BF	F-Acc↑	IoU↑	IoD↑
MIL-NCE [46]	✗ (ZS)	59.3	46.8	65.1
Ours (S1+S2)	✗ (ZS)	**65.1**	**50.6**	**68.6**
D^3TW [9]	✓	57.0	-	56.3
CDFL [40]	✓	63.0	45.8	63.9
DP-DTW [10]	✓	67.7	50.8	66.5
Ours (S1+S2)	✓	**68.3**	**51.7**	**69.3**

Table 2: **Temporal alignment on the Breakfast-Action (BF) dataset.** We split the previous methods into two groups. For the upper group, the model has not seen any samples in Breakfast-Action dataset since Breakfast-Action videos are not download-able from YouTube. For the lower group, the model is trained with weak supervision on the Breakfast-Action training set.

Method	Trained on YC2	R@1↑	R@5↑	R@10↑	Median R↓
ActBERT [77]	✗ (ZS)	9.6	26.7	38.0	19
MIL-NCE [46]	✗ (ZS)	15.1	38.0	51.2	10
MIL-NCE [46]†	✗ (ZS)	13.9	36.3	48.9	11
TaCo [31]	✗ (ZS)	19.9	43.2	55.7	8.0
Ours (S1)	✗ (ZS)	16.8	41.3	54.8	8.0
Ours (S1 + S2)	✗ (ZS)	**20.1**	**45.5**	**59.5**	**7.0**

Table 3: **Text-based video retrieval on the YouCook2 (YC2) dataset.** ZS refers to "zero-shot", where the alignment network is only trained on HTM-180K, and directly evaluated on YouCook2. †: reproduced in [76]. For our results, **S1** denotes only training Stage-1 (initialization), which is the model-A from Table 1. **S1+S2** denotes training with two stages (initialization followed by co-training), which is the model-E from Table 1.

Datasets. To evaluate the alignment network, we use *Breakfast-Action* [34] and *YouCook2* [75] for downstream tasks. To evaluate the end-to-end representation learning, we use *UCF101* [61], *HMDB51* [35] and *K400* [32].

Temporal Alignment on Breakfast-Action. Given a video with multiple actions and the corresponding action descriptions, the model needs to densely label each video timestamp with one given text description, often defined as weakly-supervised action segmentation by the community. Following previous work [9, 10, 18, 40], we report three metrics: frame-wise accuracy (**F-Acc**), segment-wise Intersection-over-Union (**IoU**) and Intersection-over-Detection (**IoD**). Please refer to Appendix for more details.

We evaluate our method with both the zero-shot and finetune settings. In the former case, our alignment network was *only* trained on HTM-370K, and directly evaluated on Breakfast; while in the latter, we finetune our alignment network on Breakfast with a soft-DTW loss [16] stacked on top of the output alignment matrix for 50 epochs. During inference, the alignment network takes as input a single video and the given list of action labels, *i.e*. 'crack egg', 'fry egg', *etc*, and outputs the alignment matrix $\mathbb{A}$, which is passed through a DTW, ending up with the action boundaries.

As shown in Table 2, in the zero-shot setting, our proposed alignment network surpasses the strong baseline (MIL-NCE) by a large margin on all metrics ($> 3\%$), and even achieves comparable results to those supervised approaches. After finetuning, we see a further performance boost, obtaining state-of-the-art results.

Text-based Video Retrieval on YouCook2. We evaluate the model for text-based video retrieval on the YouCook2 dataset. For this task, we pass each pre-cropped video segment through the *dual encoder*, and take the visual features (v_{enc}) from the Video Transformer Encoder. Also we pass the task description phrases into the dual encoder and take the word2vec features. For each query text, we rank the video segments based on cosine similarity among 3.5k candidates. Following previous works [46, 47], we report retrieval Recall @$\{$1,5,10$\}$ and Median Rank.

As shown in Table 3, under the zero-shot setting, where the proposed alignment network was only trained on HTM-370K, our model surpasses previous works by a clear margin, especially on R@5, R@10 and Median R. Importantly, the results show that the co-training stage substantially improves the performance of the *dual encoder* (R@1 20.1 vs 16.8), also our method surpasses the baseline method MIL-NCE by a large margin (R@1 20.1 vs 15.1).

End-to-end Representation Learning. The output of the Temporal Alignment Network can be used to clean-up (automatically align) long-video datasets. We use model-H to automatically align the HTM dataset, and finetune the S3D-word2vec backbone **end-to-end** with an Info-NCE loss on the auto-aligned text-video pairs for only 10 epochs. We evaluate the visual representation by linear probing on action classification, and find the auto-aligned HTM timestamps benefits the end-to-end video representation. We refer the readers for more details in the Appendix.

Settings	Backbone	UCF101	HMDB51	K400
reported by [46]	S3D	82.7	53.1	-
reproduce of [46]	S3D	82.1	55.2	55.7
finetuned with TAN	S3D	**83.2**	**56.7**	**56.2**

Table 4: **Linear-probing action classification performance.** We evaluate the end-to-end trained visual representations on UCF101, HMDB51 and K400 by linear probing (LP). We show the reported LP results from [46] (1st row), our reproduction of LP results of the official S3D weights (2nd row), and our finetuned S3D performance with auto-aligned HTM under the exact same setting (3rd row).

7. Conclusion

In summary, we introduce a temporal alignment network, with a co-training method for denoising the instructional video datasets. To evaluate the alignment accuracy we introduce a new benchmark dataset with 10 hours of videos, with the narrations manually aligned to corresponding video timestamp. When evaluating on **HTM-Align**, Breakfast-Action, YouCook2, under zero-shot or finetune settings, our model achieves state of the art results, surpassing multiple strong baselines (MIL-NCE, CLIP). We also show the proposed method can clean-up (by improving the alignment) large-scale public datasets and further improve the visual-textual backbone representations.

Acknowledgement This research is funded by EPSRC Programme Grant VisualAI EP/T028572/1, a Royal Society Research Professorship RP\R1\191132, and a Google-DeepMind Scholarship.

References

[1] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *Proc. ICLR*, 2020. 3

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proc. ICCV*, 2021. 2

[3] Subhabrata Bhattacharya, Mahdi M. Kalayeh, Rahul Sukthankar, and Mubarak Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *Proc. CVPR*, 2014. 3

[4] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998. 3

[5] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *Proc. ECCV*, 2014. 2, 3

[6] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *Proc. ICCV*, 2015. 2

[7] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proc. ECCV*, 2018. 3

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 3

[9] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D^3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proc. CVPR*, 2019. 2, 8

[10] Xiaobin Chang, Frederick Tung, and Greg Mori. Learning discriminative prototypes with dynamic time warping. In *Proc. CVPR*, 2021. 8

[11] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proc. CVPR*, 2018. 3

[12] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *Proc. CVPR*, 2020. 3

[13] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Yu Cheng Zhe Gan, and Jingjing Liu. Uniter: Learning universal image-text representations. In *Proc. ECCV*, 2020. 2

[14] Ramazan Gokberk Cinbis, Jakob J. Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. In *IEEE PAMI*, 2015. 3

[15] Timothee Cour, Chris Jordan, Eleni Miltsakaki, and Ben Taskar. Movie/script: Alignment and parsing of video and text transcription. In *Proc. ECCV*, 2008. 2

[16] Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In *Proc. ICML*, 2017. 2, 8

[17] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 3

[18] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proc. CVPR*, 2018. 2, 8

[19] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proc. CVPR*, 2019. 2

[20] Nikita Dvornik, Isma Hadji, Konstantinos G. Derpanis, Animesh Garg, and Allan D. Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers. In *NeurIPS*, 2021. 2

[21] Yazan Abu Farha and Juergen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentatio. In *Proc. CVPR*, 2019. 3

[22] Alireza Fathi and James M. Rehg. Modeling actions through state changes. In *Proc. CVPR*, 2013. 3

[23] Yunchao Gong, Qifa Ke, and Michael Isardand Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014. 2

[24] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *Proc. ECCV*, 2014. 2

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 6

[26] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020. 3

[27] Kaiming He, Haoqi Fan, Auxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, 2020. 3

[28] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proc. CVPR*, 2015. 3

[29] De-An Huang, Fei-Fei Li, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *Proc. ECCV*, 2016. 2

[30] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021. 1, 2

[31] Jianfeng Gao Jianwei Yang, Yonatan Bisk. Taco: Token-aware cascade contrastive learning for video-text alignment. In *Proc. ICCV*, 2021. 8

[32] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 8

[33] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Asso-

ciating neural word embeddings with deep image representations using fisher vectors. In *Proc. CVPR*, 2015. 2
[34] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proc. CVPR*, 2014. 2, 3, 8
[35] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *Proc. ICCV*, pages 2556–2563, 2011. 8
[36] Hilde Kuehne, A. Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. In *CVIU*, 2017. 2
[37] Colin Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection. In *Proc. CVPR*, 2017. 3
[38] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *Proc. CVPR*, 2018. 3
[39] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020. 2
[40] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *ICCV*, 2019. 2, 8
[41] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 2
[42] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proc. ECCV*, 2018. 3
[43] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2
[44] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proc. CVPR*, 2020. 2
[45] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *Proc. CVPR*, 2021. 2, 7
[46] 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 *Proc. CVPR*, 2020. 1, 2, 3, 6, 7, 8
[47] 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 *Proc. ICCV*, 2019. 1, 2, 6, 8
[48] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *NIPS*, 2013. 3
[49] Yasuhide Mori, Hironobu Takahashi, and Ryuichi Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *MISRM*, 1999. 2
[50] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*, 2021. 3
[51] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *Proc. CVPR*, 2016. 2
[52] Bryan A Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *Proc. CVPR*, 2017.
[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1, 2, 7
[54] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proc. CVPR*, 2018. 2
[55] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *Proc. CVPR*, 2012. 3
[56] K. Pramod Sankar, Saurabh Pandey, and C. V. Jawahar. Text driven temporal segmentation of cricket videos. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2006. 2
[57] Pramod Sankar, C. V. Jawahar, and Andrew Zisserman. Subtitle-free movie to script alignment. In *Proc. BMVC*, 2009. 2
[58] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proc. CVPR*, 2016. 3
[59] Gunnar A. Sigurdsson, Gül Varol, X. Wang, Ali Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proc. ECCV*, 2016. 3
[60] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proc. CVPR*, 2016. 3
[61] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 8
[62] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *Proc. ICLR*, 2019. 2
[63] Hao Tan and Mohit Bansal. Lxmert: Learning cross- modality encoder representations from transformers. In *Proc. EMNLP*, 2019. 2
[64] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Aligning Plot Synopses to Videos for Story-based Retrieval. *International Journal of Multimedia Information Retrieval*, 2015. 2
[65] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Proc. ECCV*, 2019. 3
[66] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4
[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4
[68] J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In *Proc. IJCAI*, pages

2764–2770, 2011. 2
[69] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, , and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proc. CVPR*, 2020. 3
[70] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: Region convolutional 3d network for temporal activity detection. In *Proc. ICCV*, 2017. 3
[71] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021. 2
[72] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proc. CVPR*, 2016. 3
[73] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proc. ICCV*, 2017. 3
[74] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, and Jason J. Corso. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, 2020. 2
[75] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2, 3, 8
[76] Luowei Zhou, Chenliang Xu, and Jason J Corso. Youcook2 leaderboard. `https://github.com/LuoweiZhou/YouCook2-Leaderboard`, 2021. 8
[77] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *Proc. CVPR*, 2020. 8
[78] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proc. CVPR*, 2015. 2
[79] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proc. CVPR*, 2019. 2, 6