End-to-End Learning of Visual Representations from Uncurated Instructional Videos

Antoine Miech\textsuperscript{1*} Jean-Baptiste Alayrac\textsuperscript{2*} Lucas Smaira\textsuperscript{2}
Ivan Laptev\textsuperscript{1} Josef Sivic\textsuperscript{1,3} Andrew Zisserman\textsuperscript{2,4}
\textsuperscript{1}ENS/Inria \textsuperscript{2}DeepMind \textsuperscript{3}CIIRC CTU \textsuperscript{4}VGG Oxford
antoine.miech@inria.fr, jalayrac@google.com
https://www.di.ens.fr/willow/research/mil-nce/

Figure 1: We describe an efficient approach to learn visual representations from misaligned and noisy narrations (bottom) automatically extracted from instructional videos (top). Our video representations are learnt from scratch without relying on any manually annotated visual dataset yet outperform all self-supervised and many fully-supervised methods on several video recognition benchmarks.

Abstract

Annotating videos is cumbersome, expensive and not scalable. Yet, many strong video models still rely on manually annotated data. With the recent introduction of the HowTo100M dataset, narrated videos now offer the possibility of learning video representations without manual supervision. In this work we propose a new learning approach, MIL-NCE, capable of addressing misalignments inherent in narrated videos. With this approach we are able to learn strong video representations from scratch, without the need for any manual annotation. We evaluate our representations on a wide range of four downstream tasks over eight datasets: action recognition (HMDB-51, UCF-101, Kinetics-700), text-to-video retrieval (YouCook2, MSR-VTT), action localization (YouTube-8M Segments, CrossTask) and action segmentation (COIN). Our method outperforms all published self-supervised approaches for these tasks as well as several fully supervised baselines. Our joint text-video pretrained model is publicly available at: https://www.di.ens.fr/willow/research/mil-nce/.

1. Introduction

What we see changes what we know.
What we know changes what we see. \textit{Jean Piaget}

Vision and language play an important role in the way humans learn to associate visual entities to abstract concepts and vice versa. This has also become the \textit{de facto} way to successfully train computer vision models. Indeed, from classification where images are categorized based on a fixed list of words to the recent captioning tasks where images or videos are annotated with rich language descriptions, this interplay is one of the driving forces behind recent progress in the field. However, one of the main limitations of this approach is that it requires manually annotating large collections of visual data.

Manual annotation is both cumbersome and expensive. Moreover, for videos, which are the main focus of this work, annotation is also even more challenging due to the ambiguities of choosing the right vocabulary of actions and annotating action intervals in video. This significantly limits the scale at which fully supervised video dataset can be obtained and hence slows down the quest to improve visual representations. Recent work has proposed a promising alternative to this fully supervised approach: leveraging narrated videos that are readily available at scale on the web.

Of particular interest, the recent HowTo100M dataset \cite{52} contains more than 100 million pairs of video clips and associated narrations. It was automatically collected by querying YouTube for instructional videos. Such videos usually depict someone explaining orally...
how to perform a complex human activity, e.g. preparing a particular meal or repairing a car. Our objective in this paper is to learn strong video representations using only this narrated material.

End-to-end learning from instructional videos is a highly challenging task. Indeed, these videos are made in general with the goal of maximizing the number of views, and with no specific intention to provide a training signal for machine learning algorithms. This means that the supervision present in the narration is only weak and noisy. Among typical sources of noise, the principal one by far is the weak alignment between the video and the language: although for the most part the spoken words correlate with what is happening in these videos, this alignment is far from perfect. People might talk about something before actually demonstrating it, but they might also omit to talk about something that is happening because it is clear enough visually. Conversely they could only mention an action without showing it in the case where the step is not essential or trivial to convey with language alone. This is without even considering the irrelevant information given throughout the video (e.g. jokes or credits) as well as the general difficulty of working with spoken language obtained from potentially erroneous speech recognition algorithm as opposed to written text.

In this work, we propose a bespoke training loss, dubbed MIL-NCE as it inherits from Multiple Instance Learning (MIL) and Noise Contrastive Estimation (NCE). Our method is capable of addressing visually misaligned narrations from uncurated instructional videos as illustrated in Figure 1. Equipped with this novel training scheme and a simple joint video and text embedding model, we show that we can successfully train video representations from scratch directly from pixels on the HowTo100M [52] dataset. To demonstrate the quality of the learnt representations, we employ an extensive set of evaluation benchmarks on a wide variety of video understanding tasks: action recognition (HMDB-51, UCF-101, Kinetics-700), text-to-video retrieval (YouCook2, MSR-VTT), action localization (YouTube-8M Segments, CrossTask) and action segmentation (COIN). Notably, our learnt video representations outperform fully supervised baselines trained on Kinetics or ImageNet for several of the tasks. We also show improvements over other self-supervised approaches on HMDB51 and UCF101 even without fine-tuning the learnt representations. Finally, by leveraging the joint video and text representations, our off-the-shelf trained model also reaches state-of-the-art results on YouCook2 and CrossTask, without any training on the target datasets.

Contributions. The contributions of this work are threefold. (i) We propose a method to learn a joint text video embedding in an end-to-end fashion from unlabelled, uncurated narrated videos using the recently introduced HowTo100M [52] dataset. In particular, we introduce a specific loss, dubbed MIL-NCE for Multiple Instance Learning Noise Contrastive Estimation, that enables the learning to cope with the highly misaligned narration descriptions. (ii) We provide a thorough ablation study to quantitatively assess the importance of the different design choices of the approach. (iii) Finally, we demonstrate that the representations thus obtained are competitive with their strongly supervised counterparts on four downstream tasks over eight video datasets.

2. Related work

Learning visual representations from unlabeled videos. As labeling videos is cumbersome, expensive and not scalable, a significant number of prior works have studied the task of learning visual representations from unlabeled videos. Currently, the most effective approach is to collect a large amount of data from social media and use the available metadata as supervision [2, 26]. However, this metadata is often in the form of keywords or tags, rather than (spoken) natural language considered in this work. In addition, the meta data is often platform dependent and rarely publicly available. Self-supervised approaches do not suffer from these issues as the idea is to define a supervised proxy task using labels directly generated from videos. Some of these tasks include: temporal ordering of video clips or frames [24, 46, 54, 86], predicting geometric transformations [38], maximizing the mutual information of multiple views [74], predicting motion and appearance [78], predicting the future, the past or a portion of masked input in the feature space [30, 71, 76], colorizing videos [77], predicting 3D geometry from synthetic data [25], predicting the audio in a feature space [8, 43] or tasks leveraging temporal cycle consistency [23, 81]. In our work, we leverage as supervision for our proxy task, the output of automatic speech recognition (ASR) run on narrated instructional videos. The nature of this supervision has the potential to also provide semantic information [52, 67], which is often missing in works that only exploit pixel-wise cues. Moreover, most of the top performing prior works only study their method on curated video datasets (e.g. Kinetics [15]) where labels have been removed. However, this is not truly learning from unlabeled data as these videos have been carefully selected and verified to belong to classes of interests. Caron et al. [13] further explain the performance gap between training on such curated data versus uncurated ones, truly available at scale. Instead, our approach focuses on the learning of representations only from uncurated videos.

Vision, speech and language. A common alternative to training visual models using manually defined sets of labels is to exploit semantic supervision from natural language or speech. Numerous prior works [19, 21, 27, 28, 42, 51, 55, 60, 62, 88, 79, 80, 83, 84] have used image / video description datasets [48, 63, 65, 87, 90] to learn an embed-
Leveraging Uncurated Instructional Videos

This section describes the proposed approach to train joint video and text embeddings from unlabeled narrated videos in an end-to-end fashion. To start with, we are given \( n \) pairs of video clips and associated narrations. In practice, a pair is composed of a short 3.2 seconds video clip (32 frames at 10 FPS) together with a small number of words (not exceeding 16) that correspond to what the person is saying in the video. For example, someone might be sanding wood while mentioning the action “sanding down” or the object “sander” as illustrated in Figure 2a. Given this input, our goal is to learn a joint embedding space where similarity between the narration and video embedding is high when the text and visual content are semantically similar and low otherwise, and we wish to learn this starting from raw pixels in the video and text descriptions. As illustrated in Figure 1, this is a very challenging problem due to the often severely misaligned visual descriptions.

In this work, we address this issue by introducing the...
MIL-NCE objective:

$$\max_{f,g} \sum_{i=1}^{n} \log \left( \frac{\sum_{(x,y) \in \mathcal{P}_i} e^{f(x)^\top g(y)}}{\sum_{(x,y) \in \mathcal{P}_i} e^{f(x)^\top g(y)} + \sum_{(x',y') \sim \mathcal{N}_i} e^{f(x')^\top g(y')}} \right)$$  (1)

where \( x \) represents a video clip and \( y \) a narration. \( f \) and \( g \) are the two embedding functions that respectively operate over video and text. Given a specific sample (indexed by \( i \)), we construct \( \mathcal{P}_i \) to be a valid set of positive video/narration candidate pairs (see Figure 2) while \( \mathcal{N}_i \) conversely refers to an associated set of negative video/narration pairs. This objective function comes down to maximizing the ratio of the sum of the positive candidate scores from \( \mathcal{P}_i \) to the sum of the scores of all negatives sampled from \( \mathcal{N}_i \), where the score is measured by the exponentiated dot product of the corresponding video and language embeddings, \( f(x) \) and \( g(y) \).

In the following, we describe more precisely the motivation behind the MIL-NCE objective (1). First, section 3.1 introduces the chosen probabilistic model for joint text and video embedding. Given that model, Section 3.2 details the choice behind the training objective (1) explaining how it is specifically adapted to handle the misalignment noise inherent in narrated videos in comparison with existing approaches.

### 3.1. A simple joint probabilistic model

In the following, \( x \in \mathcal{X} \) stands for a video clip and \( y \in \mathcal{Y} \) for a narration. Given a set of \( n \) pairs of video clips and associated narrations \( \{(x_i, y_i)\}_{i=1}^{n} \in (\mathcal{X} \times \mathcal{Y})^n \) sampled from the joint data distribution \( P(\mathcal{X} \times \mathcal{Y}) \), our goal is to learn a joint embedding space where semantically related videos and texts are close and far away otherwise.

Formally, we learn two parametrized mappings: \( f: \mathcal{X} \to \mathbb{R}^d \) maps a video clip \( x \) into a \( d \)-dimensional vector \( f(x) \in \mathbb{R}^d \), and \( g: \mathcal{Y} \to \mathbb{R}^d \) maps a narration \( y \) into the same \( d \)-dimensional vector space, \( g(y) \in \mathbb{R}^d \). We assume that we can estimate up to a constant factor the joint probability of a pair of video and narration \( (x, y) \) by exponentiating the dot product of the two embeddings:

$$p(x, y; f, g) \propto e^{f(x)^\top g(y)}.$$  (2)

In this work, \( f \) takes the form of a CNN that runs over a fixed-length clip. For \( g \), we consider simple sentence based models that transform a set of words into a single vector. Note, for simplicity and with a slight abuse of notation, we refer to \( f \) (or \( g \)) as both a function and the parameters that define it. Also, we will refer to (2) as simply \( p(x, y) \), i.e. we keep the dependence in \( f \) and \( g \) implicit for the clarity of simpler equations. More details about the exact architecture of the models are provided in Section 4.

### 3.2. Learning from uncurated data: MIL-NCE

Recall that our goal is to learn a joint video and text representation only from uncurated narrated videos. In this section, we start by detailing why this is a highly challenging endeavor due to misalignments present in that data. Next, we explain how the introduced MIL-NCE objective (1) enables to learn despite that noise. Finally, we contrast our proposed approach to similar works in the self-supervised domain.

**Misalignment in narrated videos.** In [52], the authors estimate that around 50% of clip-narration pairs from the HowTo100M dataset are not aligned. In fact, people are likely to describe an event after or before performing it in the video as illustrated in Figure 1. This visual misalignment makes it more challenging to learn video representations than with manually annotated and aligned labels.

**How to learn despite noisy supervision?** To address the aforementioned issues, we propose to consider multiple options for matching a video and a narration instead of only comparing a single video \( x \) with a single narration \( y \) as done in [52]. Let’s consider the example illustrated in Figure 2a. Given a clip \( x \), \( K \) narrations \( \{y_k\}_{k=1}^{K} \) that happen close in time within the same video can be considered as positive candidates. By doing so, the chance that spoken words correlate with what is happening in the video increases. In that case, we would like to match at least one of the narrations \( \{y_k\}_{k=1}^{K} \) with video \( x \). Given the probabilistic model (2), a natural way to express this is by computing the joint probability of \( x \) happening with any of the \( y_k \). Because we can make the assumption that \( y_k \)’s are mutually exclusive (i.e. \( (x, y_i) \neq (x, y_j), \forall i \neq j \)), this can be expressed mathematically by (2) as follows:

$$p(\cup_k \{(x, y_k)\}) = \sum_k p(x, y_k) \propto \sum_k e^{f(x)^\top g(y_k)}.$$  (3)

This is a MIL extension, which, as opposed to MIL-SVM, does not explicitly select a single positive sample per bag at training. More generally, and symmetrically, the case where several video clips are candidates for a given narration can also be envisioned. Hence, for generality, we assume that instead of having a single pair \((x, y)\), we have a set of candidate positive pairs \( \mathcal{P} = \{(x_k, y_k)\}_{k=1}^{K} \), and we can simply repurpose (3) as \( p(\mathcal{P}) \propto \sum_{(x,y) \in \mathcal{P}} e^{f(x)^\top g(y)} \). We denote by \( \{P_i\}_{i=1}^{n} \) the training set of candidate positive pairs deduced from the original training set \( \{(x_i, y_i)\}_{i=1}^{n} \). With this extension, we have the tools to address misalignments. Details about how to construct \( \mathcal{P} \) are given in Section 4.1

**How to train this model?** We wish to learn a video representation based on the previously described probabilistic model \( p(\mathcal{P}) \). However, this is challenging as one cannot directly apply standard generative techniques such as maximum likelihood due to the intractability of computing
Table 1: Video (left) and text (right) model architectures.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input video</td>
<td>32 × 200 × 200 × 3</td>
</tr>
<tr>
<td>I3D / S3D → Mixed 5c</td>
<td>4 × 6 × 6 × 1024</td>
</tr>
<tr>
<td>Global avg pool</td>
<td>1 × 1 × 1 × 1024</td>
</tr>
<tr>
<td>Linear</td>
<td>1 × 1 × 1 × 512</td>
</tr>
<tr>
<td>Embedding</td>
<td>16 × 300</td>
</tr>
<tr>
<td>Linear + ReLU</td>
<td>16 × 2048</td>
</tr>
<tr>
<td>Max pool</td>
<td>1 × 2048</td>
</tr>
<tr>
<td>Linear</td>
<td>1 × 512</td>
</tr>
</tbody>
</table>

Table: We first describe implementation details of our method in Section 4.1. The datasets used in our evaluation are outlined in Section 4.2. We present an ablation study emphasizing key ingredients of our approach in Section 4.3. Finally, we compare our learnt representations to previous self and fully-supervised methods in Section 4.4.

4. Experiments

We first describe implementation details of our method in Section 4.1. The datasets used in our evaluation are outlined in Section 4.2. We present an ablation study emphasizing key ingredients of our approach in Section 4.3. Finally, we compare our learnt representations to previous self and fully-supervised methods in Section 4.4.

4.1. Implementation details

Model and Inputs. For the 3D CNN backbone, we use the standard I3D implementation from [15] for all ablation studies and for the comparison to state-of-the-art, we report result on both I3D and S3D [85]. We use the Google News self-supervised pre-trained word2vec (d=300) embedding from [53] for our word representation. Each video clip at training contains 32 frames sampled at 10 fps (3.2 seconds) with a 200x200 resolution (224x224 at test time). For each narration, we take a maximum of 16 words. More details about the model architecture and input dimensions are provided in Table 1. A detailed illustration of the architecture is also given in the appendix [1].

Visual representations evaluation. We evaluate our visual representations at two different semantic levels. First, we use the output of the I3D (or S3D) Global avg pool (see Table 1), to evaluate our representation for action recognition, action segmentation and action localization. Next, the output of the last I3D (or S3D) Linear layer (see Table 1), which maps the video to the joint text-video semantic space, is used in conjunction with the output of the language model for the text-video retrieval tasks.

Training dataset. We train our model using the HowTo100M [52] narrated video dataset. It consists of more than 1.2M videos accompanied with automatically generated speech transcription. We use the provided transcription to create pairs of video / caption defined by each caption time stamp. Note that while the original dataset [52] consists of 136M pairs, we only used 120M pairs to comply with the YouTube wipe out policy. Each video shorter than 5 seconds is extended symmetrically in time so that the duration is at least 5 seconds. Then we randomly sample, a fixed length clip of 3.2 seconds within each video at training. For each clip-narration training pair \((x, y)\) sampled, we construct the bag of positive candidate pairs \(P\) by considering the nearest captions in time to \(y\) as depicted in Figure 2a. For example, if we set the number of positive candidate pairs to 3, we would have \(P = \{(x, y), (x, y^1), (x, y^2)\}\) where \(y^1\) and \(y^2\) are the 2 closest narrations in time to \(y\). We work with batch containing \(B\) positive video-narration pairs \(\{(x_i, y_i)\}_{i \in [1, B]}\). We construct the set \(\mathcal{N}\) by simply creating negative pairs from this batch by combining \(\{(x_i, y_j)\}_{i \neq j}\). Since all representations are already computed, computing negative scores is cheap and efficient.

Optimization. We use the ADAM [41] optimizer with an initial learning rate of \(10^{-3}\) with linear warm up of 5k steps. The learning rate is decayed twice by a factor of 10. We train our model using Cloud TPUs v3 1, each Cloud TPU having a batch size of 128 videos. Given the high computational load required for training on HowTo100M, we run ablation studies on 4 Cloud TPUs and train our model for 500k steps (\(\sim 3\) days). For our final evaluation in Section 4.4, we pick the best parameters based on our ablation study and then use 64 Cloud TPUs for 400k steps (also \(\sim 3\) days) as we observed that training on bigger batch size, and

1https://cloud.google.com/tpu/
thus more epochs, had a positive impact on performance.

4.2. Downstream tasks

To show the generality of our learned representations, we perform evaluation on five diverse downstream tasks using eight datasets described below.

**Action Recognition:** HMDB-51 [45], UCF-101 [69], Kinetics-700 [14]. We evaluate our video-only representation on the traditional HMDB-51 / UCF-101 as well as the recent Kinetics-700 action recognition tasks.

**Text-to-Video retrieval:** YouCook2 [90], MSR-VTT [87]. We use the YouCook2 and MSR-VTT text-to-video retrieval benchmarks to evaluate our off-the-shelf learnt joint text-video representation. We follow the same evaluation protocol as described in [52]. We report the retrieval performance using the recall at K (R@K) metric (with K=1,5,10) which measures the percentage of clips retrieved at the top K (the higher the better). We also report the median rank (MedR) of videos to be retrieved (the lower the better). Note from [52] that there is no intersection between YouCook2 testing and HowTo100M training videos.

**Action Localization:** YouTube-8M [2] Segments. We evaluate our video representation on YouTube-8M Segments², a subset of the YouTube-8M [2] with precise temporal annotation. We follow the YouTube-8M Segments challenge evaluation protocol and report the mAP metric.

**Action Step Localization:** CrossTask [91]. We use the recently released CrossTask instructional video dataset to evaluate our off-the-shelf learnt joint text-video representation on the task of action step localization. We perform the same evaluation protocol as in [91] and report the average recall (CTR) metric for the localization task.

**Action Segmentation:** COIN [73]. We evaluate our video-only representation on the COIN action segmentation task and follow the evaluation protocol of [71] by reporting the frame-wise accuracy (FA).

4.3. Ablation studies

We perform the ablation studies on the following downstream tasks: MSR-VTT R@10 (MR10), YouCook2 R@10 (YR10), HMDB-51 and UCF-101 recognition accuracy on split 1 and CrossTask average recall (CTR). This subset of downstream tasks has been chosen for their simplicity of evaluation and because they cover a wide range of tasks.

**Which loss is better for learning the joint embedding?** In this ablation study (Table 2a), we compare different losses for matching the text and video embeddings in the standard single-instance learning setting where we pair each video clip to its closest narration in time. We compare the NCE based approach (ours) to the frequently used max margin loss and a binary classification loss (i.e. sigmoid cross entropy loss) that has shown to be effective in video-audio matching [7, 8]. Overall, the NCE loss outperforms other losses or works similarly on all five tested datasets.

**The more negatives, the better.** We keep the same single-instance learning setting and assess the quality of our representations trained with different number of sampled negative examples per positive pair in Table 2b. We can see that the overall performance increases with the number of negatives. For the rest of the ablation studies, we use 512 negative samples per positive.

**How many positive candidates pairs to consider?** We evaluate the benefit of going from the single-instance learning approach to the proposed multiple-instance based approach in Table 2c. In this experiment, we vary the number of positive candidate training pairs ||P|| for each video clip from 1 (i.e. single-instance learning setting) up to 33 candidates. Adding candidates significantly improves the performance.
formance upon the single-instance learning baseline. Moreover, we observe a trade-off between having too many candidates and not having enough of them, as we reach the best results by considering 3 to 5 positive candidates. We believe that adding too many contextual narrations increases the chance for irrelevant ones as they are sampled further in time from the considered video clip. For the rest of the paper we fix the number of positive candidate pairs to 5.

**MIL-NCE vs other MIL based approaches.** In Table 2d, we compare our MIL-NCE approach with methods that can also handle multiple possible candidate captions at training time. The max-pool based approach [5, 8, 58] (Max+NCE) only optimizes over the clip-caption pair with the highest similarity score among the positive candidates. On the other hand, the attention-based approach [36] (Attn+NCE) computes cross-modal attention weights between all the clip-caption pairs and perform a weighted average of the similarity scores in order to consider the most relevant positive candidate pairs. More details about these baselines are provided in the appendix [1]. Finally, we also compare to the single-instance learning baseline where we concatenate all of the candidate narrations as one longer narration (Cat+NCE). Our proposed method outperforms these two standard approaches on five out of six tasks. Qualitative figures in [1] illustrate examples of selected pairs from a hold-out set of HowTo100M videos using MIL-NCE.

**Symmetric or asymmetric negative sampling?** Recall that given a pair of video/narration \((x, y)\), we create \(N\) in a symmetric manner by sampling negative narrations for the video \(x\) and negative videos for the narration \(y\). Table 2e compares that approach \((x, y)\) to asymmetric alternatives: (i) by fixing the video \(x\) and only sampling negative captions \((y|x)\) and (ii) by fixing the narration \(y\) and only sampling negative videos \((x|y)\). Overall, the best results are achieved when sampling jointly the negatives \((x, y)\), i.e. when we equally sample both video and narration negatives.

**Which language model?** Finally, we also experiment with different language models (1 layer LSTM [35] or GRU [18], 1 layer and 8 attention heads Transformer [75] and NetVLAD with 32 clusters [6]) and compare them to our simple model (see Table 1) in Table 2f. Even though our language model is similar to simple bag-of-word approach, it performs in average better and is more consistent over the five tasks than the other models. In particular, our model significantly outperforms the other language models on the text-to-video retrieval tasks (YR10 and MR10), where language plays the most important role. We believe that a sophisticated language understanding is not key for our learning task. Instead, most of the time, detecting and matching the main keywords in each narration is enough. In our supplementary material [1], we replace word2vec with a pretrained BERT model and obtain worse performance. We believe this is due to the domain gap between web text

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>MM</th>
<th>Model</th>
<th>Frozen</th>
<th>HMDB</th>
<th>UCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours HTM Text</td>
<td></td>
<td></td>
<td>I3D</td>
<td>✓</td>
<td>54.8</td>
<td>83.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S3D</td>
<td>✓</td>
<td>59.2</td>
<td>89.1</td>
</tr>
<tr>
<td>Fully-supervised SOTA [85]</td>
<td></td>
<td></td>
<td>S3D</td>
<td>✓</td>
<td>53.1</td>
<td>82.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61.0</td>
<td>91.3</td>
</tr>
</tbody>
</table>

Table 3: Comparison to self-supervised methods on HMDB/UCF. Results are reported by averaging the accuracy over the 3 splits for both datasets. *Shuffle & Learn and 3DRotNet reported numbers are reimplemented in [71] by using a better backbone (S3D). The MM column indicates whether or not other modalities than the video frames have been used for the learning of the visual features, FC: FlyingChairs.

4.4. Comparison to the state-of-the-art

**Video only representation.** In Table 3, we evaluate our learnt representations on the HMDB-51 [45] and UCF-101 [69] action recognition benchmarks by extracting averaged pooled Mixed_{5c} features from the HowTo100M pretrained backbone. More specifically, we compare to self-supervised approaches, which similarly to our work, do not make use of any annotated video or image dataset when training the visual representation. We outperform state-of-the-art on UCF-101 and perform on par with AVTS [43] on HMDB-51. Most importantly, our learnt representation significantly outperforms many prior approaches even without fine-tuning. This result is significant as it demonstrates generalization of our representation to diverse sets of actions despite being trained on uncurated instructional videos.

Next, we evaluate our visual representation on COIN [73] action segmentation task in Table 4a. We split videos in subsequent clips of 1.5 seconds and represent them by concatenating three features: the local representation from I3D (or S3D), the global average pooled representation across the entire video and the relative positional embedding of the video clip. We train a logistic regression to predict the label for each clip. We compare our HowTo100M pretrained I3D network to an I3D fully-supervised on Kinetics-400, Kinetics-700 as well as a ResNet-50 fully supervised on ImageNet. We also compare
Table 4: Evaluation on action segmentation (a), localization (b, d) and recognition (c) benchmarks. K400: Kinetics-400, K600: Kinetics-600, K700: Kinetics-700, HTM: HowTo100M, ImNet: ImageNet, YT8M-S: YouTube-8M Segments, R50: 2D ResNet-50.

<table>
<thead>
<tr>
<th>Method</th>
<th>Labeled dataset used</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MedR↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>None</td>
<td>0.03</td>
<td>0.15</td>
<td>0.3</td>
<td>1675</td>
</tr>
<tr>
<td>HGLMM FV CCA [42]</td>
<td>ImNet + K400 + YouCook2</td>
<td>4.6</td>
<td>14.3</td>
<td>21.6</td>
<td>75</td>
</tr>
<tr>
<td>Miech et al. [52]</td>
<td>ImNet + K400</td>
<td>6.1</td>
<td>17.3</td>
<td>24.8</td>
<td>46</td>
</tr>
<tr>
<td>Miech et al. [52]</td>
<td>ImNet + K400 + YouCook2</td>
<td>8.2</td>
<td>24.5</td>
<td>35.3</td>
<td>24</td>
</tr>
<tr>
<td>Ours (I3D)</td>
<td>None</td>
<td>11.4</td>
<td>30.6</td>
<td>42.0</td>
<td>16</td>
</tr>
<tr>
<td>Ours (S3D)</td>
<td>None</td>
<td>15.1</td>
<td>38.0</td>
<td>51.2</td>
<td>10</td>
</tr>
</tbody>
</table>

(a) YouCook2

Table 5: Zero-shot evaluation on text-to-video retrieval.

<table>
<thead>
<tr>
<th>Method</th>
<th>Labeled dataset used</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MedR↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>None</td>
<td>0.01</td>
<td>0.05</td>
<td>0.1</td>
<td>500</td>
</tr>
<tr>
<td>Miech et al. [52]</td>
<td>ImNet + K400</td>
<td>7.5</td>
<td>21.2</td>
<td>29.6</td>
<td>38</td>
</tr>
<tr>
<td>Ours (I3D)</td>
<td>None</td>
<td>9.4</td>
<td>22.2</td>
<td>30.0</td>
<td>35</td>
</tr>
<tr>
<td>Ours (S3D)</td>
<td>None</td>
<td>9.9</td>
<td>24.0</td>
<td>32.4</td>
<td>29.5</td>
</tr>
</tbody>
</table>

(b) MSRVT

<table>
<thead>
<tr>
<th>Method</th>
<th>Labeled dataset used</th>
<th>FA</th>
<th>mAP</th>
<th>Init</th>
<th>Net</th>
<th>Top1</th>
<th>CTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scratch I3D</td>
<td>R50 ImNet</td>
<td>52.0</td>
<td>73.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours I3D</td>
<td>K400</td>
<td>52.9</td>
<td>74.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours I3D</td>
<td>K700</td>
<td>54.2</td>
<td>75.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBT [71]</td>
<td>S3D K600+HTM</td>
<td>53.9</td>
<td>77.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours I3D</td>
<td>HTM</td>
<td>59.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours S3D</td>
<td>HTM</td>
<td>61.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) COIN

Joint text-video representation. We report text-to-video retrieval results on the YouCook2 (Table 5a) and MSR-VTT (Table 5b) using our off-the-shelf model trained on HowTo100M. Note that our model has not seen any YouCook2 nor MSR-VTT annotated videos, hence for fair comparison on the MSR-VTT dataset we only compare to prior work [52] that did not finetune on MSR-VTT. On YouCook2, our model significantly outperforms all prior work. More specifically, it performs better than [52] which uses visual representation trained on Kinetics-400 + ImageNet and trains the joint text-video representation on both HowTo100M and YouCook2. On MSR-VTT, our method performs also better than [52], yet without using any manually annotated dataset. Finally, we also evaluate our off-the-shelf model trained on HowTo100M on the CrossTask [91] action localization benchmark in Table 4d. We perform localization via a video-to-text retrieval approach similarly to [52]. Our method outperforms state-of-the-art approaches, here again, without using manual supervision.

5. Conclusion

We have addressed the challenging task of learning visual representations from scratch using uncurated instructional videos. Our approach did not rely on any manually annotated video or image dataset. To deal with the misalignment between narrations and videos, we have introduced MIL-NCE, a multiple instance learning approach derived from noise contrastive estimation. We have applied MIL-NCE to the uncurated HowTo100M dataset and obtained strong visual representations that outperformed self-supervised as well as fully-supervised representations on downstream tasks. More generally, we believe MIL-NCE can be applicable in several multiple instance learning problems where representation learning is key.

Acknowledgements. We would like to thank: Relja Arandjelović, Pauline Luc, Gunnar Sigurdsson for helpful discussions. The project was partially supported by Antoine Miech Google PhD fellowship, the ERC grant LEAP (No. 336845), the CIFAR Learning in Machines&Brains program, the Louis Vuitton - ENS Chair on Artificial Intelligence, the French government program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), and the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468).
References


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


[34] David Harwath, Adrià Recasens, Didac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In ECCV, 2018. 3


[51] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makaran Tapaswi, Ivan Laptev, and Josef Sivic. Howto100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 1, 2, 3, 4, 5, 6, 8
[59] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 2


Liwei Wang, Yin Li, and Svethena Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016.


