# Space-Time Crop & Attend:
# Improving Cross-modal Video Representation Learning.

Mandela Patrick,* Po-Yao Huang*, Ishan Misra, Florian Metze, Andrea Vedaldi
Facebook AI Research
mandelapatrick,berniehuang,imisra,fmetze,vedaldi@fb.com

Yuki M. Asano*, João Henriques
Oxford University
yuki, joao@robots.ox.ac.uk

## Abstract

*The quality of the image representations obtained from self-supervised learning depends strongly on the type of data augmentations used in the learning formulation. Recent papers have ported these methods from still images to videos and found that leveraging both audio and video signals yields strong gains; however, they did not find that spatial augmentations such as cropping, which are very important for still images, work as well for videos. In this paper, we improve these formulations in two ways unique to the spatio-temporal aspect of videos. First, for space, we show that spatial augmentations such as cropping do work well for videos too, but that previous implementations, due to the high processing and memory cost, could not do this at a scale sufficient for it to work well. To address this issue, we first introduce Feature Crop, a method to simulate such augmentations much more efficiently directly in feature space. Second, we show that as opposed to naïve average pooling, the use of transformer-based attention improves performance significantly, and is well suited for processing feature crops. Combining both of our discoveries into a new method, Space-Time Crop & Attend (STiCA) we achieve state-of-the-art performance across multiple video-representation learning benchmarks. In particular, we achieve new state-of-the-art accuracies of 67.0% on HMDB-51 and 93.1% on UCF-101 when pre-training on Kinetics-400. Code and pretrained models are available[1].*

## 1. Introduction

Visual representations have evolved significantly in the last two decades. The first generation of representations

---

*Equal contribution.
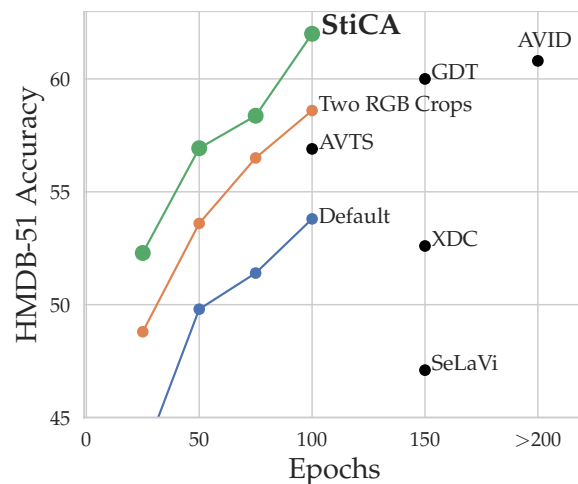[1]https://github.com/facebookresearch/GDT



**Figure 1: HMDB-51 accuracy vs epoch.** Our method, **STiCA**, combines space-time crops in feature space with self-attention of time in latent space. This yields significant benefits not only in performance but also in speed compared to cropping in input space using *two RGB crops*, or simply using the *default* cross-modal only loss. Compared to recent state-of-the-art cross-modal self-supervised learning methods (XDC [6], GDT [106], AVID-CMA [97], SeLaVi [9]) pre-trained on Kinetics-400 [69] STiCA is able to achieve significantly better results in fewer epochs.

comprises algorithms such as SIFT [87] and HOG [30] that were designed manually. The second generation comprises representations learned from data by using deep neural networks and manual supervision [31, 59, 76]. We are now transitioning to the third generation, where representations are learned from data without using any manual annotations by means of self-supervision. Current self-supervised rep-

resentations, obtained from methods such as MoCo [57], SimCLR [24] or SwAV [20], convincingly outperform supervised ones on downstream tasks such as image classification, segmentation and object detection. Furthermore, most of these methods are based on noise-contrastive instance discrimination, which was proposed in ExemplarC-NNs [39] and put in its current form in [142] and [101]. The idea is to learn representations that are invariant to irrelevant factors of variations, modelled by strong augmentations such as image cropping, while remaining distinctive for the identity of the image.

Noise-contrastive learning is of course not limited to still images. In particular, a number of recent approaches [54, 93, 97, 106] have used noise-contrastive formulations to learn visual or audio-visual representations. However, these methods are not as well developed as their counterparts for still images, with current state-of-the-art methods [54, 106] still lagging behind their supervised counterparts.

In this paper, we identify two areas in which current video representation learning formulations are lacking and improve on them, thus significantly improving upon the current state of the art in this area.

The first shortcoming is the lack of a sufficient encoding of **spatial invariances**. For still images, learning spatial invariances has been shown to be one of the most important factors for performance [20, 24]. Almost all methods achieve some form of spatial invariance simply by applying different spatial augmentations to the images in different epochs of training. However, learning spatial invariances in this manner requires a slow training process that lasts for many epochs ($\sim$800). Authors have suggested that packing several augmentations of the same image in a single data batch is more effective as it provides a much stronger and more direct incentive for the network to learn invariances [20].

For videos, both strategies are less feasible. Training a model for 200 epochs on Kinetics-400 [69] already requires around 1.5K GPU hours on recent Nvidia V100 architectures, and with recent datasets such as IG65M [45] and HowTo100M [94] only a handful of epochs can realistically be completed. On the other hand, including multiple augmentations of the same video in a batch rapidly exhausts the memory of GPUs. Since batch sizes per GPU are already in the single digits due to the size of video data, including several augmentations is unfeasible. This is particular detrimental for recent contrastive learning approaches such as [24, 58], where reducing the batch size means reducing the pool of negative contrastive samples.

In order to solve this problem, we propose to move spatial augmentations to the feature space, in a manner specifically tailored to contrastive learning. Instead of extracting a large number $R$ of different augmentations in the input RGB space, we extract only two of them, apply the trunk of the neural network to extract corresponding features, and then extract $R/2$ more augmentations directly in feature space. In this way, one needs to evaluate the slow and memory taxing feature extraction part of the network only twice, regardless of the number of augmentations that are produced. We show that this *feature-level augmentation* significantly improves representation learning performance.

The second challenge that we tackle is how to best encode **temporal information** in self-supervised video representation learning. Currently, most self-supervised video representation learning approaches use 3D-CNNs [21, 132, 133, 144] that compute convolutions across space and time, but the final representation is generated by naïve global average pooling over space and time, crucially discarding temporal ordering.

In order to address this shortcoming, in this work we propose to use a contextualized pooling function based on the transformer architecture [135] for both self-supervised pretraining and supervised finetuning. The intuition is that, via multi-head self-attention, the transformer can capture temporal dependencies much better than average pooling, especially for longer inputs. Transformers can also benefit from our feature-level crops, as the latter resemble the common approach of randomly masking the inputs to the transformer [62]. Experimental results show that this modification improves the performance of the learned video representations substantially, and is cumulative with the benefit of feature crops, at about the same cost of average pooling.

We combine both of our proposed improvements into a new self-supervised learning approach: Space-Time Attention and Cropping (**STiCA**). To summarize, with STiCA we make the following three main contributions:

- We demonstrate the benefits of stronger spatial invariances in self-supervised video representation learning for the first time and we propose feature-level augmentation to implement the latter efficiently.

- We propose to use transformers to model time more effectively in self-supervised video representations, replacing average as the pooling function.

- We demonstrate strong performance gains by using the two techniques and obtain state-of-the-art performance on two standard benchmarks (67.0% on HMDB-51 and 93.1% on UCF-101).

## 2. Related Works

**Self-supervised Image Representation Learning.** Self-supervised learning uses pretext tasks to automatically and easily generate differentiable learning signals from the data itself in order to train convolutional neural networks. A variety of pretext tasks have been proposed such as colorization [148, 149], predicting artificial rotations [46], in-

painting [105], spatial context [35, 100], and clustering features [11, 18, 19, 20, 64, 83]. Recently, contrastive methods [50, 51] have proven to be particularly effective at learning transferable image representations [13, 24, 49, 57, 95, 101, 130].

**Self-supervised Video Representation Learning.** For videos, pretext tasks often seek to leverage the temporal dimension to learn representations. Such tasks include predicting clip and sequence order [79, 96, 145], future events [52, 53], the arrow of time [140], 3D geometric transformations [65, 71], playback speed [14, 40, 63, 137], or motion statistics [136].

**Multi-Modal Learning.** The co-occurrence and synchronicity of multiple modalities from videos have been used to learn visual representations from both audio-video [6, 7, 9, 74, 90, 97, 102, 106], and speech-video [5, 73, 85, 92, 93, 98, 107, 123, 125, 126] data. Multi-modal representation learning has several practical applications: lip reading [3, 26, 27], audio-visual source separation and localization [2, 4, 8, 56, 150, 151], speech recognition [1, 111], efficient inference [43, 75], egocentric action recognition [70] and audio-visual navigation [22].

**Data Augmentations.** Data augmentation has proven to be useful in training deep learning models in many domains, from vision [28, 29, 146] to speech [103]. Data transformations are the foundation of most self-supervised works, and there has been early attempts to even learn the optimal distribution of transformations [16, 29]. Particularly for contrastive learning, the choice of data transformations has been shown to be particularly important to learn desirable invariances and equivariances [95, 106, 130, 131].

**Transformations in Feature-Space.** Some works have proposed forms of augmentation in feature-space, by adding noise and linear transformations [129], and by associating samples to prototypes in feature-space [78]. These augmentations do not correspond to interpretable geometric operations, however. Crops in feature-space are commonly used in supervised detection pipelines, such as Faster R-CNN and region-based architectures [115], and in earlier detectors based on manually-engineered features [30]. However, the objective of these transformations is to enumerate a space of outputs (e.g. bounding box predictions) for supervised prediction. In self-supervised learning, while [66] uses feature mixing to create harder negatives for contrastive learning, we are instead interested in using feature crop augmentation to achieve spatial invariance.

**Temporal Modeling.** Videos extend images by adding a temporal dimension. Therefore, there has been a large family of research that has looked into how to model temporal information in videos. Early works incorporated temporal information via average pooling of frame/clip-level

features [48, 68, 138], while later work used 3D convolution neural networks [132, 133, 144] and recurrent-neural networks [37]. Other approaches leverage long-term temporal convolutions [134], self-attention [139], relation networks [152], multi-scale temporal convolutions [61], or optical flow in a two stream network [119].

**Transformers in Vision.** With the success of the transformer architecture [135] in natural language processing [62], transformers are being used in various vision domains such as image representation learning [23, 32, 38, 118, 141], image generation [104], object detection [17, 86], few-shot learning [36], video action recognition [15, 47, 99, 139], video question-answering [67], image-text [84, 88, 124, 127, 128] and video-text [42, 73, 107, 125, 126, 154] representation learning.

## 3. Method

Our goal is to learn a general-purpose *data representation* $\Phi : \mathcal{X} \to \mathcal{Z} = \mathbb{R}^D$ that maps data $\boldsymbol{x} \in \mathcal{X}$ to feature vectors $\boldsymbol{z} = \Phi(\boldsymbol{x})$. In the supervised setting, representations are learned end-to-end as components of larger systems that solve certain tasks of interest, such as image or video classification, under the assumption that supervision is available to drive the learning process. When supervision is not available, representations can still be learned via self-supervision by means of suitable pretext tasks. Among the latter, *noise contrastive learning* is one of the most popular and successful ones [24, 101]. We summarize this background next and discuss our extensions in the following sections.

### 3.1. Background: Multi-modal contrastive learning

The idea is to train the representation $\Phi$ to identify data points up to the addition of noise or, more generally, the application of certain nuisance transformations. To this end, let $g : \mathcal{X} \to \mathcal{X}$ be transformations sampled in a set $\mathcal{G}$ of possible nuisances (for example random image crops). Let $\text{sim}(\boldsymbol{z}', \boldsymbol{z}'')$ be a similarity function comparing representations $\boldsymbol{z}'$ and $\boldsymbol{z}''$, such as the cosine similarity:

$$\text{sim}(\boldsymbol{z}', \boldsymbol{z}'') = \frac{\langle \boldsymbol{z}', \boldsymbol{z}'' \rangle}{\|\boldsymbol{z}'\| \, \|\boldsymbol{z}''\|}.$$

Consider a dataset or batch $\mathcal{B} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ of data samples. Slightly modifying [24], for each sample $\boldsymbol{x}_i$, draw a set of random nuisance transformations $\{g_{\alpha i}\}_{1 \leq i \leq N}$ and let $\boldsymbol{z}_{\alpha i} = \Phi(g_{\alpha i}(\boldsymbol{x}_i))$ be the representations of the transformed samples. Likewise, consider a second set $\beta$ of transformations $\{g_{\beta i}\}_{1 \leq i \leq N}$. The noise contrastive loss (NCE) is given by:

$$\mathcal{L}(\alpha, \beta) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{\frac{1}{\tau} \text{sim}(\boldsymbol{z}_{\alpha i}, \boldsymbol{z}_{\beta i})}}{\sum_{j=1}^{N} e^{\frac{1}{\tau} \text{sim}(\boldsymbol{z}_{\alpha i}, \boldsymbol{z}_{\beta j})}} \quad (1)$$
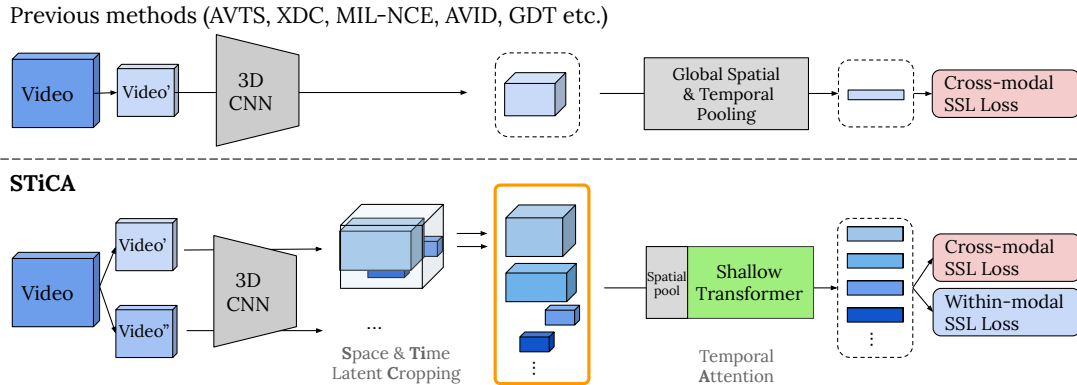
**Figure 2: Approach Overview.** We present a self-supervised approach that learns video representations without labels. **(Top)** Prior work in video representation learning did not capture spatial invariances, as taking many crops of the input (at varying locations and scales), quickly gets expensive in both compute and memory. **(Bottom)** The proposed method generates a large variety of views from only two RGB-crops by cropping in latent space and is particularly tailored to self-supervised contrastive learning. The latent crops are essentially masked features, which are then further processed by a light-weight temporal transformer. Compared to global pooling, this allows our method to further capture the rich temporal signal.

where $\tau > 0$ is a temperature parameter. This loss pulls together the representations of samples that only differ by the transformation while pushing apart the others. Note that this definition is not symmetric in the two arguments $\alpha$ and $\beta$ (i.e., $\mathcal{L}(\alpha, \beta) \neq \mathcal{L}(\beta, \alpha)$. Note also that we can introduce any number of transformation sets $\alpha, \beta, \gamma, \ldots$ and, for each pair, we can obtain a different variant of eq. (1).

Recently, works such as [106] have ported this technique to the video domain by contrasting modalities. Each video $\boldsymbol{x} = (\boldsymbol{v}, \boldsymbol{a})$ consists of a visual component $\boldsymbol{v}$ and an audio component $\boldsymbol{a}$. One consider two sets of transformations $g_v$, extracting and augmenting the visual component, and $g_a$, extracting and augmenting the audio component. We still write $\Phi(g_\alpha(\boldsymbol{x}))$ for the feature computed for either visual and audio components, but the symbol means that a modality-specific neural network is applied as needed.[2]

With this, we can derive three variants of eq. (1), involving mixed visual-audio and homogeneous visual-visual and audio-audio comparisons. Their combinations are:

$$\lambda_{va}\mathcal{L}(v, a) + \lambda_{av}\mathcal{L}(a, v) + \lambda_{vv}\mathcal{L}(v_1, v_2) + \lambda_{aa}\mathcal{L}(a_1, a_2).$$
(2)

where $\lambda_{va}$, $\lambda_{av}$, $\lambda_{vv}$ and $\lambda_{aa}$ are non-negative mixing weights.

**Challenge 1: Encoding within-modality invariance.** While all terms in 2 code for desirable invariances of the representation, several recent papers [90, 97, 106] have found that the mixed term $\lambda_{va}$ is far more important than the other two; in fact, performance *degrades* if one sets $\lambda_{aa}, \lambda_{vv} \neq 0$, meaning that within-modal invariances are

---

[2]In other words, $\Phi = (\Phi_v, \Phi_a)$ is really a pair of networks, producing embedding vectors $\boldsymbol{z}_\alpha$ that are compatible regardless of the modality $\alpha \in \{v, a\}$.

not successfully leveraged. Our hypothesis is that within-modality invariance can be beneficial, and that these early negative results are due to the fact that current learning formulations are ineffective at capitalizing on this signal.

As suggested in Sec. 1, the fact that video data is large means that the batch size used in learning must be small. As a consequence, a batch can contain only a very small number of different augmentations of the same video sample. In current multi-modal learning formulations, each video is already transformed twice in order to extract video and audio components, so cross-modal invariance is learned well. However, the downside is that there is no space left in the batch for multiple visual or audio augmentations. Thus, within-modality invariance is learned only indirectly — in particular, as noted in Sec. 1, two different visual or audio augmentations of the same video are visited by the model only after an entire training epoch. Next, we address this issue by making it feasible to extract several within-modality transformations in the same batch even for video data.

### 3.2. Efficient spatial cropping for augmentation

It has been found that self-supervised learning benefits from, and requires more and stronger augmentations compared to the supervised counterpart for optimal performance [24]. In particular, several papers [10, 20, 24] have suggested that, in the case of still images, the most important type of augmentation is *cropping*. Namely, given an RGB image $\boldsymbol{x} \in \mathbb{R}^{3 \times H \times W}$ with three channels and height and width $H$ and $W$ respectively, a crop is given by a box $B = (x_{\min}, x_{\max}, y_{\min}, y_{\max})$. The image tensor is first cropped as:

$$C_B(\boldsymbol{x}) = \boldsymbol{x}_{[:, \, y_{\min}:y_{\max}, \, x_{\min}:x_{\max}]}$$
(3)

where the : symbol is used to denote an index range. The cropped tensor is then resized to a tensor $\tilde{\boldsymbol{x}} = g(\boldsymbol{x}) = R_{H_0 W_0}(C_B(\boldsymbol{x})) \in \mathbb{R}^{3 \times H_0 \times W_0}$ with a given height and width $H_0 \times W_0$. In practice, $R_{H_0 W_0}$ may also apply additional augmentations such as color jittering, as detailed in the experiments.

As for the visual part $\boldsymbol{v} \in \mathbb{R}^{3 \times T \times H \times W}$ of a video, the situation is similar, except that the video also contains an additional temporal dimension $T$. To avoid extreme spatial jittering and keep objects aligned, a spatial crop is usually taken at the same location in the input space throughout the whole temporal dimension, so we consider the tube $B = (x_{\min}, x_{\max}, y_{\min}, y_{\max}, t_{\min}, t_{\max})$ and define $\tilde{\boldsymbol{v}} = g_v(\boldsymbol{v}) = R_{H_0 W_0}(C_B(\boldsymbol{v})) \in \mathbb{R}^{3 \times T_0 \times H_0 \times W_0}$ by extending (3) in the obvious way.

The deep neural network $\boldsymbol{z} = \Phi(\tilde{\boldsymbol{v}})$ mapping $\boldsymbol{v}$ to its corresponding code $\boldsymbol{z}$ is fed with tensors with two spatial dimensions and a temporal one. Such networks, often called 3D for this reason, include R3D [55], S3D [144] and R(2+1)D [133]. As customary in deep convolutional neural networks, they first produce an intermediate tensor with lower space-time resolution and then pool the latter to obtain a single code vector for the entire video. We explicitly break this down into three functions

$$\Phi(\tilde{\boldsymbol{v}}) = (\mathcal{P}_t \circ \mathcal{P}_s \circ \Psi)(\tilde{\boldsymbol{v}}) \qquad (4)$$

Here, the first function is a 3D convolutional neural network $\Psi(\tilde{\boldsymbol{v}}) \in \mathbb{R}^{D \times T_1 \times H_1 \times W_1}$ producing a tensor with reduced resolution $T_1 < T_0$, $H_1 < H_0$, $W_1 < W_0$. The operators $\mathcal{P}_s$ and $\mathcal{P}_t$ collapse, respectively, spatial and time dimensions via average pooling.

Now consider implementing term $\mathcal{L}(v_1, v_2)$ in 2. In this case, one samples from each video $\boldsymbol{x}_i$ two different space-time crops $g_{v_1 i}(\boldsymbol{x}_i)$ and $g_{v_2 i}(\boldsymbol{x}_i)$, each corresponding to random tubes $B_1$ and $B_2$ respectively. The tubes are not sampled entirely independently, however, as they have the same temporal extent $(t_{\min}, t_{\max})$.

**Naïve multiple spatial cropping** In practice, [20, 24, 82, 95] show that taking multiple image crops improves self-supervised image representations. We can achieve a similar effect for videos by summing losses $\mathcal{L}(v_\alpha, v_\beta)$ for sets of visual transformations $v_\alpha \neq v_\beta$, obtained by sampling multiple space-time tubes for each video, but this is practically difficult, both due to the large memory footprint and the compute overhead of the slow 3D CNN for each crop.

The Multi-Crop approach introduced by SwAV [20] in the image domain combined with our asymmetric contrastive formulation (1) can partially reduce the complexity. For Multi-Crop, we consider three crop sizes $\alpha \in \{L_1, L_2, S\}$ where $L_1$ and $L_2$ stands for large and $S$ for small. The use of a small crop allows to reduce the memory consumption when the representation $\Phi$ is computed. We

then have losses:

$$\mathcal{L}(v_{L_1}, v_{L_2}) + \mathcal{L}(v_{L_2}, v_{L_1}) + \mathcal{L}(v_{L_1}, v_S) + \mathcal{L}(v_{L_2}, v_S).$$

While operating on small videos saves some computation, in practice this approach is insufficient to allow using more than a handful of crops in total.

**Efficient cropping in feature space.** As illustrated in Fig. 2, a much more efficient alternative to cropping the input video is to crop intermediate features.

To do so, we first apply the trunk $\Psi$ of the representation to an input-space crop of the visual component of the video $\tilde{\boldsymbol{v}} = R_{H_0 W_0}(C_B(\boldsymbol{v})) \in \mathbb{R}^{D \times T_1 \times H_1 \times W_1}$. Then we can efficiently construct a new view of this data by applying the *Feature Crop* $C_{\bar{B}}$ directly on each intermediate representation, yielding

$$\bar{\boldsymbol{v}} = C_{\bar{B}}(\Psi(\tilde{\boldsymbol{v}})) = \Psi(\tilde{\boldsymbol{v}})_{[t_{\min}:t_{\max}, \, y_{\min}:y_{\max}, \, x_{\min}:x_{\max}]} \qquad (5)$$

Since the operator $C_{\bar{B}}$ is lightweight, it can be used to compute several such random views efficiently; by comparison, cropping the input RGB images requires recomputing the trunk $\Psi$ multiple times.

In practice, given an input video $\boldsymbol{v}$, we generate the following views. First, we apply two crops in RGB space, producing two large crops $L_1$ and $L_2$. Then, for each of those, we use the operator (5) to generate $m$ medium-sized and $n$ small-sized crops $\mathcal{T}_i = \{M_1 L_i, \ldots, M_m L_i, S_1 L_i, \ldots, S_n L_i\}$. We define an overall within-modality loss by summing losses for each pairs of views in $\mathcal{T}$ with exception of pairs where both crops are small:

$$L_{vv} = \sum_{\alpha, \beta} \mathcal{L}(v_\alpha, v_\beta) + \mathcal{L}(v_\beta, v_\alpha), \quad \text{where}$$

$$(\alpha, \beta) \in (\mathcal{T}_1 \times \mathcal{T}_2) - (\mathcal{S}_1 \times \mathcal{S}_2) \qquad (6)$$

Note that there are $2((m+n)^2 - n^2)$ terms in this loss. This is a far greater number of comparison than afforded by the two initial input-space RGB crops.

### 3.3. Temporal modelling with transformers

We now discuss our second improvement: better modelling of time.

**Challenge 2: Modelling time better.** Contrary to spatial invariance, models should not be fully invariant to time as the latter can encode causality and with it semantics: a video of someone starting a fire is very different from its reversed version, in which someone extinguishes it. In standard 3D networks, features in the trunk are sensitive to the temporal order, but this information is lost in the final stage, where temporal averaging is applied. We argue that the value of the lost information increases with the length of the video, and that this information can be leveraged by switching to a different pooling function.

**Temporal transformer.** We propose to tackle this issue by replacing average pooling in time $\mathcal{P}_t$ in eq. (4) with a transformer $\mathcal{P}_{\text{transf}}$. Transformers [135] have been shown effective for representing sequential inputs in the NLP domain [62, 81, 113, 114]. After spatial averaging, the output $\boldsymbol{h} = \mathcal{P}_s(\Psi(\tilde{\boldsymbol{v}})) \in \mathbb{R}^{D \times T_1}$ of the network has one feature vector per time step, and is thus amenable to processing by a transformer. The feature $\boldsymbol{h}$, which differs in latent time-dimension size from its uncropped variant can be seen as masking the transformer's attention. Masking attention has been used in transformer encoder-decoder training to prevent the model from cheating [33] and encourage it to leverage information from the context. We use a shallow and light-weight transformer on top of our feature cropping procedure, which we show to be sufficient to reap the benefit of better temporal modelling incurring only a very small computational cost. We use 2-layers and 4 self-attention heads and provide further details on the transformer architecture in the Appendix.

## 3.4. Overall loss

Our combined model, **STiCA**, better learns space-time invariances and relationships by cropping in space-time and leveraging temporal attention with a transformer. For training, we sample $N$ videos in a batch and, for each of them, compute two 'large' visual crops in RGB space, $2(n + m)$ small and medium feature crops (Sec. 3.2), and an audio augmentation $a$. With those, the overall objective is obtained by summing the within-modality loss $L_{vv}$ from eq. (6) to the cross modality losses:

$$L = \lambda_{vv} L_{vv} + \lambda_{va} L_{va}, \tag{7}$$

where $L_{va} = \mathcal{L}(v_{L_1}, a) + \mathcal{L}(v_{L_2}, a) + \mathcal{L}(a, v_{L_1}) + \mathcal{L}(a, v_{L_2})$.

## 4. Experiments

We first describe the datasets (Sec. 4) and implementation details (Sec. 4) for pretraining. In Sec. 4.1, we describe the downstream tasks for evaluating the representation obtained from self-supervised learning. In Sec. 4.2, we ablate the various components of our method, and the importance of temporal context and multi-modality in Sec. 4.3. Lastly, in Sec. 4.4, we compare with prior work in video and multimodal representation learning.

**Data.** We pretrain on the Kinetics-400 dataset [69], which contains about 230K training videos and 13K validation videos belonging to 400 action classes. This dataset is the "ImageNet" for video representation learning due to its moderate size and being public, allowing for broad access and comparability. After pretraining, we evaluate using video action retrieval and action recognition on HMDB-51 [77] and UCF-101 [120]. HMDB-51 [77] consists of 7K

video clips spanning 51 different human activities. HMDB-51 has three train/test splits of size 5K/2K respectively. UCF-101 [120] contains 13K videos from 101 human action classes, and has three train/test splits of size 11K/2K respectively.

**Implementation details.** Following [106], we use the R(2+1)-18 [133] network as visual encoder and ResNet [59] with 9 layers as audio encoder. We train for 100 epochs and use 30 frames with temporal stride of 1 at sampling rate of 30fps at spatial resolution of $112 \times 112$ as input. In our ablations, we evaluate the learned representation by finetuning the visual encoder on fold 1 of the HMDB-51 [77] action recognition dataset. Further implementation details are given in the Appendix.

### 4.1. Downstream tasks

**Video action retrieval.** For video retrieval, we follow the standard protocol described in [145]. We use the split 1 of UCF-101, and additionally HMDB-51. We uniformly sample 10 clips per video, max pool and then average the features after the last residual block for each clip per video. We use these averaged features from the validation set to query the videos in the training set. If the class of a retrieved video matches the class of query video, we count it as a match. We measure recall at $k=1, 5, 20$.

**Video action recognition.** As is standard in the literature, we evaluate our pretrained representations by finetuning our visual backbone on the video action recognition task on HMDB-51 and UCF-101 datasets. We closely follow the finetuning schedule of GDT [106]. During finetuning, we use SGD with initial learning rate 0.0025, which we gradually warm up to 0.02 in the first 2 epochs The weight decay is set to 0.005 and momentum to 0.9. We use a mini-batch size of 32 and train for 12 epochs with the learning rate multiplied by 0.05 at 6 and 10 epochs. For training, we randomly sample 1s clips per video, and during evaluation, we uniformly sample 10 clips from each video and apply 3-crop evaluation as in [41].

### 4.2. Comparison experiments and ablations

**Cropping augmentation.** In Tab. 1a, we ablate the importance of spatial augmentation in learning video representations. We compare our proposed Feature Crop augmentation, $C_{\bar{B}}$, to the recently proposed Multi-Crop augmentation strategy [20] and other baseline approaches. Multi-Crop has proven to be effective in image self-supervised learning because it forces the model to learn local-to-global associations, by explicitly enforcing invariance between features of large-crops and those of multiple small crops. While effective, it can be particularly computationally intensive, which, with our hardware, limits its use to only two large crops and one small crop when applied to video representation learn-

**Table 1: Comparison experiments and ablations.** We compare key parameters and settings of our proposed method. We report results model performance at epoch 100 and with 30 frames and without transformer unless noted otherwise.

| Cropping-strategy | Resolution | GPU-h/epoch | Acc. |
|---|---|---|---|
| Default | $1\times112^2$ | 17.3 | 54.0 |
| Two RGB Crops | $2\times112^2$ | 29.3 | 58.6 |
| Multi RGB Crops [20] | $2\times112^2 + 1\times96^2$ | 46.7 | 59.3 |
| Ours (Feature Crop) | $2\times112^2 +$ latent | 29.3 | **60.4** |

**(a) Cropping** yields benefits but requires more compute. Our feature crops are efficient *and* outperform [20]. Note that all models are trained for 100 epochs.

| $l$-Spatial size | | $l$-Temporal size | | Acc. |
|---|---|---|---|---|
| $M$ | $S$ | $M$ | $S$ | |
| $1\times7^2$ | | $1\times4$ | | 54.0 |
| $1\times6^2$ | $2\times4^2$ | $1\times4$ | | 59.9 |
| $1\times6^2$ | $2\times4^2$ | $2\times3$ | $1\times2$ | 58.4 |
| $2\times6^2$ | $4\times4^2$ | $2\times3$ | $1\times2$ | **60.4** |

**(b) Feature crops.** Heavier augmentations in latent ($l$) space and time lead to better representations.

| Pretraining | Finetuning | Acc |
|---|---|---|
| $\mathcal{P}_t$ | $\mathcal{P}_t$ | 54.0 |
| $\mathcal{P}_t$ | $\mathcal{P}_{\text{transf}}$ | 54.6 |
| $\mathcal{P}_{\text{transf}}$ | $\mathcal{P}_t$ | 52.1 |
| $\mathcal{P}_{\text{transf}}$ | $\mathcal{P}_{\text{transf}}$ | **60.3** |

**(c) Pooling.** Compared to Average-Pooling ($\mathcal{P}_t$), Transformer-based pooling ($\mathcal{P}_{\text{transf}}$) gives stronger performance.

| Transf.? | Layers | Params | GFLOPS | Acc. |
|---|---|---|---|---|
| ✗ | 0 | 37.2M | 77.7 | 54.0 |
| ✗ | 0 | 42.8M | 80.0 | 57.3 |
| ✓ | 2 | 42.4M | 77.8 | **60.3** |
| ✓ | 4 | 47.7M | 77.8 | 58.3 |

**(d) Architecture.** Using up to two transformer layers gives gains, not due to more trainable parameters.

| $C_{\text{space}}$ | $C_{\text{time}}$ | T? | Acc. |
|---|---|---|---|
| ✗ | ✗ | ✗ | 54.0 |
| ✓ | ✗ | ✗ | 59.9 |
| ✓ | ✓ | ✗ | 60.4 |
| ✓ | ✓ | ✓ | **62.0** |

**(e) Combined gains.** Feature crop in space $C_{\text{space}}$ and time $C_{\text{time}}$ and transformer pooling (T) add cumulative benefits.

| Method | **RGB-Crops** | | | **Multi-scale RGB-Crops** | | | **Feature Crops** | |
|---|---|---|---|---|---|---|---|---|
| | 1x | 2x | 4x* | 2x112 + 1x96 | 2x112 + 2x96 | 2x112 + 6x96* | (1x7, 1x4) | (2x6 + 4x4, 2x3 + 1x2) |
| GPU-h/epoch | 17.3 | 29.3 | 60.0 | 46.7 | 53.3 | 100.7 | 29.3 | 30.0 |

**(f) Speed.** Input-crops are slow: ∗ methods require reducing batch sizes (see Appendix) as activations do not fit on GPU.

ing. Our proposed Feature Crop is not only more efficient, but outperforms Multi-Crop by 1.1% when the learned representation is applied to action classification in HMDB-51. By cropping in feature space, we achieve a similar effect but can increase the number of small crops from 1 to 6 without increasing compute time.

**Feature crop parameters.** In Tab. 1b, we study the parameters of our Feature Cropping approach. We find that even our basic variant, which does one medium $6\times6$ crop and two $4\times4$ small crops (by cropping a $7\times7$ tensor) increases performance by nearly 6%, which is a relative improvement of more than 10%. If we further increase the number of crops in time and space, the performance increases from 59.9% to 60.4%.

**Pooling Function.** In Tab. 1c, we test temporal aggregation. We find that using a shallow transformer significantly outperforms simple average pooling by more than 5%; however, transformer pooling must be used both for pre-training the representation and for finetuning it on the target dataset.

**Transformer architecture.** In Tab. 1d, we test variants of the transformer architecture, including ablating iit altogether. We find that temporal modelling as measured by downstream performance peaks at two layers, likely due to optimization difficulties of deeper transformers with SGD. We also compare to a model with approximately the same number of parameters as our 2-layer transformer (achieved by increasing the networks' last block's hidden dimension

to 640). We find that the transformer still yields gains of 3%, indicating that it not the number of parameters but the modelling of time that is crucial for strong performance.

**Combining Feature Crops and Transformer Pooling.** In Tab. 1e, we show that combining Feature Crops in space and time, and then adding transformer pooling yield additive gains, with the best result obtained by combining all effects (which corresponds to **STiCA**). This shows that space-time augmentations and transformer pooling are complementary.

**Cropping efficiency.** In Tab. 1f, we compare training times (normalized to GPUs×hours) for Kinetics-400 epochs for the various spatial crops considered. We make two observations: First, the compute cost of RGB crops scales proportionally to their number because a full forward pass is required for each crop. Second, using a larger number of RGB crops eventually requires to decrease the batch size, which increases significantly the training time. In contrast, the cost of Feature Crop remains roughly constant no matter the number of crops.

### 4.3. Temporal Context and Multi-modality

**Length of temporal context.** In Tab. 2, we show the importance of leveraging longer context to improve video self-supervised representation learning. Similar to the supervised regime [134, 139], we observe improved accuracy as we increase the number of frames used during pretraining and fine-tuning. More importantly, the transformer pooling

| Frames | | Accuracy | |
|---|---|---|---|
| Pretrain | Finetune | GAP | Transf. |
| 30 | 30 | 54.0 | **60.3** |
| 60 | 60 | 62.4 | **66.1** |
| 90 | 90 | 58.0 | **66.9** |

**Table 2: Temporal context.** We report results with different number of frames on finetuning accuracy.

| $\lambda_{va}$ | $\lambda_{vv}$ | F. Crop? | Acc. |
|---|---|---|---|
| 0 | 1 | No | 43.3 |
| 1 | 0 | No | 54.0 |
| 0.5 | 0.5 | No | 58.6 |
| 0.5 | 0.5 | Yes | **60.3** |

**Table 3: Loss.** Combining within-modal and cross-modal loss with Feature-crops is key.

| Method | Architecture | Dataset | Top-1 Acc% HMDB | UCF |
|---|---|---|---|---|
| Supervised | R(2+1)D-18 | K-400 | 70.4 | 95.0 |
| SeLaVi [9] | R(2+1)D-18 | K-400 | 47.1 | 83.1 |
| TempTrans [63] | R3D-18 | K-400 | 49.8 | 79.3 |
| PEMT [80] | SlowFast | K-400 | - | 85.2 |
| XDC [6] | R(2+1)D-18 | K-400 | 52.6 | 86.2 |
| MemDPC [53] | R-2D3D | K-400 | 54.5 | 86.1 |
| AVSF [143] | AVSF | K-400 | 54.6 | 87.0 |
| AVTS [74] | MC3-18 | K-400 | 56.9 | 85.8 |
| CPD [85] | R3D-50 | K-400 | 57.7 | 88.7 |
| AVID [97] | R(2+1)D-18 | K-400 | 60.8 | 87.5 |
| GDT [106] | R(2+1)D-18 | K-400 | 60.0 | 89.3 |
| ACC [90] | R3D-18 | K-400 | 61.8 | 90.2 |
| GLCM [91] | R3D-18 | K-400 | 61.9 | 91.2 |
| CoCLR [54] | S3D | K-400 | 62.9 | 90.6 |
| CVLR [112][3] | R3D-50 | K-400 | 66.7 | 92.2 |
| **Ours: STiCA** | R(2+1)D-18 | K-400 | **67.0** | **93.1** |
| SeLaVi [9] | R(2+1)D-18 | VGGS | 53.1 | 87.7 |
| Speech2Act [98] | S3D-G | Movie | 58.1 | – |
| DynamoNet [34] | ResNext101 | Y8M | 58.6 | 87.3 |
| MIL-NCE [93] | S3D | HT | 61.0 | 91.3 |
| AVTS [74] | MC3-18 | AS | 61.6 | 89.0 |
| AVID [97] | R(2+1)D-18 | AS | 64.7 | 91.5 |
| Textual [123] | S3D-G | WVT-70M | 65.3 | 90.3 |
| GDT [106] | R(2+1)D-18 | AS | 66.1 | 92.5 |
| ACC [90] | R(2+1)D-18 | AS | 67.2 | 93.5 |
| ELo [110] | R(2+1)D-50 | Y2M | 67.4 | 93.8 |
| XDC [6] | R(2+1)D-18 | IG65M | 68.9 | **95.5** |
| GDT [106] | R(2+1)D-18 | IG65M | 72.8 | 95.2 |
| MMV [5] | TSM-50x2 | AS+HT | **75.0** | 95.2 |

**Table 4: Comparison to SoTA for action recognition.** Dashed line indicates position of our Kinetics-400 model in comparison to models trained with many more videos. We follow standard evaluation protocol across 3-folds. For linear evaluation results see Tab. 7.

| Recall @ | UCF | | | HMDB | | |
|---|---|---|---|---|---|---|
| | 1 | 5 | 20 | 1 | 5 | 20 |
| MemDPC [53] | 20.2 | 40.4 | 64.7 | 7.7 | 25.7 | 57.7 |
| VSP [25] | 24.6 | 41.9 | 62.7 | 10.3 | 26.6 | 76.8 |
| SeLaVi [9] | 52.0 | 68.6 | 84.5 | 24.8 | 47.6 | 75.5 |
| CoCLR [54] | 55.9 | 70.8 | 82.5 | 26.1 | 45.8 | 69.7 |
| GDT [106] | 57.4 | 73.4 | 88.1 | 25.4 | 51.4 | 75.0 |
| **Ours: STiCA** | **59.1** | **76.2** | **88.1** | **26.3** | **49.2** | **76.4** |

**Table 5: Comparison to SoTA for retrieval.** Nearest neighbor action retrieval performance $@k = \{1, 5, 20\}$.

CLR [24] to the video domain, by learning invariances to spatial augmentations of two large-crops. Compared to this, the cross-modal baseline (row 2) already achieves gains of more than 10%. While adding a within-modal invariance adds another 4.6%, we find that the best performance is obtained with our feature crops, adding another 1.7% in performance and showing its unique potential to supplement cross-modal signals.

### 4.4. Comparison with the state of the art

**Video Action Recognition.** In Tab. 4, we evaluate our pretraining approach on the standard HMDB-51 and UCF-101 action recognition benchmarks after pretraining on the Kinetics-400 dataset. Firstly, we find our model outperforming the similar NCE-based GDT [106] model by 7.0% and 3.8% on HMDB-51 and UCF-101. We further significantly outperform the current state-of-the art methods CoCLR [54] by 4.1% and 2.5% and CVLR [112] by 2.6% and 1.0% on HMDB-51 and UCF-101, respectively. Even more impressively, our approach is able to out-perform most prior works that use AudioSet [44] pre-training, which is around 10× larger than Kinetics-400. This shows how effective and data-efficient our approach is, significantly closing the gap to supervised learning.

**Video Action Retrieval.** Lastly, we directly evaluate the transfer-ability of our pretrained representations on action retrieval on UCF-101 and HMDB-51. Similarly to full fine-tuning setting, we outperform all prior works.

## 5. Conclusion

We have address two shortcomings of current self-supervised video representation learning: insufficient spatial invariance, especially compared to the image domain, and inadequate modelling of time. We have introduced STiCA, improving spatial invariance at very little cost by implementing cropping in feature space, and improving modelling of time via a shallow transformer. Our method brings self-supervised video representation learning one step closer to the supervised case, providing significant gains w.r.t. the state-of-the-art.

layer is better able to exploit this additional context, outperforming average pooling by over 4% for all frame lengths. Notably, there is a drop in performance when using GAP for extremely long contexts (90 frames).

**Loss.** Lastly, in Tab. 3, we study the effect of combining multi-modal learning signals with our contributions. In the first row, we have the baseline of naively extending Sim-

---
[3]Concurrent work.

# References

[1] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. *Interspeech*, 2018.

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Deep lip reading: A comparison of models and an online application. 2018.

[4] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *ECCV*, 2020.

[5] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020.

[6] Humam Alwassel, Bruno Korbar, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020.

[7] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017.

[8] Relja Arandjelović and Andrew Zisserman. Objects that sound. In *ECCV*, 2018.

[9] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020.

[10] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *ICLR*, 2020.

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

[12] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016.

[13] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.

[14] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, 2020.

[15] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?, 2021.

[16] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*, 2018.

[17] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

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

[19] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

[20] 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.

[21] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[22] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020.

[23] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[24] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[25] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *arXiv preprint arXiv:2003.02692*, 2020.

[26] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. *CVPR*, 2017.

[27] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016.

[28] E. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. In *CVPRW*, 2020.

[29] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2020.

[30] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[32] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations, 2020.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[34] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *ICCV*, 2019.

[35] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[36] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *NeurIPS*, 2020.

[37] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolu-

tional networks for visual recognition and description. In *CVPR*, 2015.

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[39] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9), 2015.

[40] D. Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *CVPR*, 2020.

[41] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

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

[43] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020.

[44] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[45] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

[46] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

[47] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019.

[48] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan C. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.

[49] 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.

[50] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

[51] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[52] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCVW*, 2019.

[53] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, 2020.

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

[55] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCVW*, 2017.

[56] David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *ECCV*, 2018.

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

[58] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv.cs*, abs/1911.05722, 2019.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[60] Di Hu, Feiping Nie, and Xuelong Li. Deep multimodal clustering for unsupervised audiovisual learning. In *CVPR*, 2019.

[61] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. Timeception for complex action recognition. In *CVPR*, 2019.

[62] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018.

[63] S. Jenni, Givi Meishvili, and P. Favaro. Learning video representations by transforming time. In *ECCV*, 2020.

[64] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

[65] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018.

[66] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *NeurIPS*, 2020.

[67] Yash Kant, Dhruv Batra, Peter Anderson, Alex Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multimodal transformers for textvqa. In *ECCV*, 2020.

[68] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[69] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[70] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019.

[71] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019.

[72] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[73] Bruno Korbar, Fabio Petroni, Rohit Girdhar, and Lorenzo Torresani. Video understanding as machine translation, 2020.

[74] Bruno Korbar, Du Tran, and Lorenzo Torresani. Co-operative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[75] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, 2019.

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[77] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[78] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *ECCV*, 2020.

[79] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017.

[80] Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas Breuel, Jan Kautz, and Yale Song. Parameter efficient multimodal transformers for video representation learning. In *International Conference on Learning Representations*, 2021.

[81] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[82] Hao Li, Xiaopeng Zhang, Ruoyu Sun, Hongkai Xiong, and Qi Tian. Center-wise local image mixture for contrastive representation learning, 2020.

[83] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

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

[85] Tianhao Li and Limin Wang. Learning spatiotemporal features via video and text pair discrimination. *arXiv preprint arXiv:2001.05691*, 2020.

[86] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.

[87] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

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

[89] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks, 2017.

[90] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Learning audio-visual representations with active contrastive coding, 2020.

[91] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Contrastive self-supervised learning of global-local audio-visual representations, 2021.

[92] Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from video and text via large-scale discriminative clustering. In *Proc. ICCV*, 2017.

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

[94] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[95] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

[96] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016.

[97] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. *arXiv preprint arXiv:2004.12943*, 2020.

[98] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *CVPR*, 2020.

[99] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network, 2021.

[100] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[101] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[102] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.

[103] D. Park, William Chan, Y. Zhang, Chung-Cheng Chiu, Barret Zoph, E. D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. In *INTERSPEECH*, 2019.

[104] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018.

[105] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[106] Mandela Patrick, Yuki Markus Asano, Ruth Fong, João F. Henriques, G. Zweig, and A. Vedaldi. Multi-modal self-supervision from generalized data transformations. *ArXiv*, abs/2003.04298, 2020.

[107] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning, 2020.

[108] Karol J. Piczak. Environmental sound classification with convolutional neural networks. *MLSP*, 2015.

[109] Karol J. Piczak. Esc: Dataset for environmental sound classification. In *ACM Multimedia*, 2015.

[110] AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo.

Evolving losses for unsupervised video representation learning. In *CVPR*, 2020.

[111] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.

[112] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, H. Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. 2020.

[113] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[114] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[115] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[116] Guido Roma, Waldo Nogueira, and Perfecto Herrera. Recurrence quantification analysis features for environmental sound recognition. *WASPAA*, 2013.

[117] Hardik B. Sailor, Dharmesh M Agrawal, and Hemant A Patil. Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification. In *INTERSPEECH*, 2017.

[118] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, 2020.

[119] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[120] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[121] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D. Plumbley. Detection and classification of acoustic scenes and events. *TM*, 2015.

[122] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 2015.

[123] Jonathan C. Stroud, D. Ross, Chen Sun, Jun Deng, R. Sukthankar, and C. Schmid. Learning video representations from textual web supervision. *ArXiv*, abs/2007.14937, 2020.

[124] 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 *ICLR*, 2020.

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

[126] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.

[127] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[128] Hao Tan and Mohit Bansal. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *EMNLP*, 2020.

[129] V Terrance and W Taylor Graham. Dataset augmentation in feature space. In *Proceedings of the international conference on machine learning (ICML), workshop track*, 2017.

[130] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[131] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *NeurIPS*, 2020.

[132] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[133] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[134] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term Temporal Convolutions for Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[135] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[136] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019.

[137] Jiangliu Wang, Jianbo Jiao, and Y. Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, 2020.

[138] Limin Wang, Yuanjun Xiong, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[139] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[140] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018.

[141] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.

[142] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[143] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slow-fast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[144] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. In *ECCV*, 2018.

[145] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019.

[146] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[147] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models?, 2020.

[148] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[149] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[150] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *ICCV*, 2019.

[151] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018.

[152] Bolei Zhou, Alex Andonian, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.

[153] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

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