Self-Supervised Audio-Visual Representation Learning with Relaxed Cross-Modal Synchronicity

Pritam Sarkar[1, 2], **Ali Etemad**[1]
[1] Queen's University, Canada
[2] Vector Institute
{pritam.sarkar, ali.etemad}@queensu.ca

Abstract

We present **CrissCross**, a self-supervised framework for learning audio-visual representations. A novel notion is introduced in our framework whereby in addition to learning the intra-modal and standard 'synchronous' cross-modal relations, CrissCross also learns 'asynchronous' cross-modal relationships. We perform in-depth studies showing that by relaxing the temporal synchronicity between the audio and visual modalities, the network learns strong generalized representations useful for a variety of downstream tasks. To pretrain our proposed solution, we use 3 different datasets with varying sizes, Kinetics-Sound, Kinetics400, and AudioSet. The learned representations are evaluated on a number of downstream tasks namely action recognition, sound classification, and action retrieval. Our experiments show that CrissCross either outperforms or achieves performances on par with the current state-of-the-art self-supervised methods on action recognition and action retrieval with UCF101 and HMDB51, as well as sound classification with ESC50 and DCASE. Moreover, CrissCross outperforms fully-supervised pretraining while pretrained on Kinetics-Sound.

1 Introduction

In recent years, self-supervised learning has shown great promise in learning strong representations without human-annotated labels (Chen et al. 2020; Chen and He 2021; Caron et al. 2018), and emerged as a strong competitor for fully-supervised pretraining. There are a number of benefits to such methods. Firstly, they reduce the time and resources required for expensive human annotations and allow researchers to directly use large uncurated datasets for learning meaningful representations. Moreover, the models trained in a self-supervised fashion learn more abstract representations, which are useful for a variety of downstream tasks without needing to train the models from scratch. Given the abundance of videos, their spatio-temporal information-rich nature, and the fact that in most cases they contain both audio and visual streams, self-supervised approaches are strong alternatives to fully-supervised methods for video representation learning. Moreover, the high dimensionality and multi-modal nature of videos make them difficult to annotate, further motivating the use of self-supervision.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The common and standard practice in self-supervised audio-visual representations learning is to learn intra-modal and cross-modal relationships between the audio and visual streams by maintaining tight temporal synchronicity between the two modalities (Alayrac et al. 2020; Korbar, Tran, and Torresani 2018; Alwassel et al. 2020; Asano et al. 2020). Yet, the impact of learning temporally asynchronous cross-modal relationships in the context of self-supervised learning has not been explored. This notion deserves deeper exploration as learning such temporally asynchronous cross-modal relationships may in fact result in increased invariance and distinctiveness in the learned representations.

In this study, in an attempt to explore the notion above, we present **CrissCross**, a self-supervised framework to learn robust generalized audio-visual representations from videos. CrissCross is built upon SimSiam (Chen and He 2021) to jointly learn self-supervised audio-visual representations through a mixture of intra- and cross- modal optimizations. In addition to learning intra-modal and standard synchronous cross-modal relations, CrissCross introduces the novel idea of learning cross-modal representations through *relaxing* time-synchronicity between corresponding audio and visual segments. We refer to this as 'asynchronous cross-modal' optimization, a concept that has not been explored in prior works. We use 3 datasets of different sizes: Kinetics-Sound (Arandjelovic and Zisserman 2017), Kinetics400 (Kay et al. 2017), and AudioSet (Gemmeke et al. 2017), to pretrain CrissCross. We evaluate CrissCross on different downstream tasks, namely action recognition, sound classification, and action retrieval. We use 2 popular benchmarks UCF101 (Soomro, Zamir, and Shah 2012) and HMDB51 (Kuehne et al. 2011) to perform action recognition and retrieval, while ESC50 (Piczak 2015) and DCASE (Stowell et al. 2015) are used for sound classification.

The key contributions of this work are as follows:

- We present a novel framework for multi-modal self-supervised learning by relaxing the audio-visual temporal synchronicity to learn effective generalized representations. Our method is simple, data efficient and less resource intensive, yet learns robust multi-modal representations for a variety of downstream tasks.
- We perform an in-depth study to explore the performance of the proposed framework and its major concepts. Moreover, we perform thorough analyses, both quantitatively

and qualitatively, in different setups, showing the benefit of learning asynchronous cross-modal relations.

- Comparing the performance of our method to prior works, CrissCross achieves state-of-the-arts on UCF101, HMDB, ESC50, and DCASE when pretrained on Kinetics400. Moreover, when trained with AudioSet, CrissCross achieves better or competitive performances versus the current state-of-the-arts.
- Lastly, when pretrained on the small-scale Kinetics-Sound (Arandjelovic and Zisserman 2017), CrissCross outperforms fully-supervised pretraining (Ma et al. 2020) by 1.4% and 7.4%, as well as prior self-supervised state-of-the-art (Ma et al. 2020) by 11.1% and 19.9% on UCF101 and HMDB51 respectively. To the best of our knowledge, very few prior works have attempted to pretrain on such small datasets, and in fact, this is the first time where self-supervised pretraining outperforms full supervision on action recognition in this setup.

We hope our proposed self-supervised method can motivate researchers to further explore the notion of *asynchronous* multi-modal representation learning. The codes, pretrained models, and supplementary material are available on the project website[1].

2 Related Work

2.1 Self-supervised Learning

Self-supervised learning aims to learn generalized representations of data without any human annotated labels through properly designed pseudo tasks (also known as pretext tasks). Self-supervised learning has recently drawn significant attention in different areas such as image (Chen et al. 2020; Chen and He 2021; Misra and Maaten 2020; Caron et al. 2020; Grill et al. 2020; Caron et al. 2018), video (Morgado, Vasconcelos, and Misra 2021; Morgado, Misra, and Vasconcelos 2021; Alwassel et al. 2020; Asano et al. 2020; Patrick et al. 2021a; Alayrac et al. 2020; Min et al. 2021), and wearable data (Sarkar and Etemad 2020b,a; Sarkar et al. 2020) analysis among others.

In self-supervised learning, the main focus of interest lies in designing novel pseudo-tasks to learn useful representations. We briefly mention some of the popular categories in the context of self-supervised video representation learning, namely, i) context-based, ii) generation-based, iii) clustering-based, and iv) contrastive learning-based. Various pretext tasks have been proposed in the literature exploring the spatio-temporal context of video frames, for example, temporal order prediction (Lee et al. 2017), puzzle solving (Kim, Cho, and Kweon 2019; Misra, Zitnick, and Hebert 2016; Ahsan, Madhok, and Essa 2019), rotation prediction (Jing et al. 2018), and others. Generation-based video feature learning methods refer to the process of learning feature representations through video generation (Vondrick, Pirsiavash, and Torralba 2016; Tulyakov et al. 2018; Saito, Matsumoto, and Saito 2017), video colorization (Tran et al. 2016), and frame or clip prediction (Mathieu, Couprie, and LeCun 2016; Reda et al. 2018; Babaeizadeh et al. 2018; Liang et al. 2017; Finn, Goodfellow, and Levine 2016), among a few others. Clustering-based approaches (Alwassel et al. 2020; Asano et al. 2020) rely on self-labeling where data is fed to the network and the extracted feature embeddings are clustered using a classical clustering algorithm such as k-means, followed by using the cluster assignments as the pseudo-labels for training the neural network. The key concept of contrastive learning (Chen and He 2021; Misra and Maaten 2020; Grill et al. 2020; Caron et al. 2020; Morgado, Vasconcelos, and Misra 2021; Patrick et al. 2021a) is that in the embedding space, 'positive' samples should be similar to each other, and 'negative' samples should have discriminative properties. Using this concept, several prior works (Morgado, Vasconcelos, and Misra 2021; Morgado, Misra, and Vasconcelos 2021; Patrick et al. 2021a; Ma et al. 2020) have attempted to learn representations by minimizing the distance between positive pairs and maximizing the distance between negative pairs.

2.2 Audio-Visual Representation Learning

Typically in multi-modal self-supervised learning, multiple networks are jointly trained on the pseudo tasks towards maximizing the mutual information between multiple data streams (Alwassel et al. 2020; Morgado, Vasconcelos, and Misra 2021; Korbar, Tran, and Torresani 2018; Xu et al. 2019; Wang et al. 2021; Khare, Parthasarathy, and Sundaram 2021; Siriwardhana et al. 2020). Following, we briefly discuss some of the prior works (Korbar, Tran, and Torresani 2018; Alwassel et al. 2020; Morgado, Vasconcelos, and Misra 2021; Ma et al. 2020) on audio-visual representation learning. A multi-modal self-supervised task is introduced in AVTS (Korbar, Tran, and Torresani 2018), leveraging the natural synergy between audio-visual data. The network is trained to distinguish whether the given audio and visual sequences are 'in sync' or 'out of sync'. In XDC (Alwassel et al. 2020), the authors introduce a framework to learn cross-modal representations through a self-labeling process. Specifically, cross-modal pseudo-labeling is performed where the pseudo-labels computed from audio embeddings are used to train the visual backbone, while the pseudo-labels computed using visual embeddings are used to train the audio network. A self-supervised learning framework based on contrastive learning is proposed in AVID (Morgado, Vasconcelos, and Misra 2021) to learn audio-visual representations from videos. AVID performs instance discrimination as the pretext task by maximizing the cross-modal agreement of the audio-visual segments in addition to visual similarity. Though earlier works focus on learning cross-modal relations while maintaining a tight synchronicity between the audio and visual data, our proposed framework also considers asynchronous cross-modal relationships in addition to the standard synchronous relations.

3 Method

3.1 Approach

Let be given v, a sequence of visual frames, and a, the corresponding audio waveform. We can obtain n augmented views of v as $\{v_i\}_{i=0}^{n}$, and equal number of augmented

[1] https://pritamqu.github.io/CrissCross

views of a as $\{a_i\}_{i=0}^{n}$. A common way to learn individual representations from v and a is to minimize the embedding distances ($\mathcal{D}$) between the augmented views of the each modality as $\mathcal{L}_{vv} = \sum_{i,j=0,i\neq j}^{n} \mathcal{D}(v_i, v_j)$ and $\mathcal{L}_{aa} = \sum_{i,j=0,i\neq j}^{n} \mathcal{D}(a_i, a_j)$ respectively in a self-supervised setting (Caron et al. 2020; Bardes, Ponce, and LeCun 2021; Chen and He 2021; Grill et al. 2020; Niizumi et al. 2021). Further, to learn multi-modal representations from $\{v, a\}$, a standard technique is to simply optimize a joint intra-modal loss $\mathcal{L}_{intra} = \mathcal{L}_{vv} + \mathcal{L}_{aa}$. Prior works (Alwassel et al. 2020; Morgado, Vasconcelos, and Misra 2021; Morgado, Misra, and Vasconcelos 2021) have demonstrated that in addition to $\mathcal{L}_{intra}$, a cross-modal optimization can be performed directly across visual and audio segments to further learn strong joint representations as $\mathcal{L}_{av} = \sum_{i=0}^{n} \mathcal{D}(a_i, v_i)$.

All of these learning procedures maintain a tight synchronicity between the two modalities, given that both a_i and v_i are segmented from the same timestamps. We conjecture, however, that *relaxing the synchronicity between modalities by a reasonable margin will enable more generalized representations to be learned* across time, to achieve better and more robust performance. Accordingly, we introduce asynchronous cross-modal loss $\mathcal{L}_{async}$, which exploits the relationship between audio and visual segments sampled at different timestamps. We define the final objective as $\mathcal{L}_{CrissCross}$ which exploits the combination of $\mathcal{L}_{intra}$, synchronous $\mathcal{L}_{av}$ (which we refer to as $\mathcal{L}_{sync}$), and $\mathcal{L}_{async}$ in an attempt to learn more generalized representations. While we present the detailed experiments and analysis of our proposed approach in the subsequent sections of the paper, here we perform a quick visualization to demonstrate the benefits of this concept. Figure 1 depicts the distributions of representations learned with and without $\mathcal{L}_{async}$, demonstrating that indeed relaxing the tight synchronicity helps in widening the distribution of the learned representations which could result in improved performance in a wide variety of downstream tasks.

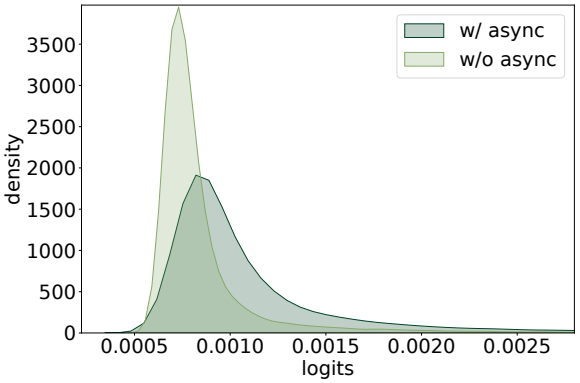


Figure 1: Distribution of the learned representations with and without the asynchronous cross-modal optimization.

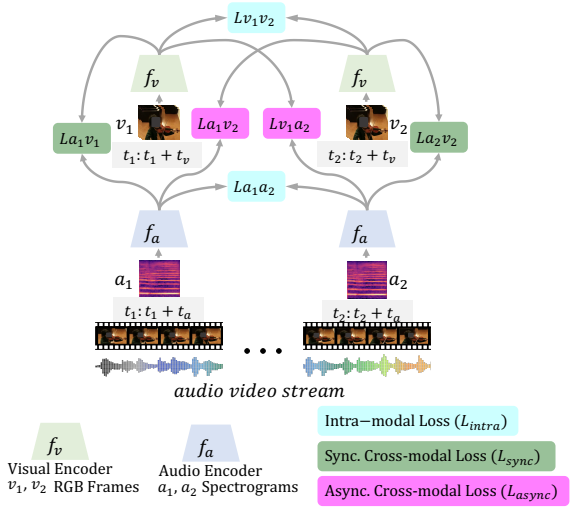


Figure 2: Our proposed framework. CrissCross learns strong audio-visual representations by exploiting intra-modal, as well as, sync. and async. cross-modal relations.

3.2 Training Objective

To accomplish the notion above, let's define two neural networks, a visual encoder f_v and an audio encoder f_a. Here, f_v and f_a are composed of convolutional backbones and MLP projection heads. Moreover, we adopt a Siamese (Bromley et al. 1993) representation learning setup, where the networks share weights on two or more inputs. Next, We obtain two augmented views of $v = \{\mathrm{v_t}\}_{\mathrm{t=0}}^{\mathrm{T}}$, denoted by v_1 and v_2, defined as $\{\mathrm{v_t}\}_{\mathrm{t=t_1}}^{\mathrm{t_1+t_v}}$ and $\{\mathrm{v_t}\}_{\mathrm{t=t_2}}^{\mathrm{t_2+t_v}}$ respectively. Here, v_1 and v_2 have a duration of $\mathrm{t_v}$, and are sampled at times $\mathrm{t_1}$ and $\mathrm{t_2}$ respectively. Note that v_1 and v_2 are augmented differently. Similarly, two augmented views of $a = \{\mathrm{a_t}\}_{\mathrm{t=0}}^{\mathrm{T}}$ can be obtained as a_1 and a_2 as $\{\mathrm{a_t}\}_{\mathrm{t=t_1}}^{\mathrm{t_1+t_a}}$ and $\{\mathrm{a_t}\}_{\mathrm{t=t_2}}^{\mathrm{t_2+t_a}}$, respectively. Next, to learn intra-modal representations, the distance between $f_v(v_1)$ and $f_v(v_2)$, as well as, $f_a(a_1)$ and $f_a(a_2)$ can be minimized to train f_v and f_a respectively. However, such a naive approach would lead to mode collapse as pointed out in (Grill et al. 2020; Niizumi et al. 2021; Chen and He 2021; Caron et al. 2020). To tackle this, we follow the technique proposed in (Chen and He 2021). In particular, we minimize the cosine embedding distance $\mathcal{D}$ of two output vectors p and $\mathtt{S}(z)$, where p is the output vector obtained from the predictor head and z represents the output vector obtained from the feature encoder followed by the `stop-gradient` operation. Here, the predictor head consists of an MLP head, which is used as an identity mapping, while the `stop-gradient` operation prevents the model from collapsing to a degenerated solution (Chen and He 2021). Here, $\mathcal{D}$ is defined as:

$$\mathcal{D}(p, z) = -\frac{p}{||p||_2} \cdot \frac{z}{||z||_2} \,. \tag{1}$$

We use h_v and h_a as the predictor heads corresponding to visual and audio representations. Next, we obtain p_{v_1} and z_{v_2} as $h_v(f_v(v_1))$ and $\mathtt{S}(f_v(v_2))$. Similarly, p_{a_1} and z_{a_2} are obtained as $h_a(f_a(a_1))$ and $\mathtt{S}(f_a(a_2))$. To calculate the symmetrized loss, we further obtain p_{v_2} and z_{v_1}, as well as, p_{a_2} and z_{a_1}. Therefore, to learn the intra-modal relations, we op-

timize the intra-modal loss $\mathcal{L}_{intra}$ defined as:

$$\mathcal{L}_{intra} = (\frac{1}{2}\mathcal{D}(p_{v_1}, \mathtt{S}(z_{v_2})) + \frac{1}{2}\mathcal{D}(p_{v_2}, \mathtt{S}(z_{v_1})) + \frac{1}{2}\mathcal{D}(p_{a_1}, \mathtt{S}(z_{a_2})) + \frac{1}{2}\mathcal{D}(p_{a_2}, \mathtt{S}(z_{a_1})))/2 \,. \tag{2}$$

Next, to learn synchronous cross-modal relations, we optimize the synchronous cross-modal loss $\mathcal{L}_{sync}$, defined as:

$$\mathcal{L}_{sync} = (\frac{1}{2}\mathcal{D}(p_{v_1}, \mathtt{S}(z_{a_1})) + \frac{1}{2}\mathcal{D}(p_{a_1}, \mathtt{S}(z_{v_1})) + \frac{1}{2}\mathcal{D}(p_{v_2}, \mathtt{S}(z_{a_2})) + \frac{1}{2}\mathcal{D}(p_{a_2}, \mathtt{S}(z_{v_2})))/2 \,. \tag{3}$$

Additionally, based on our earlier intuition, to relax the temporal synchronicity, we minimize the distance between the audio and visual segments originated from different timestamps. We define asynchronous cross-modal loss $\mathcal{L}_{async}$ as:

$$\mathcal{L}_{async} = (\frac{1}{2}\mathcal{D}(p_{v_1}, \mathtt{S}(z_{a_2})) + \frac{1}{2}\mathcal{D}(p_{a_2}, \mathtt{S}(z_{v_1})) + \frac{1}{2}\mathcal{D}(p_{v_2}, \mathtt{S}(z_{a_1})) + \frac{1}{2}\mathcal{D}(p_{a_1}, \mathtt{S}(z_{v_2})))/2 \,. \tag{4}$$

Finally, to exploit intra-modal, as well as, synchronous and asynchronous cross-modal relations we define the final objective function as:

$$\mathcal{L}_{CrissCross} = \frac{1}{3}(\mathcal{L}_{intra} + \mathcal{L}_{sync} + \mathcal{L}_{async}) \,. \tag{5}$$

We present the proposed CrissCross framework in Figure 2. Please note, for the sake of simplicity, we omit showing the stop-grad and predictor head connections in Figure 2. We present the pseudocode in Appendix A.

4 Experiments and Results

The details of the experiment setup and the findings of our thorough ablation studies investigating the major concepts of our proposed framework are presented here. Additionally, we extensively investigate a wide range of audio-visual augmentation techniques capable of learning strong audio-visual representations within our framework, the details are as follows.

4.1 Experiment Setup

Following the standard practice among the prior works (Morgado, Vasconcelos, and Misra 2021; Alwassel et al. 2020; Asano et al. 2020; Patrick et al. 2021a; Ma et al. 2020), we use Kinetics-Sound, Kinetics400, and AudioSet for pretraining. Additionally, Kinetics400, UCF101, HMDB51, ESC50 and DCASE are used for downstream evaluation. We use R(2+1)D (Tran et al. 2018) and ResNet (He et al. 2016) as the visual and audio backbones. To pretrain the network in a self-supervised fashion with audio-visual inputs, we downsample the visual streams to 16 frames per second and feed 8 frames of resolution 112^2 to the visual encoder. Next, we downsample the audio signals to 16kHz, and segment them into 2-second segments. We transform the segmented raw audio waveforms to mel-spectrograms using 80 mel filters, we set the hop size as 10 milliseconds and FFT window length as 1024. Finally, we feed spectrograms of shape 80×200 to the audio encoder. We use Adam (Kingma and Ba 2015) optimizer with a cosine learning rate scheduler (Loshchilov and Hutter 2017) to pretrain the encoders and use a fixed learning rate to train the predictors. Please note that during the design exploration, we use Kinetics-Sound for pretraining, while the downstream evaluations are performed on UCF101 and ESC50 unless stated otherwise. We perform linear evaluations using 8 frames of visual input and 2 seconds of audio input. Next, a linear SVM classifier is trained using the extracted features, and report the top-1 accuracy for sample-level predictions. We provide the additional details of the experiment setup, datasets, architectures, and evaluation protocols in the Appendix.

Method	**UCF101**	**ESC50**
$\mathcal{L}_{v_1v_2}$	69.1	–
$\mathcal{L}_{a_1a_2}$	–	62.0
$\mathcal{L}_{intra}$	69.7	71.8
$\mathcal{L}_{sync}$	70.1	75.8
$\mathcal{L}_{async}$	69.1	74.8
$\mathcal{L}_{sync} + \mathcal{L}_{intra}$	73.8	78.0
$\mathcal{L}_{sync} + \mathcal{L}_{async}$	69.1	74.8
$\mathcal{L}_{async} + \mathcal{L}_{intra}$	72.4	75.3
$\mathcal{L}_{v_1v_2} + \mathcal{L}_{sync} + \mathcal{L}_{async}$	71.3	78.5
$\mathcal{L}_{a_1a_2} + \mathcal{L}_{sync} + \mathcal{L}_{async}$	70.8	75.3
$\mathcal{L}_{\mathbf{CrissCross}}$	**74.8**	**79.0**

Table 1: We present the top-1 accuracy of CrissCross and its ablation variants, pretrained on Kinetics-Sound.

Pretrain	**Downstream**	**w/o $\mathcal{L}_{\mathbf{async}}$**	**w/ $\mathcal{L}_{\mathbf{async}}$**
KS	UCF101	73.8(↓ 1.0)	**74.8**
KS	ESC50	78.0(↓ 1.0)	**79.0**
K400	UCF101	75.8(↓ 4.1)	**79.9**
K400	ESC50	78.5(↓ 3.5)	**82.0**
K400	KS (a)	43.2(↓ 3.9)	**47.1**
K400	KS (v)	53.3(↓ 2.4)	**55.7**
K400	KS (a+v)	65.0(↓ 1.7)	**66.7**

Table 2: Impact of $\mathcal{L}_{async}$ optimization in different pretraining and evaluation setups. Here, K400: Kinetics400, KS: Kinetics-Sound.

4.2 Ablation Study

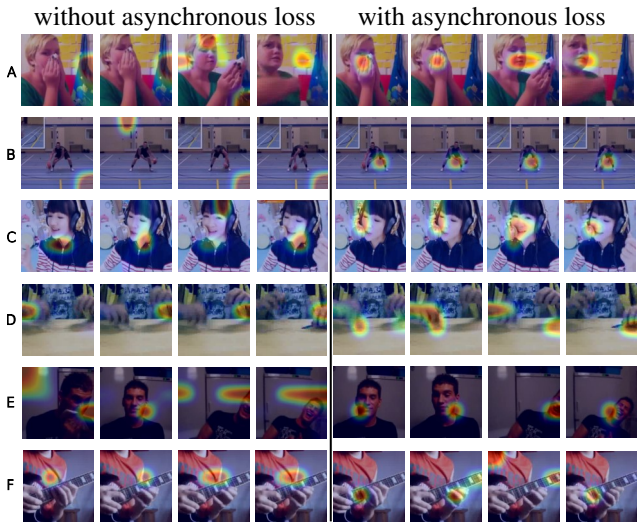
We present the ablation results in Tables 1 and 2 to show the improvements made by optimizing asynchronous cross-modal loss in addition to intra-modal and synchronous cross-modal losses. First, using Kinetics-Sound, we train the framework in uni-modal setups, denoted as $\mathcal{L}_{v_1v_2}$ and $\mathcal{L}_{a_1a_2}$. We report the top-1 accuracy of UCF101 and ESC50 as 69.1% and 62.0% respectively. Next, we train the network in a multi-modal setup, where we find that $\mathcal{L}_{sync}$ outperforms the other multi-modal variants including $\mathcal{L}_{intra}$ and $\mathcal{L}_{async}$, as well as, uni-modal baselines $\mathcal{L}_{v_1v_2}$ and $\mathcal{L}_{a_1a_2}$. Further study shows that combining all the multi-modal

losses improves the model performance. $\mathcal{L}_{CrissCross}$ outperforms $\mathcal{L}_{sync}$ by 4.7% and 3.2% on action recognition and sound classification, respectively.

Further, to study the effect of $\mathcal{L}_{async}$ in particular, we perform ablation studies using small-scale Kinetics-Sound and large-scale Kinetics400. We present the results in Table 2, where we observe that $\mathcal{L}_{async}$ improves the performance on both the pretraining datasets. In particular, while pretrained on Kinetics400, optimizing $\mathcal{L}_{async}$ in addition to $\mathcal{L}_{sync}$ and $\mathcal{L}_{intra}$ improves the performances by 4.1% and 3.5% on action recognition and sound classification respectively, showing the significance of asynchronous cross-modal optimization in a multi-modal setup. While pretrained on Kinetics-Sound, adding $\mathcal{L}_{async}$ improves the performances by 1% on both the UCF101 and ESC50. We interestingly find that learning asynchronous cross-modal loss significantly improves the model performance when pretrained on large-scale Kinetics400. Our intuition is that as Kinetics-Sound consists of a few hand-picked classes which are prominently manifested in both audio and visual modalities, the performance gain due to $\mathcal{L}_{async}$ is less prominent. However, Kinetics400 is considerably larger in scale and comprises highly diverse action classes which are not always very prominent both audibly and visually. It therefore benefits more from the generalized representations learned by asynchronous cross-modal optimization. Moreover, to demonstrate the benefit of optimizing $\mathcal{L}_{async}$ throughout the pretraining process, we present the top-1 accuracy vs. pretraining epoch in Figure 4. It shows that $\mathcal{L}_{async}$ significantly improves the model performance throughout the pretraining.

Multi-modal fusion. Next, we investigate if learning asynchronous cross-modal relations helps in multi-modal fusion. To test this, we use Kinetics-Sound as the downstream dataset and Kinetics400 as the pretraining dataset. We choose Kinetics-Sound for downstream evaluation as it consists of action classes that are represented prominently in both audio and visual domains. The results are presented in Table 2, where it is shown that learning asynchronous cross-modal relations improves multi-modal fusion by 1.7%. Additionally, we show the linear evaluation results obtained from the uni-modal feature representations for reference. It shows that optimizing $\mathcal{L}_{async}$ improves the action classification accuracy by 2.4% and 3.9% using visual and audio representations, respectively.

Qualitative analysis. Lastly, to perform a qualitative analysis on the impact of $\mathcal{L}_{async}$ we visualize the saliency maps obtained from the models when pretrained with and without the presence of the asynchronous loss. In this experiment, we directly use the models pretrained on Kinetics400 and use Grad-CAM (Omeiza et al. 2019) to visualize randomly selected samples from Kinetics400. A few examples are presented in Figure 3, where we observe that learning asynchronous relations helps the model focus better on the salient information. Specifically, we notice that optimizing $\mathcal{L}_{async}$ helps in correctly locating the sound sources on the visual streams, as shown by the examples of 'dribbling basketball', 'laughing', 'tapping guitar', etc.



A: blowing nose, B: dribbling basketball, C: singing, D: tapping pen, E: laughing, F: tapping guitar

Figure 3: Visualization of saliency maps while pretrained without (left) and with (right) asynchronous loss.

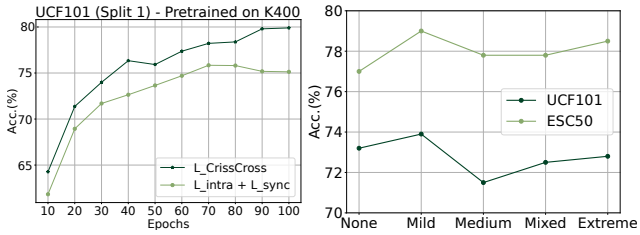


Figure 4: Left: Linear eval. top-1 acc. vs. pretraining epochs. Right: Exploring different temporal relaxation techniques.

4.3 Exploring Relaxed Time-synchronicity

Audio and visual modalities from the same source clip generally maintain a very strong correlation, which makes them suitable for multi-modal representation learning as one modality can be used as a supervisory signal for the other in a self-supervised setup. However, our intuition behind CrissCross is that these cross-modal temporal correlations do not necessarily need to follow a strict frame-wise coupling. Instead, we hypothesize that relaxing cross-modal temporal synchronicity to some extent can help in learning more generalized representations.

To facilitate this idea within CrissCross, we exploit 5 different temporal sampling methods to explore varying amounts of temporal synchronicity when learning cross-modal relationships. (*i*) *None:* where both the audio and visual segments are sampled from the exact same time window. (*ii*) *Mild:* where the two views of the audio-visual segments share 50% overlap amongst them. (*iii*) *Medium:* where adjacent frame sequences and audio segments are sampled. (*iv*) *Extreme:* in which we sample one view from the first half of the source clip, while the other view is sampled from the second half of the source clip. (*v*) *Mixed:*

	$\text{lr}_\text{p} = \text{lr}_\text{b}$	comm. pred.	2 layers proj.	default
UCF101	59.0	73.6	72.4	**74.8**
ESC50	62.3	75.3	75.0	**79.0**

Table 3: A comparative study of different predictor and projector setups. Here, lr_b: base LR and lr_p: pred LR

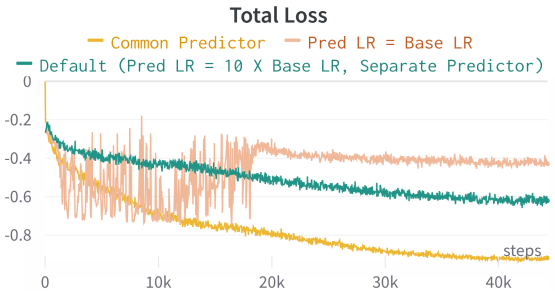


Figure 5: Loss curves of predictor head design exploration.

where the two audio-visual segments are sampled in a temporally random manner. The results presented in Figure 4 show that the *mild* relaxation works best for both action recognition and sound classification. Interestingly, we find that *medium* relaxation shows worse performance in comparison to others, whereas, *extreme* relaxation works somewhat well in our setup.

4.4 Exploring Design Choices

Predictor. Our empirical study shows that the predictor head plays an important role in effectively training the audio and visual encoders to learn good representations. The predictor architecture is similar to (Chen and He 2021). For the sake of completeness, we provide the details of the predictor head in Appendix F. We explore (*i*) different learning rates, and (*ii*) using a common vs. a separate predictor in the multi-modal setup. It should be noted that none of the variants cause a collapse, even though we notice considerable differences in performance. We present the findings below.

Following (Chen and He 2021), we use a constant learning rate for the predictors. However, unlike (Chen and He 2021), where the predictor learning rate is the same as the base learning rate of the encoder, we find that a higher predictor learning rate helps the network to learn better representations. In particular, setting the predictor learning rate to be the same as the base learning rate results in unstable training, and the loss curve shows oscillating behavior. We empirically find that setting the predictor learning rate to 10 times the base learning rate works well. We present the results in Table 3 and training curves in Figure 5.

Next, we evaluate whether the framework can be trained with a common predictor head instead of separate predictor heads (default setup). In simple terms, one predictor head would work towards identity mapping for both audio and visual feature vectors. To test this, l2-normalized feature vectors $f_v(v)$ and $f_a(a)$ are fed to the predictor, which are then used in a usual manner to optimize the cost function. The results are presented in Table 3. We observe that though such a setup works somewhat well, having separate predictors is beneficial for learning better representations. We present the training curves in Figure 5, it shows using common predictor head results in training losses saturate very quickly ultimately yielding worse performance compared to the use of separate predictor heads.

	Pretraining Dataset		
	KS (22K)	**K400 (240K)**	**AS (1.8M)**
HMDB51	45.7	50.0	56.2
UCF101	78.1	83.9	87.7
Kinetics400	39.0	44.5	50.1
ESC50	82.8	86.8	90.5
DCASE	93.0	96.0	97.0

Table 4: We present the top-1 acc. of linear evaluation on action recognition and sound classification.

Projector. We present a comparative study of projection heads with 2 layers vs. 3 layers (default setup). We notice 2.4% and 4% improvements in top-1 accuracies when using 3 layers instead of 2 on action recognition and sound classification respectively (please see Table 3). The architecture details are presented in Appendix F.

4.5 Exploring Audio-Visual Augmentations

We perform an in-depth study to explore the impact of different audio and visual augmentations.

Visual Augmentations. We explore a wide range of visual augmentations. As a starting point, we adopt the basic spatial augmentations used in (Morgado, Vasconcelos, and Misra 2021), which consists of Multi-Scale Crop (MSC), Horizontal Flip (HF), and Color Jitter (CJ). Additionally, we explore other augmentations, namely Gray Scale (GS), Gaussian Blur (GB) (Chen et al. 2020), and Cutout (C) (DeVries and Taylor 2017), which show great performance in image-based self-supervised learning (Chen et al. 2020; Van Gansbeke et al. 2020). We explore almost all the possible combinations of different visual augmentations in a uni-modal setup and present the results in Table 5. The results show that strong augmentations improve the top-1 accuracy by 6.8% in comparison to basic augmentations used in (Morgado, Vasconcelos, and Misra 2021).

Temporal Consistency of Spatial Augmentations. While investigating different spatial augmentations, we are also interested to know if the spatial augmentations should be consistent at the frame level or whether they should be random (i.e., vary among consecutive frames within a sequence). We refer to these concepts as *temporarily consistent* or *temporarily random*. We perform an experiment where we apply MSC−HF−CJ−GS randomly at the frame level and compare the results to applying the same augmentations consistently across all the frames of a sequence. Our results show that maintaining temporal consistency in spatial augmentations across consecutive frames is beneficial, which is in line with

	Visual	UCF101	Audio	ESC50
Uni	MSC−HF−CJ	62.3	VJ	44.8
	MSC−HF−CJ−GS	68.1	VJ−M	49.5
	MSC−HF−CJ−GS−C	68.3	VJ−M−TW	49.5
	MSC−HF−CJ−GS−GB	68.7	VJ−M−RC	**62.0**
	MSC−HF−CJ−GS−GB−C	**69.1**		
	Visual + Audio	**UCF101**		**ESC50**
Multi	MSC−HF−CJ−GS−C + VJ−M−RC	73.9		79.0
	MSC−HF−CJ−GS−GB + VJ−M−RC	73.5		79.0
	MSC−HF−CJ−GS−GB−C + VJ−M−RC	**74.8**		**79.0**

Table 5: Exploring audio-visual augmentations.

the findings in (Qian et al. 2021). Specifically, *Temporally random* augmentations, results in top-1 accuracy of 53.69%, whereas, the same augmentations applied in a *temporally consistent* manner results in 68.09%.

Audio Augmentations. Similar to visual augmentations, we thoroughly investigate a variety of audio augmentations. Our audio augmentations include, Volume Jitter (VJ), Time and Frequency Masking (Mask) (Park et al. 2019), Random Crop (RC) (Niizumi et al. 2021), and Time Warping (TW) (Park et al. 2019). We also explore almost all the possible combinations of these augmentations and present the results in Table 5. Our findings show that time-frequency masking and random crop improve the top-1 accuracy by 17.25% compared to the base variant. We also notice that time warping doesn't improve performance and is also quite computationally expensive. Hence, going forward we do not use time warping during pretraining.

Audio-Visual Augmentations. We conduct further experiments on a few combinations of augmentations in a *multi-modal* setup. We pick the top-performing augmentations obtained from the uni-modal variants and apply them concurrently. The results are presented in Table 5 where we find that the results are consistent with the uni-modal setups, as the combination of MSC−HF−CJ−GS−GB−C and VJ−M−RC performs the best in comparison to the other combinations. Finally, We summarize the augmentation schemes used for pretraining and evaluation in Tables S3 and S4 in the Appendix.

4.6 Linear Evaluation and Scalability

To evaluate the quality of the self-supervised representations, we perform linear evaluations on action recognition (HMDB51, UCF101, and Kinetics400) and sound classification (ESC50 and DCASE). We use 3 different-sized datasets, i.e., Kinetics-Sound, Kinetics400, and AudioSet for pretraining. In Table 4 we report the top-1 accuracies averaged over all the splits. We notice a steady improvement in performance as the pertaining dataset size increases, which shows CrissCross can likely be scaled on even larger datasets like IG65M (Ghadiyaram, Tran, and Mahajan 2019). Please note that in order to evaluate scalability we choose linear evaluation accuracy instead of full-finetuning as it gives more accurate measurements of learned representations obtained through self-supervised pretraining.

Method	Compute	Backbone	U101	H51
Pretrained Dataset: Kinetics-Sound (Finetune input 32×224^2)				
CM-ACC(2020)	40 GPUs	3D-ResNet-18	77.2	40.6
CrissCross	**4 GPUs**	R(2+1)D-18	**88.3**	**60.5**
Supervised (2020)	-	3D-ResNet-18	86.9	53.1
Pretrained Dataset: Kinetics400 (Finetune input 8×224^2)				
XDC (2020)	64 GPUs	R(2+1)D-18	74.2	39.0
AVID (2021)	64 GPUs	R(2+1)D-18	83.7	49.5
Robust-xID (2021)	8 GPUs	R(2+1)D-18	81.9	49.5
CrissCross	**8 GPUs**	R(2+1)D-18	**86.9**	**54.3**
Pretrained Dataset: Kinetics400 (Finetune input 32×224^2)				
SeLaVi (2020)	64 GPUs	R(2+1)D-18	83.1	47.1
XDC (2020)	64 GPUs	R(2+1)D-18	86.8	52.6
CM-ACC* (2020)	40 GPUs	3D-ResNet18	90.2	61.8
AVID (2021)	64 GPUs	R(2+1)D-18	87.5	60.8
GDT (2021a)	64 GPUs	R(2+1)D-18	90.9	62.3
CMAC (2021)	8 GPUs	R(2+1)D-18	90.3	61.1
Robust-xID (2021)	8 GPUs	R(2+1)D-18	85.6	55.0
CrissCross	**8 GPUs**	R(2+1)D-18	**91.5**	**64.7**
Supervised (2021a)	-	R(2+1)D-18	95.0	74.0
Pretrained Dataset: AudioSet (Finetune input 8×224^2)				
XDC (2020)	64 GPUs	R(2+1)D-18	84.9	48.8
AVID (2021)	64 GPUs	R(2+1)D-18	88.6	57.6
CrissCross	**8 GPUs**	R(2+1)D-18	**89.4**	**58.3**
Pretrained Dataset: AudioSet (Finetune input 32×224^2)				
XDC (2020)	64 GPUs	R(2+1)D-18	93.0	63.7
MMV (2020)	32 TPUs	R(2+1)D-18	91.5	70.1
CM-ACC (2020)	40 GPUs	R(2+1)D-18	93.5	67.2
BraVe** (2021)	16 TPUs	R(2+1)D-18	**93.6**	**70.8**
AVID (2021)	64 GPUs	R(2+1)D-18	91.5	64.7
CrissCross	**8 GPUs**	R(2+1)D-18	92.4	67.4
Supervised (2021)	-	R(2+1)D-18	96.8	75.9

$*$ refers to 240K samples from Kinetics700. $**$ pretrained with very high temporal resolutions (2 views of 32 & 128 frames) compared to others ($8/16/32$).

Table 6: SOTA comparison on action recognition.

4.7 Comparison to the State-of-the-Arts

Action Recognition. In line with (Alwassel et al. 2020; Asano et al. 2020; Morgado, Vasconcelos, and Misra 2021; Patrick et al. 2021a; Ma et al. 2020), we benchmark CrissCross using UCF101 and HMDB51 on action recognition. For a fair comparison to earlier works, we adopt 2 setups for finetuning, once with 8 frames, and the other with 32 frames. In both these setups, we use a spatial resolution of 224^2. We tune the model using the split-1 of both datasets and report the top-1 accuracy averaged over all the splits. We notice large variability in experimental setups in the literature in terms of different backbones (e.g., deeper ConvNets, Transformer-based architectures, etc.) (Piergiovanni, Angelova, and Ryoo 2020; Qian et al. 2021; Patrick et al. 2021b), pretraining inputs (e.g., the addition of optical flow or text in addition to audio-visual data, etc.) (Piergiovanni, Angelova, and Ryoo 2020; Qian et al. 2021; Alayrac et al. 2020), and pretraining datasets, making it impractical to

Method	UCF101			HMDB51		
	R@1	R@5	R@20	R@1	R@5	R@20
ST Order (2018)	25.7	36.2	49.2	-	-	-
SpeedNet (2020)	13.0	28.1	49.5	-	-	-
Clip Order (2019)	14.1	30.3	51.1	7.6	22.9	48.8
VCP (2020)	18.6	33.6	53.5	7.6	24.4	53.6
VSP (2020)	24.6	41.9	76.9	10.3	26.6	54.6
CoCLR (2020)	55.9	70.8	82.5	26.1	45.8	69.7
SeLaVi (2020)	52.0	68.6	84.5	24.8	47.6	75.5
Robust-xID (2021)	60.9	79.4	90.8	**30.8**	55.8	79.7
GDT (2021a)	57.4	73.4	88.1	25.4	51.4	75.0
CrissCross	**63.8**	78.7	89.9	26.4	50.5	77.7

Table 7: SOTA comparison on action retrieval.

Method	ESC50		DCASE	
	K400	AS	K400	AS
AVTS (2018)	76.7	80.6	91	93
XDC (2020)	78.0	84.8	91	95
AVID (2021)	79.1	89.1	93	96
MMV (2020)	-	85.6	-	-
BraVe (2021)	-	90.4	-	-
CrissCross	**86.8**	**90.5**	**96**	**97**

Table 8: SOTA comparison on sound classification.

compare to all the prior works. Following the inclusion criteria of earlier works (Patrick et al. 2021a; Alwassel et al. 2020; Morgado, Vasconcelos, and Misra 2021), we compare CrissCross with methods that use similar backbones, inputs, and pretraining datasets.

The comparison of CrissCross with recent works is presented in Table 6. When pretrained with Kinetics400, CrissCross outperforms all the prior works by considerable margins on UCF101 and HMDB51 in both the fine-tuning setups. Moreover, CrissCross outperforms the current state-of-the-art AVID (Morgado, Vasconcelos, and Misra 2021), when pretrained on AudioSet and fine-tuned with 8-frame inputs, on both the UCF101 and HMDB51. When fine-tuned with 32-frame inputs, CrissCross achieves competitive results amongst the leading methods. We note that some of the prior works show slightly better performance compared to ours in some settings. We conjecture this to be due to the use of higher spatio-temporal resolution pretraining inputs in these models. E.g., BraVe (Recasens et al. 2021) is pretrained with 2 views of 32×112^2 and 128×112^2, and the input size for MMV (Alayrac et al. 2020) and CM-ACC (Ma et al. 2020) are 32×224^2 and 16×224^2, respectively. In comparison, CrissCross is pretrained with visual inputs of size 8×112^2. However, we expect the performance of our model to improve further by using such higher resolutions, given the trend shown in (Recasens et al. 2021).

In addition to the commonly used Kinectis400 and AudioSet, we further evaluate CrissCross while pretrained on the small-scale Kinetics-Sound. Here, we observe significant improvements compared to the current state-of-the-art CM-ACC (Ma et al. 2020) on both UCF101 (88.3 vs. 77.2) and HMDB51 (60.5 vs. 40.6). Additionally, CrissCross outperforms fully-supervised pretraining by 1.4% and 7.4% on UCF101 and HMDB51 respectively when both the fully-supervised and self-supervised methods are pretrained on Kinetics-Sound. To the best of our knowledge, this is the first time that self-supervision outperforms full-supervised pretraining on action recognition using the same small-scale pretraining dataset, showing that our method performs well on limited pretraining data.

Action Retrieval. In addition to full finetuning, we also compare the performance of CrissCross in an unsupervised setup. Following prior works (Morgado, Misra, and Vasconcelos 2021; Patrick et al. 2021a; Asano et al. 2020), we perform action retrieval using the split-1 of both UCF101 and HMDB51. The results are presented in Table 7 shows that CrissCross outperforms the current state-of-the-arts on UCF101 while achieving competitive results for HMDB51.

Sound Classification. We use two popular benchmarks ESC50 and DCASE to perform sound classification. We find large variability of experimental setups in the literature for evaluating audio representations. For instance, different backbones, input lengths, datasets, and evaluation protocols (linear evaluation, full-finetuning) have been used, making it impractical to compare to all the prior works. Following (Recasens et al. 2021; Alayrac et al. 2020), we perform linear evaluations using 5-second inputs on ESC50 and 1-second input for DCASE. As presented in Table 8, CrissCross outperforms current state-of-the-art AVID (Morgado, Vasconcelos, and Misra 2021) and BraVe (Recasens et al. 2021) on ESC50, while pretrained on Kinetics400 and AudioSet respectively. Additionally, CrissCross sets new state-of-the-art by outperforming all the prior works on DCASE when pretrained on both Kinetics400 and AudioSet.

5 Summary

We propose a novel self-supervised framework to learn audio-visual representations by exploiting intra-modal, as well as, synchronous and *asynchronous* cross-modal relationships. We conduct a thorough study investigating the major concepts of our framework. Our findings show that relaxation of cross-modal temporal synchronicity is beneficial for learning effective audio-visual representations. These representations can then be used for a variety of downstream tasks including action recognition, sound classification, and action retrieval.

Acknowledgments

We are grateful to the Bank of Montreal and Mitacs for funding this research. We are thankful to SciNet HPC Consortium for helping with the computation resources.

References

Ahsan, U.; Madhok, R.; and Essa, I. 2019. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *WACV*, 179–189.

Alayrac, J.-B.; Recasens, A.; Schneider, R.; Arandjelovic, R.; Ramapuram, J.; De Fauw, J.; Smaira, L.; Dieleman, S.; and Zisserman, A. 2020. Self-Supervised MultiModal Versatile Networks. *NeurIPS*, 2(6): 7.

Alwassel, H.; Mahajan, D.; Korbar, B.; Torresani, L.; Ghanem, B.; and Tran, D. 2020. Self-Supervised Learning by Cross-Modal Audio-Video Clustering. *NeruIPS*, 33.

Arandjelovic, R.; and Zisserman, A. 2017. Look, listen and learn. In *ICCV*, 609–617.

Asano, Y. M.; Patrick, M.; Rupprecht, C.; and Vedaldi, A. 2020. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*.

Babaeizadeh, M.; Finn, C.; Erhan, D.; Campbell, R. H.; and Levine, S. 2018. Stochastic Variational Video Prediction. In *ICLR*.

Bardes, A.; Ponce, J.; and LeCun, Y. 2021. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*.

Benaim, S.; Ephrat, A.; Lang, O.; Mosseri, I.; Freeman, W. T.; Rubinstein, M.; Irani, M.; and Dekel, T. 2020. Speednet: Learning the speediness in videos. In *CVPR*, 9922–9931.

Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; and Shah, R. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Buchler, U.; Brattoli, B.; and Ommer, B. 2018. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *ECCV*, 132–149.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*, 15750–15758.

Cho, H.; Kim, T.; Chang, H. J.; and Hwang, W. 2020. Self-Supervised Spatio-Temporal Representation Learning Using Variable Playback Speed Prediction. *arXiv preprint arXiv:2003.02692*.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Finn, C.; Goodfellow, I.; and Levine, S. 2016. Unsupervised learning for physical interaction through video prediction. *NeurIPS*, 29: 64–72.

Gemmeke, J. F.; Ellis, D. P.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R. C.; Plakal, M.; and Ritter, M. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 776–780.

Ghadiyaram, D.; Tran, D.; and Mahajan, D. 2019. Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition. In *CVPR*, 12038–12047.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Pires, B.; Guo, Z.; Azar, M.; et al. 2020. Bootstrap Your Own Latent: A new approach to self-supervised learning. In *NeurIPS*.

Han, T.; Xie, W.; and Zisserman, A. 2020. Self-supervised Co-training for Video Representation Learning. In *NeurIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Jing, L.; Yang, X.; Liu, J.; and Tian, Y. 2018. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Khare, A.; Parthasarathy, S.; and Sundaram, S. 2021. Self-Supervised learning with cross-modal transformers for emotion recognition. In *SLT*, 381–388.

Kim, D.; Cho, D.; and Kweon, I. S. 2019. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, volume 33, 8545–8552.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Korbar, B.; Tran, D.; and Torresani, L. 2018. Cooperative learning of audio and video models from self-supervised synchronization. In *NeruIPS*, 7774–7785.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *ICCV*, 2556–2563.

Lee, H.-Y.; Huang, J.-B.; Singh, M.; and Yang, M.-H. 2017. Unsupervised representation learning by sorting sequences. In *CVPR*.

Liang, X.; Lee, L.; Dai, W.; and Xing, E. P. 2017. Dual motion gan for future-flow embedded video prediction. In *ICCV*, 1744–1752.

Loshchilov, I.; and Hutter, F. 2017. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*.

Luo, D.; Liu, C.; Zhou, Y.; Yang, D.; Ma, C.; Ye, Q.; and Wang, W. 2020. Video cloze procedure for self-supervised spatio-temporal learning. In *AAAI*.

Ma, S.; Zeng, Z.; McDuff, D.; and Song, Y. 2020. Active Contrastive Learning of Audio-Visual Video Representations. In *ICLR*.

Mathieu, M.; Couprie, C.; and LeCun, Y. 2016. Deep multi-scale video prediction beyond mean square error. In *ICLR*.

Min, S.; Dai, Q.; Xie, H.; Gan, C.; Zhang, Y.; and Wang, J. 2021. Cross-Modal Attention Consistency for Video-Audio Unsupervised Learning. *arXiv preprint arXiv:2106.06939*.

Misra, I.; and Maaten, L. v. d. 2020. Self-supervised learning of pretext-invariant representations. In *CVPR*, 6707–6717.

Misra, I.; Zitnick, C. L.; and Hebert, M. 2016. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 527–544.

Morgado, P.; Misra, I.; and Vasconcelos, N. 2021. Robust Audio-Visual Instance Discrimination. In *CVPR*, 12934–12945.

Morgado, P.; Vasconcelos, N.; and Misra, I. 2021. Audio-visual instance discrimination with cross-modal agreement. In *CVPR*, 12475–12486.

Niizumi, D.; Takeuchi, D.; Ohishi, Y.; Harada, N.; and Kashino, K. 2021. BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation. *arXiv preprint arXiv:2103.06695*.

Omeiza, D.; Speakman, S.; Cintas, C.; and Weldermariam, K. 2019. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

Patrick, M.; Asano, Y. M.; Kuznetsova, P.; Fong, R.; Henriques, J. F.; Zweig, G.; and Vedaldi, A. 2021a. On compositions of transformations in contrastive self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9577–9587.

Patrick, M.; Huang, P.-Y.; Misra, I.; Metze, F.; Vedaldi, A.; Asano, Y. M.; and Henriques, J. F. 2021b. Space-Time Crop & Attend: Improving Cross-modal Video Representation Learning. In *ICCV*, 10560–10572.

Piczak, K. J. 2015. ESC: Dataset for Environmental Sound Classification. In *ACM Conference on Multimedia*, 1015–1018. .

Piergiovanni, A.; Angelova, A.; and Ryoo, M. S. 2020. Evolving losses for unsupervised video representation learning. In *CVPR*, 133–142.

Qian, R.; Meng, T.; Gong, B.; Yang, M.-H.; Wang, H.; Belongie, S.; and Cui, Y. 2021. Spatiotemporal contrastive video representation learning. In *CVPR*, 6964–6974.

Recasens, A.; Luc, P.; Alayrac, J.-B.; Wang, L.; Strub, F.; Tallec, C.; Malinowski, M.; Patraucean, V.; Altché, F.; Valko, M.; et al. 2021. Broaden Your Views for Self-Supervised Video Learning. *arXiv preprint arXiv:2103.16559*.

Reda, F. A.; Liu, G.; Shih, K. J.; Kirby, R.; Barker, J.; Tarjan, D.; Tao, A.; and Catanzaro, B. 2018. Sdc-net: Video prediction using spatially-displaced convolution. In *ECCV*, 718–733.

Saito, M.; Matsumoto, E.; and Saito, S. 2017. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2830–2839.

Sarkar, P.; and Etemad, A. 2020a. Self-supervised ECG representation learning for emotion recognition. *IEEE Transactions on Affective Computing*.

Sarkar, P.; and Etemad, A. 2020b. Self-supervised learning for ecg-based emotion recognition. In *ICASSP*, 3217–3221.

Sarkar, P.; Lobmaier, S.; Fabre, B.; Berg, G.; Mueller, A.; Frasch, M. G.; Antonelli, M. C.; and Etemad, A. 2020. Detection of Maternal and Fetal Stress from ECG with Self-supervised Representation Learning. *arXiv e-prints*, arXiv–2011.

Siriwardhana, S.; Kaluarachchi, T.; Billinghurst, M.; and Nanayakkara, S. 2020. Multimodal Emotion Recognition With Transformer-Based Self Supervised Feature Fusion. *IEEE Access*, 8: 176274–176285.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Stowell, D.; Giannoulis, D.; Benetos, E.; Lagrange, M.; and Plumbley, M. D. 2015. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10): 1733–1746.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2016. Deep end2end voxel2voxel prediction. In *CVPRW*, 17–24.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 6450–6459.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 1526–1535.

Van Gansbeke, W.; Vandenhende, S.; Georgoulis, S.; Proesmans, M.; and Van Gool, L. 2020. Scan: Learning to classify images without labels. In *ECCV*, 268–285.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. *NeurIPS*, 29: 613–621.

Wang, J.; Jiao, J.; Bao, L.; He, S.; Liu, W.; and Liu, Y.-H. 2021. Self-supervised Video Representation Learning by Uncovering Spatio-temporal Statistics. *PAMI*.

Xu, D.; Xiao, J.; Zhao, Z.; Shao, J.; Xie, D.; and Zhuang, Y. 2019. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 10334–10343.