SPEEDAUG: A SIMPLE CO-AUGMENTATION METHOD FOR UNSUPERVISED AUDIO-VISUAL PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a speed co-augmentation method for unsupervised audio-visual pre-training. A playback speed is randomly selected and applied to both audio and video data to diversify audio-visual views. By applying this augmentation, we observe an interesting phenomenon that multi-modal co-augmentation leads to data entanglement and even semantic meaning shift (*e.g.*, a sped-up sound from a *cat* can be mistaken as the sound from a *mouse*). This differs from the common intuition in single-modality representation learning, where samples are invariant to different augmentations. To combat this, augmented audio-visual views are formulated as a partial relationship via our proposed SoftInfoNCE loss during unsupervised pre-training. The learned representations are evaluated on three downstream tasks, including action recognition and video retrieval on the UCF101 and HMDB51 datasets, and video-audio retrieval on the Kinetics-Sounds dataset. Extensive experimental results show that we achieve a new state-of-the-art.

1 INTRODUCTION

Learning from unlabeled data enables deep neural networks to produce general representations and frees people from tedious annotation work. Contrastive learning has remarkably improved unsupervised representation learning in recent years (He et al., 2020; Chen et al., 2020b). The essential idea is to minimize the distances of positive pairs while maximizing those of negative pairs. Usually, one sample with different augmentations is considered positive while other samples in the same batch are considered negative. Under this framework, extensive investigations were conducted to develop approaches for learning representations from different modalities, including images (He et al., 2020; Chen et al., 2020b; Caron et al., 2021), videos (Feichtenhofer et al., 2021; Qian et al., 2021; Pan et al., 2021), and audio (Wang et al., 2022; Wang & Oord, 2021), to name a few. By training neural networks to be invariant to strong augmentations, representations learned from contrastive learning demonstrate great generalization ability and even exceed its supervised learning counterparts (He et al., 2020; Chen et al., 2021).

Multi-modal contrastive learning (Radford et al., 2021; Morgado et al., 2021b; Patrick et al., 2021a) (*e.g.*, audio-visual) has received growing attention within the community due to the observation that video content is usually accompanied by audio signals. Under this framework, quite a few existing approaches (Morgado et al., 2021a; Patrick et al., 2021a) focus on achieving better discrimination of positive and negative pairs to improve audio-visual representation learning. While promising results have been achieved, these works (Ma et al., 2020; Morgado et al., 2021a) usually apply data augmentations to each modality individually, which may limit the diversity of the generated data views and restrict the potential of augmentation for contrastive learning.

Inspired by the success of augmentations in single-modality learning, we propose SpeedAug, a simple co-augmentation method that changes the playback speed of both audio and visual data to synthesize more views. This method is easy to implement as it only changes the sampling rate of the raw data. Fig. 1 shows examples of audio and video with the speed augmentation applied. We observe that speed augmentation does not change the appearance characteristics of the video modality much. But spectrograms of the sped-up audio data demonstrate notable discrepancy that the amplitude of high-frequency signals is increased and the amplitude of low-frequency signals is decreased. In this case, the audio and video pair from the same clip are no longer clearly positively related as considered in the common practice. We hypothesize that there is a partial relationship

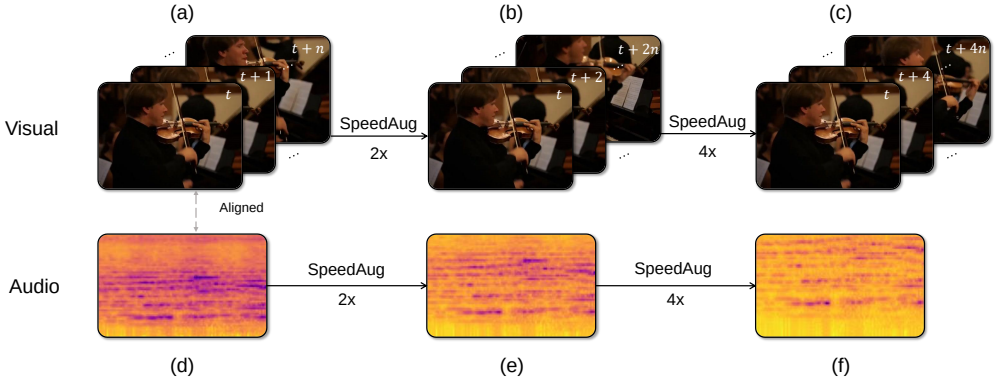

Figure 1: **Illustration of the SpeedAug transformation. Top:** applying SpeedAug on a video clip in **(a)**; and the corresponding transformed video clips after two and four times of speed-up in **(b, c)**. **Bottom:** applying SpeedAug on an audio sample in **(d)**, shown in spectrogram, where the horizontal axis represents time and vertical axis represents frequency; and the corresponding transformed spectrograms after two and four times of speed-up in **(e, f)**. The speed co-augmentation leads to discrepancy between the augmented audio-visual pairs.

between the augmented audio-visual pairs, where this relationship is affected by the intensity of the speed augmentation. We model such a relationship with a cross-affinity module that automatically learns the audio-visual correlations across views. The learned correlations quantitatively measure the audio-visual consistency, which is then utilized for the contrastive loss computations.

Combining the proposed speed augmentation and the cross-affinity module, we present a Speed-augmented visual-audio Contrastive Learning framework, which we call *SvaCLR*, for unsupervised representation learning. We validate our proposed framework on three downstream tasks, including action recognition, video retrieval, and audio-video retrieval. We show that, despite its simplicity, our approach significantly outperforms current state-of-the-art (Ma et al., 2020) by 14.3% on HMDB51 (Kuehne et al., 2011) and 9.2% on UCF101 (Soomro et al., 2012), when using Kinetics-Sounds (Arandjelovic & Zisserman, 2017) as the pre-training dataset. Our approach also demonstrates its scalability to larger pre-training datasets, where when using a large-scale VGGSound (Chen et al., 2020a) for pre-training, our approach surpasses the state-of-the-art (Patrick et al., 2021a) by 5.5% on HMDB51 and 1.7% on UCF101.

2 Related Works

Unsupervised visual representation learning. Starting from static images, early unsupervised visual representation learning methods mainly focus on pretext task design, aiming at designing tasks that can train the deep model with supervision extracted from the data itself, without using human-annotated labels. Such pretext tasks are explored from different perspectives, including spatial relationships (Doersch et al., 2015; Noroozi & Favaro, 2016), low-level reconstruction (Pathak et al., 2016), colorization (Zhang et al., 2016), and rotation classification (Gidaris et al., 2018), to name a few. Later on, some of these image-based pretext tasks are extended to dynamic videos, *e.g.*, spatio-temporal relationships (Kim et al., 2019; Luo et al., 2020) and video colorization (Vondrick et al., 2018). There are also some pretext tasks specifically designed for video modality, *e.g.*, frame ordering (Misra et al., 2016; Xu et al., 2019), spatio-temporal statistics prediction (Wang et al., 2019a), and future prediction (Han et al., 2019; 2020b) and tracking (Wang et al., 2019b). More recently, some works (Benaim et al., 2020; Wang et al., 2020) study the distinct underlying property of video – playback speed – and learn video representations by predicting the speed. Though simple, such speed modeling pretext tasks are shown to be effective in learning good video representations. In this work, we also take speed into consideration for representation learning by applying speed augmentation to an additional audio data modality.

Unsupervised multi-modal representation learning. Recent works have investigated multi-modal data for contrastive learning, *e.g.*, vision with text, and vision with audio. DeVise (Frome et al., 2013) proposed the idea of joint image-text embedding, in which the semantic similarity is measured in the feature space via dot product. With the availability of large-scale datasets, joint embeddings have been widely explored (Xu et al., 2015; Klein et al., 2015; Pan et al., 2016; Miech et al., 2020), especially for the recently proposed instructional video datasets (Alayrac et al., 2016; Sener et al., 2015; Miech et al., 2019). Though promising, the text and language data within the above sources still requires manual annotation, even with Automatic Speech Recognition for text generation, as the ASR models need to be trained with annotation. On the other hand, audio, as an accompanied signal, often comes with video data and does not require any kind of external manual annotation. As a result, many works have proposed to learn unsupervised joint video-audio representations (Arandjelovic & Zisserman, 2017; 2018; Korbar et al., 2018; Owens & Efros, 2018), where the key idea is to model the audio-visual correspondence, *i.e.*, whether the audio sample and video sample are temporally aligned. We also learn based on unsupervised audio-visual contrastive learning, but take a step further by introducing multi-modal speed transformation. There are also some recent works (Alayrac et al., 2020; Aytar et al., 2017) that combine vision, text, and audio together to learn joint embeddings, which is beyond the scope of this paper. Instead, we focus on learning a better joint representations for video and audio, by considering *speed* — a by-nature built-in property.

Data augmentation. Augmentation is a widely used technique to enlarge the diversity of the training data and prevent over-fitting (Baird, 1992; Simard et al., 2003; Krizhevsky et al., 2012). In supervised learning, common augmentations for images consist of color jittering, cropping, flipping, and rotation, to name but a few. For automatic speech recognition, augmentations are applied either on the raw data (Ko et al., 2015) or on the spectrogram (Park et al., 2019). Some recent works also propose to automatically search for the best augmentation policies (Cubuk et al., 2019; 2020). Existing methods usually assume data is invariant to these augmentations and the intuition is to teach the model of such invariances. While in our work, we show that the visual-audio pair in the cross-modality scenario is not invariant to speed co-augmentation and should not be considered as strict positive pairs. Instead, we hypothesize that there is a partial relationship between the augmented visual-audio views and model it with a cross-affinity module to further improve the pre-training.

3 Proposed Method

We introduce the SvaCLR framework for unsupervised audio-visual pre-training. Our SvaCLR is built upon the contrastive learning framework (Chen et al., 2020b) and models data correlations across audio and video modalities after speed augmentations. We first present an overview of the proposed method in Section 3.1. Then we describe the speed augmentation with vanilla InfoNCE loss in Section 3.2 and the proposed cross-affinity module with SoftInfoNCE loss in Section 3.3. Finally, the network architectures and training details are shown in Section 3.4.

3.1 Overview

Our target is to train video and audio encoders via unsupervised contrastive learning. Fig. 2 shows an overview of the proposed SvaCLR framework. Given an aligned visual-audio pair (v, a), we apply speed augmentations on both v and a to synthesize two additional views $\widetilde{v}$ and $\widetilde{a}$. These audio and video samples are then fed into the audio and video encoders $f(\cdot)$ and $g(\cdot)$ to extract representations y. We project the video and audio representations separately via projectors $h_v(\cdot)$ and $h_a(\cdot)$. The projected embeddings z are then utilized to compute the contrastive InfoNCE loss (Oord et al., 2018). In parallel, we introduce a cross-affinity module to model the audio-visual correlations. The modeled correlations are used to re-weight the InfoNCE loss, resulting in our proposed SoftInfoNCE loss. In the following, we first introduce the speed augmentation along with the vanilla InfoNCE loss, and then introduce the cross-affinity module for SoftInfoNCE loss computation.

3.2 Speed augmentation with InfoNCE

For speed augmentations, we use a speed library to diversify training data pairs. We use $\mathcal{T}$ to represent the speed augmentation set in which the maximum speed is denoted by S. Each time, two speed amounts for the audio and video data are selected stochastically from $\mathcal{T}$ and are applied to each data.

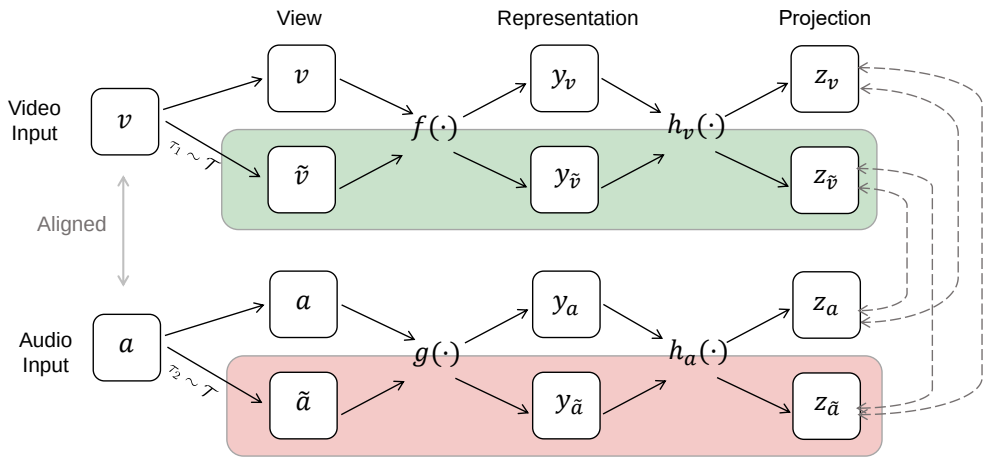

Figure 2: **Overview of the proposed SvaCLR framework.** We apply speed augmentations to both audio and video samples by changing their playback speed (only one audio-video pair is shown for simplicity). We use target backbone encoders $f(\cdot)$ and $g(\cdot)$ to extract their embeddings. These embeddings are sent to the projectors $h_v(\cdot)$ and $h_a(\cdot)$ for contrastive learning. In parallel, we introduce a cross-affinity module (see Fig. 3 for more details) to model the audio-video embedding correlations. The modeled correlations reweigh the InfoNCE loss when learning audio-video representations. Highlighted regions in green and red are the augmented view paths of the video and audio inputs, respectively. The double-arrowed connections at the end indicate correlation modelling.

In practice, the proposed speed augmentation is implemented by applying different sampling rates to the audio and video samples (for details, see Appendix A.2). Our speed augmentation differs from off-the-shelf operations (*e.g.*, Gaussian blur, cropping, and color jittering) which assume data representations are invariant to augmentations. As shown in Fig 1, partial relationship exists between the co-augmented audio-video pairs variant according to $\mathcal{T}$. The variance is further modeled by cross-affinity module and reweighs the vanilla contrastive InfoNCE loss.

Before computing the contrastive InfoNCE loss (Oord et al., 2018), we project the video and audio representations separately via projectors. As shown in Fig. 2, we apply a video projector $h_v(\cdot)$ upon the video encoder and an audio projector $h_a(\cdot)$ upon the audio encoder. The projected representations are then utilized to compute the contrastive InfoNCE loss as follows:

$$L(i,j) = \frac{\exp(z_i \cdot z_j \,/\, \eta)}{\exp(z_i \cdot z_j \,/\, \eta) + \sum\limits_{\substack{j=1 \\ j \neq i}}^{N} \exp(z_i \cdot z_j \,/\, \eta)}, \tag{1}$$

where $z_i = h_a(y_i)$ is the audio projection, $z_j = h_v(y_j)$ is the video projection, and η is a constant temperature value. The dot product $\cdot$ measures the similarity between the projected audio and video representations. For the input audio a_i, the summation term is computed by utilizing all the video clips v_j, as long as a_i and v_j are from different samples (*i.e.*, they are unpaired).

3.3 Cross-affinity module

As discussed above, the proposed speed augmentation diversifies audio-visual pairs by changing the playback speed of both modalities and leads to a partial relationship between the augmented audio-visual pairs. It is not appropriate to directly label the co-augmented views as either positive or negative. In order to model the relationship between the augmented audio-visual pairs, we propose a cross-affinity module to measure the correlations between the video and audio representations. Fig. 3 illustrates the proposed module. Given the audio embedding y_i and the video embedding y_j, the

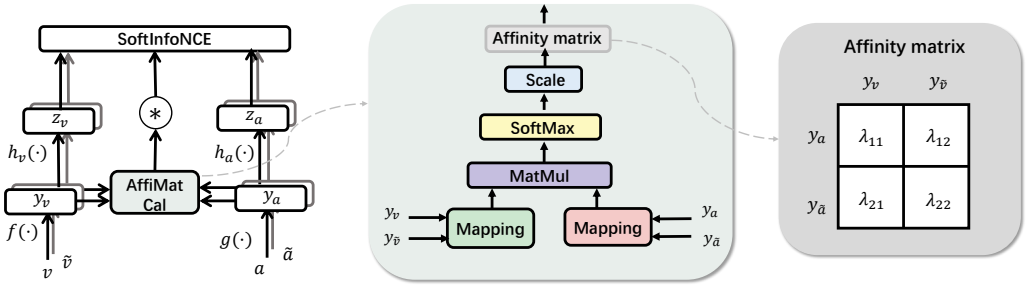

Figure 3: **The proposed cross-affinity module for SoftInfoNCE loss computation.** The cross-affinity module shown on the left takes video and audio representations as input where there are co-augmented audio-visual data. The output is a cross-modality affinity matrix shown on the right. Each element in this matrix represents the correlations between the corresponding audio-visual pairs.

cross-modality affinity $\lambda(a_i^{\tau_1}, v_j^{\tau_2})$ can be computed as follows:

$$\lambda(a_i^{\tau_1}, v_j^{\tau_2}) = \text{softmax}\left[l(y_i) \times l(y_j^{\text{T}})\right], \tag{2}$$

where $l(\cdot)$ is a mapping with learnable parameters. The projected video and audio representations are correlated via the matrix multiplication operation. In practice, three different mapping functions are examined, including identity mapping, linear mapping and nonlinear mapping, from those we find the identity mapping achieves the best results. We speculate this is because heavier mapping could deteriorate the ability of encoders to learn general representations.

We compute the cross-modality affinity in Eq. 2 for co-augmented audio-visual views. The cross-modality affinity can be formulated as a two-by-two matrix (as shown in Fig. 3 right). Each element in this matrix represents the correlation between the audio and video views. The weight between un-augmented audio-visual view is normalized to 1. By using these elements, we reweight the contributions of each co-augmented audio-visual view when computing the contrastive loss.

3.4 Training with SoftInfoNCE

Following (Patrick et al., 2021a; Ma et al., 2020), we first transform the raw audio data to spectrogram as shown in Fig 1 (bottom), and then use a 9-layered 2D ResNet (He et al., 2016) as the audio encoder and R(2+1)D-18 (Tran et al., 2018) as the video encoder (for details, see Appendix A.1). The projector is a two-layered multilayer perceptron (MLP). The training process is end-to-end, without using a two-stage setting as in previous works (Morgado et al., 2021a;b).

Given a batch of audio-visual pairs $\mathcal{A}$ and $\mathcal{V}$, where both $\mathcal{A}$ and $\mathcal{V}$ contain N samples, we denote the speed augmentation set as $\mathcal{T}$, from which we can sample augmentations $\tau \sim p(\mathcal{T})$. The encoders and projectors are trained with the following SoftInfoNCE loss:

$$\mathcal{L}(f, g, \mathcal{A}, \mathcal{V}) = \mathbb{E}_{(\tau_1, \tau_2)\sim p(\mathcal{T})}\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda(a_i^{\tau_1}, v_j^{\tau_2}) \cdot L(g(a_i^{\tau_1}), f(v_j^{\tau_2}))\right], \tag{3}$$

where $L(\cdot,\cdot)$ is the contrastive InfoNCE loss as defined in Eq. 1, $a_i \in \mathcal{A}$, and $v_j \in \mathcal{V}$. The cross-modality affinity $\lambda(\cdot,\cdot)$ takes the audio and video signal as input and measures their correlations. The output correlation value further reweights the contrastive loss during the training process consequently.

4 Experiments

In this section, we first introduce the datasets we used for pre-training, the implementation details, and the downstream evaluations. We then conduct ablation studies to investigate the proposed approach. Following that, the proposed method is compared with state-of-the-art methods.

Table 1: **Audio and video with speed augmentations from the same distribution.** The learned representations are evaluated on action recognition using HMDB51 (Kuehne et al., 2011) and audio-video retrieval on K-Sounds (Arandjelovic & Zisserman, 2017). Speed $s = [a, b]$ represents that the lower bound speed is a while the upper bound is b.

Pre-training Experimental Setup				Downstream Acc.	
speed	Re-weight	Speed	Loss	HMDB51	K-sounds@1
✗	✗	-	InfoNCE	54.2	2.6
✓	✗	$s = [1, 2]$	InfoNCE	63.8 (+9.6)	3.1 (+0.5)
✓	✗	$s = [1, 4]$	InfoNCE	65.1 (+10.9)	3.5 (+0.9)
✓	✗	$s = [1, 6]$	InfoNCE	64.2 (+10.0)	3.2 (+0.6)
✓	✓	$\boldsymbol{s = [1, 4]}$	**SoftInfoNCE**	**66.1 (+11.9)**	**4.6 (+2.0)**

4.1 Implementation details

Dataset. We use Kinetics-Sounds (K-Sounds) (Arandjelovic & Zisserman, 2017), Kinetics-400 (K400) (Kay et al., 2017), and VGGSound (Chen et al., 2020a) as the pre-training datasets to evaluate the effectiveness of our proposed approach. The K-Sounds dataset consists of 20k training samples, the K400 dataset consists of 240k training samples, and the VGGSound dataset contains 199k training samples. We evaluate the learned representations on action recognition and video retrieval tasks with the UCF101 (Soomro et al., 2012) and HMDB51 (Kuehne et al., 2011) datasets.

Unsupervised pre-training. During pre-training, we randomly select 30-frame video clips with sampling rates selected from the speed transformation. Each video clip is cropped to 112×112. For the audio preprocessing, we select audio data with the same initial time as video data and a spectrogram is computed with window length of 0.01 seconds and a half-window overlap. For optimization, we use Stochastic Gradient Descent (SGD) as the optimizer. We use 10 epochs to warm up the learning rate from 64×10^{-3} to 64×10^{-2} and then using a cosine learning rate decay to 10×10^{-2} in the remaining 90 epochs. Training is done on 64 V100 GPUs with a mini-bath size of 8 on each, resulting in total batch size of 512. The total training time is around 30 hours for 100 epochs.

Finetuning. We follow the fine-tuning setting of GDT (Patrick et al., 2021a). SGD is used as the optimizer and the initial learning rate is set to 2.5×10^{-3}, where it is warmed up to 2×10^{-2} in the first two epochs, and decreased by 5×10^{-2} at 6 and 10 epochs. Training is stopped at 12 epochs.

Evaluations. For the action recognition task, we follow the same testing procedure as in previous works (Xu et al., 2019; Luo et al., 2020; Han et al., 2019), and use video accuracy to evaluate our approach. The video accuracy is computed by averaging the softmax probabilities of uniformly selected clips in each video (Xu et al., 2019) from the testing set. For video retrieval tasks, we follow the standard protocol as described in Xu et al. (2019). We also use audio-to-video and video-to-audio retrieval tasks on K-Sounds dataset to evaluate the learned audio and video representations.

4.2 Ablation studies

In our ablation study, we first explore the optimal setting for the speed augmentation on both audio and video data. Then, we study the effectiveness of the proposed cross-affinity module for modeling the partial relationship between the augmented video and audio. Finally, we show the generality of the proposed approach by evaluating it on the audio-visual dataset VGGSound (Chen et al., 2020a).

Speed augmentation. We investigate the best speed augmentation policy using the action recognition task on HMDB51 dataset and the video-to-audio retrieval task on K-Sounds dataset.

We investigate different augmentations for audio and video data from the same speed distribution in Table 1. It can be seen that: **(1)** compared to the no speed-up augmentation, using only two speed candidates can already significantly improve the action recognition performance from 54.2% to 63.8%. This validates the effectiveness of the proposed speed-up augmentation. **(2)** When audio-video pairs are sped up from the same distribution, best action recognition accuracy is achieved when $s = [1, 4]$.

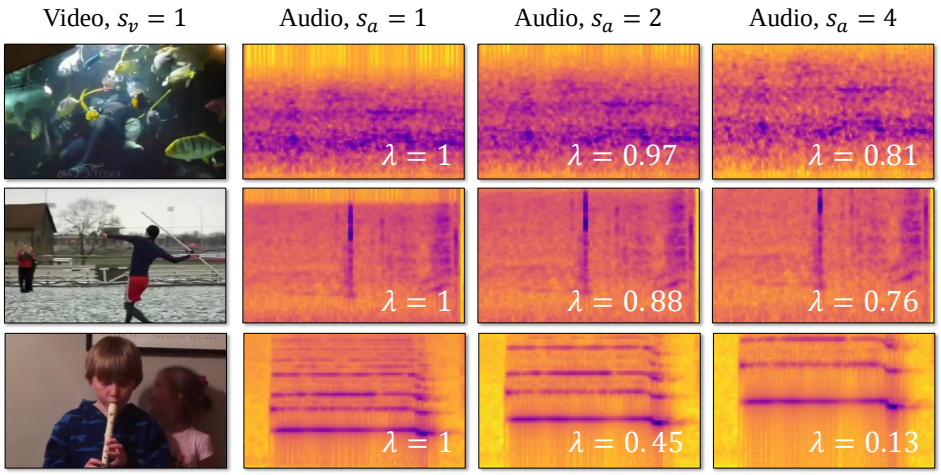

Figure 4: **Three examples of the augmented visual-audio pairs and their corresponding partial relationship weights.** For each sample from left to right: one frame from the video clip, spectrogram of the corresponding audio, spectrogram of the augmented audio with speed 2, and spectrogram of the augmented audio with speed 4. λ represents the weight learned by the cross-affinity module.

Table 2: **Audio and video with speed augmentations from different distributions.** 0.5 represents audio with slow-down augmentation.

V_speed	A_speed	HMDB51	K-Sounds@1
$s_v = [1, 4]$	$s_a = [1, 2]$	63.9	2.5
$s_v = [1, 4]$	$s_a = [0.5, 4]$	64.5	3.1
$s_v = [1, 4]$	$s_a = [1, 4]$	**66.1**	**4.6**

Table 3: **Explore different pre-training datasets**.

Dataset(duration)	HMDB51	K-Sounds@1
-	24.1	54.5
K-Sounds (2d)	54.9	5.1
K400 (28d)	66.1	4.6
VGGSound (23d)	**67.2**	**4.8**

Both $s = [1, 2]$ and $s = [1, 6]$ settings perform inferior than the $s = [1, 4]$ setting. We suspect that this is because compared to $s = [1, 2]$, using $s = [1, 4]$ provides more view synthesis while $s = [1, 6]$ could be a bit difficult for the network to learn useful semantic representations.

We also explore audio and video speed augmentations from different distributions as shown in Table 2. It can be seen that compared to audio speed from $s_a = [1, 2]$ and audio slow-down, audio and video speed from the same distribution $s_v = s_a = [1, 4]$ achieves the best performance. We hypothesize that this is because compared to $s_a = [1, 2]$, $s_a = [1, 4]$ provides more views. Slow-down augmentation may lead to under-provided information to the network while at the same time it will decrease the possibility of the speed-up augmentations.

Cross-affinity module. Based on the best speed augmentation setting $s_a = s_v = [1, 4]$, we investigate the effectiveness of the proposed cross-affinity module in Table 1. It can be seen that with the proposed cross-affinity module and the SoftInfoNCE loss design, the performances can be further improved significantly (*e.g.*, +11.9% for HMDB51). To better understand what the cross-affinity module learns, we visualize the visual-audio pairs with the speed augmentation and their corresponding relationship weights in Fig. 4. It can be seen that videos with manifested audio signals are more sensitive to the proposed speed augmentation, where the weight between the video and augmented audio is relatively small after the speed transformation.

Different datasets pre-training. We validate the generality of the proposed approach on three datasets K-Sounds, K400, and VGGSound in Table 3. It can be observed: **(1)** compared with training from scratch, pre-train with our proposed approach significantly improves the performances; **(2)** compared with K400, pre-training on VGGSound achieves better performance but with less data. We assume it is because the VGGSound dataset is designed and curated to be well-aligned for audio-visual pairs so that it makes the audio-visual contrastive learning a relatively easier task. This shows that quality of pre-training data outweighs the quantity.

Table 4: **Comparison with state-of-the-art approaches on action recognition task.** Approaches are evaluated on UCF101 (Soomro et al., 2012) and HMDB51 (Kuehne et al., 2011) datasets.

Method	Pre-training Experimental Setup			Downstream Acc.	
	Architecture	Dataset (duration)	Resolution	HMDB51	UCF101
Supervised (Xie et al., 2018)	S3D-G	K400+IN (28d)	-	75.9	96.8
Supervised (Alwassel et al., 2019)	R(2+1)D-18	K400 (28d)	-	65.1	94.2
ObjectPatch (Wang & Gupta, 2015)	AlexNet	UCF101 (2d)	227×227	42.7	15.6
ClipOrder (Misra et al., 2016)	CaffeNet	UCF101 (2d)	227×227	50.9	19.8
VCOP (Xu et al., 2019)	R(2+D)-18	UCF101 (2d)	112×112	30.9	72.4
ACC (Ma et al., 2020)	R3D-18	K-Sounds (2d)	224×224	40.6	77.2
SvaCLR (Ours)	R3D-18	K-Sounds (2d)	128×128	**54.9**	**86.4**
L^3-Net (Arandjelovic & Zisserman, 2017)	VGG-16	K400 (28d)	-	40.2	72.3
PEMT (Lee et al., 2021)	SlowFast	K400 (28d)	128×128	-	85.2
GDT (Patrick et al., 2021a)	R(2+1)D-18	K400 (28d)	128×128	62.3	90.9
SvaCLR (Ours)	R(2+1)D-18	K400 (28d)	128×128	66.1	91.5
SvaCLR (Ours)	R(2+1)D-18	VGGSound (23d)	128×128	**67.2**	**92.0**
Multisensory (Owens & Efros, 2018)	R3D-18	K400 (28d)	224×224	-	82.1
SeLaVi (Asano et al., 2020)	R(2+1)D-18	K400 (28d)	224×224	47.1	84.2
SpeedNet (Benaim et al., 2020)	S3D-G	K400 (28d)	224×224	48.8	81.1
XDC (Alwassel et al., 2019)	R(2+1)D-18	K400 (28d)	224×224	52.6	86.2
AVTS (Korbar et al., 2018)	MC3-18	K400 (28d)	224×224	56.9	85.8
STiCA (Patrick et al., 2021a)	R(2+1)D-18	K400 (28d)	224×224	60.5	-
AVID (Morgado et al., 2021b)	R(2+1)D-18	K400 (28d)	224×224	60.8	87.5
ACC (Ma et al., 2020)	R3D-18	K400 (28d)	224×224	61.8	90.2
GLCM (Zeng et al., 2021)	R3D-18	K400 (28d)	224×224	61.9	91.2
SvaCLR (Ours)	R(2+1)D-18	K400 (28d)	224×224	66.8	92.2
SvaCLR (Ours)	R(2+1)D-18	VGGSound (23d)	224×224	**67.8**	**92.6**

4.3 Comparison with state-of-the-art

Following previous works (Patrick et al., 2021a; Morgado et al., 2021a), We evaluate the effectiveness of our approach on standard action recognition and video retrieval tasks on UCF101 (Soomro et al., 2012) and HMDB51 (Kuehne et al., 2011) datasets. In addition, we also propose to evaluate the learned audio-visual representations by an audio-video retrieval task, including video-to-audio retrieval and audio-to-video retrieval, on the K-Sounds dataset (Arandjelovic & Zisserman, 2017). We compare our proposed SvaCLR with other state-of-the-art approaches.

Action recognition. As mentioned above, after the audio-visual unsupervised pre-training, the learned representations can be transferred to downstream tasks to validate their effectiveness. Here we compare our proposed SvaCLR with other state-of-the-art approaches. We evaluate the learned representations on the action recognition task. The results are shown in Table 4.

From the results we can see that: **(1)** When pre-trained on a medium-scale dataset, K-Sounds, our approach significantly outperforms previous works. We improve performances on UCF101 and HMDB51 datasets by large margins, 9.2% and 14.3%. This demonstrates that our proposed co-augmentation method enlarges the diversity of the training views and benefits contrastive learning a lot. **(2)** Our approach demonstrates great scalability in terms of dataset size. When pre-trained on a large dataset K400, our approach exceeds the state-of-the-art audio-visual representation learning approach GDT (Patrick et al., 2021a) by a large margin, especially on the HMDB51 dataset, where we outperform GDT by 3.8%. Note that GDT applies hierarchical data augmentations while we only use one-speed augmentation. **(3)** Our approach also demonstrates scalability in terms of resolution. We can further improve the performances by using a large input size. We visualize the attention maps based on our pre-trained video encoder as shown in Fig. 5.

Video retrieval. In addition to the action recognition task, we also evaluate our method on video retrieval task. We compare our approach with the-state-of-art in Table 5, and show that our approach achieves competitive results with other methods. These results show that representations learned by our proposed approach can generalize well.

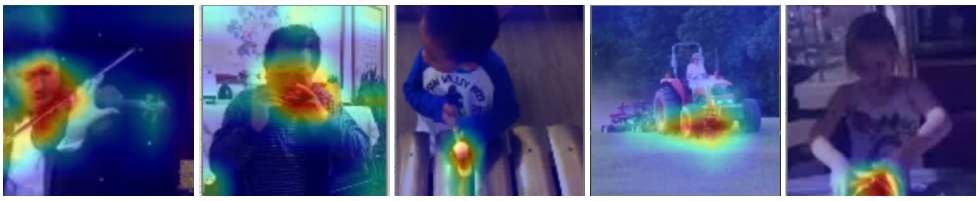

Figure 5: **Attention visualizations.** Attention computed from the Conv5 layer shows that the learned model is activated with the object that sounds. This validates that our method is able to capture audio-visual correspondence.

Table 5: **Comparison with state-of-the-art approaches on video retrieval task.** Approaches are evaluated on UCF101 Soomro et al. (2012) and HMDB51 Kuehne et al. (2011) datasets. Video samples in testing split are used to retrieve nearest neighbors from the training split. Standard video retrieval metrics R@K are reported.

Method	UCF101			HMDB51		
	R@1	R@5	R@20	R@1	R@5	R@20
VCOP (Xu et al., 2019)	14.1	30.3	51.1	7.6	22.9	48.8
VCP (Luo et al., 2020)	18.6	33.6	53.5	7.6	24.4	53.6
MemDPC (Han et al., 2020b)	20.2	40.4	64.7	7.7	25.7	57.7
CoCLR (Han et al., 2020a)	55.9	70.8	82.5	26.1	45.8	69.7
GDT (Patrick et al., 2021a)	57.4	73.4	88.1	25.4	**51.4**	75.0
STiCA (Patrick et al., 2021b)	59.1	76.2	88.1	26.3	49.2	76.4
SvaCLR (Ours)	**60.4**	**76.4**	**89.0**	**26.5**	50.8	**77.5**

Audio-Video Retrieval. To further evaluate the cross-modality ability of the proposed approach, we propose to use an audio-video retrieval task on K-Sounds (Arandjelovic & Zisserman, 2017). We compare to audio-visual contrastive learning with vanilla InfoNCE loss and current state-of-the-art GDT (Patrick et al., 2021a) in Table 6. We show that our approach achieves the best performances on both audio-to-video retrieval task and video-to-audio retrieval task. It is interesting to note that pre-trained on K-sounds can achieve better performance to retrieve top-1 nearest neighbor. But its ability to generalize to more visual-audio pairs is restricted that it performs worse than pre-training on a larger dataset K400 to retrieve top-5, top-10, and top-20 nearest neighbors.

Table 6: **Comparison with state-of-the-art approaches on video-audio retrieval task.** "Baseline" represents the vanilla audio-video contrastive learning with InfoNCE loss.

Method	Pre-train Dataset	Video → Audio				Audio → Video			
		R@1	R@5	R@10	R@20	R@1	R@5	R@10	R@20
Baseline	K400	2.6	13.8	26.1	44.3	3.6	13.7	25.3	41.0
GDT (Patrick et al., 2021a)	K400	3.1	14.3	24.6	40.8	3.6	16.1	26.4	43.9
SvaCLR (Ours)	K-Sounds	**5.1**	14.1	25.4	42.3	**4.7**	16.0	27.2	43.6
SvaCLR (Ours)	K400	4.6	**17.2**	**28.3**	**44.6**	4.2	**16.4**	**27.6**	**44.3**

5 Conclusions

The proposed speed augmentation significantly improves the performance of unsupervised audio-visual pre-training. We observed that speed co-augmentation leads to partial relationship between audio-visual pairs. This differs from the common intuition in single-modality representation learning, where samples are invariant to data augmentations. To combat this, we propose a cross-affinity module, which can adaptively model the cross-modality partial relationship and further improve performances. Extensive experimental results show that our approach achieves a new state-of-the-art on three downstream tasks.

Reproducibility Statement We provide the implementation details in Section 4.1. We provide the pseudo-code for speed augmentation in the Appendix. A code file including the implementation of the proposed approach can be found at https://anonymous.4open.science/r/ICLR-2023-SvaCLR-Anonymous.

REFERENCES

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

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. *arXiv preprint arXiv:2006.16228*, 2020.

Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems*, 2019.

Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *IEEE/CVF International Conference on Computer Vision*, 2017.

Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *European Conference on Computer Vision*, 2018.

Yuki Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. *Advances in Neural Information Processing Systems*, 2020.

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

Yusuf Aytar, Carl Vondrick, and Antonio Torralba. See, hear, and read: Deep aligned representations. *arXiv preprint arXiv:1706.00932*, 2017.

Henry S Baird. Document image defect models. In *Structured Document Image Analysis*, pp. 546–556. Springer, 1992.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *International Conference on Acoustics, Speech, & Signal Processing*, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *IEEE/CVF International Conference on Computer Vision*, 2015.

Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, 2013.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems*, 2020a.

Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, 2020b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

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.

Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association*, 2015.

Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems*, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *IEEE/CVF International Conference on Computer Vision*, 2011.

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

Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *arXiv preprint arXiv:2001.00294*, 2020.

Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Active contrastive learning of audio-visual video representations. In *International Conference on Learning Representations*, 2020.

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 *IEEE/CVF International Conference on Computer Vision*, 2019.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 2016.

Pedro Morgado, Ishan Misra, and Nuno Vasconcelos. Robust audio-visual instance discrimination. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021a.

Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021b.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 2016.

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

Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *European Conference on Computer Vision*, 2018.

Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Mandela Patrick, Yuki M Asano, Polina Kuznetsova, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. On compositions of transformations in contrastive self-supervised learning. In *IEEE/CVF International Conference on Computer Vision*, 2021a.

Mandela Patrick, Po-Yao Huang, Ishan Misra, Florian Metze, Andrea Vedaldi, Yuki M Asano, and João F Henriques. Space-time crop & attend: Improving cross-modal video representation learning. In *IEEE/CVF International Conference on Computer Vision*, 2021b.

Karol J. Piczak. Environmental sound classification with convolutional neural networks. *MLSP*, 2015a.

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

Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

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

Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *IEEE/CVF International Conference on Computer Vision*, 2015.

Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition-Volume 2*, pp. 958, 2003.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

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

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *European Conference on Computer Vision*, 2018.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019a.

Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision*, 2020.

Luyu Wang and Aaron van den Oord. Multi-format contrastive learning of audio representations. *arXiv preprint arXiv:2103.06508*, 2021.

Luyu Wang, Pauline Luc, Yan Wu, Adria Recasens, Lucas Smaira, Andrew Brock, Andrew Jaegle, Jean-Baptiste Alayrac, Sander Dieleman, Joao Carreira, et al. Towards learning universal audio representations. In *International Conference on Acoustics, Speech, & Signal Processing*, 2022.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *IEEE/CVF International Conference on Computer Vision*, 2015.

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019b.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*, 2018.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Ran Xu, Caiming Xiong, Wei Chen, and Jason Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Zhaoyang Zeng, Daniel McDuff, Yale Song, et al. Contrastive learning of global and local video representations. *Advances in Neural Information Processing Systems*, 2021.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, 2016.

A APPENDIX

A.1 ARCHITECTURE IN DETAIL

We provide the architecture details of the video encoder R(2+1)D-18 in Table 7 and the architecture details of the audio encoder ResNet-9 in Table 8.

Table 7: **The R(2+1)D-18 structure of the video encoding branch.**

stage	detail	output size $T \times HW \times C$
input data	-	$30 \times 128^2 \times 3$
conv_{11}	$1 \times 7^2, 45$ stride $1, 2^2$	$30 \times 64^2 \times 45$
conv_{12}	$3 \times 1^2, 64$ stride $1, 1^2$	$30 \times 64^2 \times 64$
conv_2	$\begin{bmatrix} 1 \times 3^2, 64 \\ 3 \times 1^2, 64 \end{bmatrix} \times 2$	$30 \times 64^2 \times 64$
conv_3	$\begin{bmatrix} 1 \times 3^2, 128 \\ 3 \times 1^2, 128 \end{bmatrix} \times 2$	$15 \times 32^2 \times 128$
conv_4	$\begin{bmatrix} 1 \times 3^2, 256 \\ 3 \times 1^2, 256 \end{bmatrix} \times 2$	$8 \times 16^2 \times 256$
conv_5	$\begin{bmatrix} 1 \times 3^2, 512 \\ 3 \times 1^2, 512 \end{bmatrix} \times 2$	$4 \times 8^2 \times 512$
pool_{avg}	global avg	$1 \times 1^2 \times 512$

Table 8: **The ResNet-9 structure of the audio encoding branch.**

stage	detail	output size $H \times W \times C$
input data	-	$40 \times 99 \times 1$
conv_1	$7 \times 7, 64$ stride 2,	$18 \times 48 \times 64$
pool_{max}	$3 \times 3, 64$ stride 2,	$8 \times 23 \times 64$
layer_1	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 1$	$8 \times 23 \times 64$
layer_2	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 1$	$4 \times 12 \times 128$
layer_3	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 1$	$2 \times 6 \times 256$
layer_4	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 1$	$1 \times 3 \times 512$
pool_{avg}	global avg	$1 \times 1^2 \times 512$

A.2 IMPLEMENTATION IN DETAIL

A pseudo-code for speed augmentation using PyTorch is as follows. A code file including the implementation of the proposed approach can be found at https://anonymous.4open.science/r/ICLR-2023-SvaCLR-Anonymous.

```
import random
from torchaudio.transforms import Resample
def video_speed_aug(video, start_frame, clip_len, video_speeds):
   # video_speeds is the speed candidate for video and audio
   new_video_sr = random.choice(video_speeds)
   video = video[start_frame:clip_len:new_video_sr]
   return video

def audio_speed_aug(audio, ori_audio_sr, audio_speeds):
   new_audio_sr = random.choice(audio_speeds)
   transform = Resample(ori_sr, new_audio_sr)
   audio = transform(audio)
   return audio
```

A.3 ANALYSIS OF CROSS-AFFINITY MODULE

To better understand what the cross-affinity module actually learns, we visualize (1) a histogram of the affinity scores from the cross-affinity module for the sped-audio and video from different samples in Fig. 6; (2) A boxplot of speed and cross-affinity weights in Fig. 7. These figures indicate that our proposed cross-affinity module can well model the partial relationship between the sped-up audio and video and a few examples are false negative samples due to the speed augmentation.

Qualitative Visualization. Based on the generated cross-affinity weights, we visualize four samples with the highest and lowest weights in Fig. 8. We find that video-audio pairs with the lowest weights

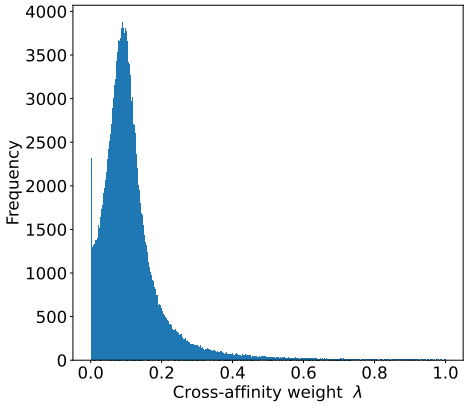

Figure 6: **Histogram of cross-affinity weights between sped-up audio and other video examples.**

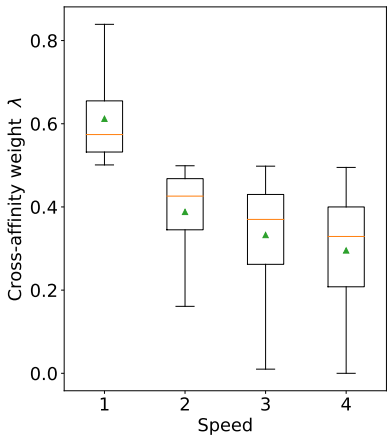

Figure 7: **boxplot of speed and cross-affinity weights.** Blue triangle represents mean and orange line represents median.

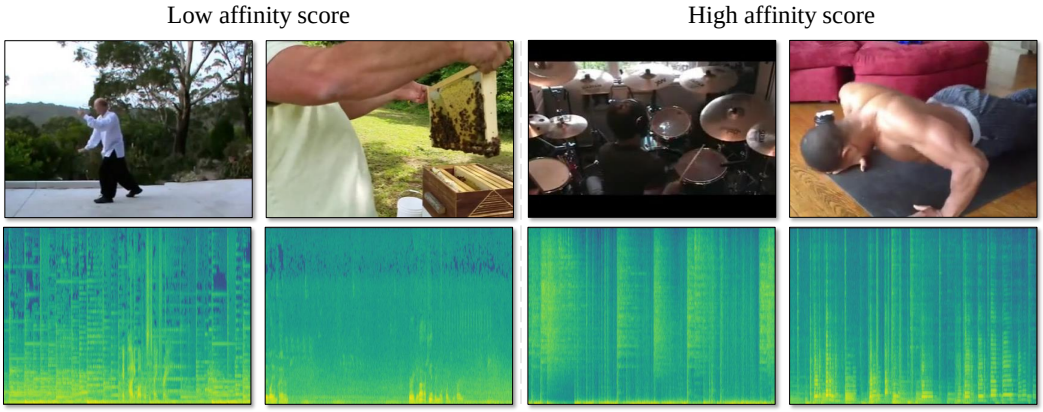

Figure 8: **Video samples and spectrograms with the highest and lowest affinity score.**

are videos contain noisy audio, e.g., taichi with background music and bee keeping with human talking. On the other hand, video-audio pairs with the highest weights are highly related, e.g., playing drums and push up.

A.4 Audio Classification

For completeness, we evaluate the learned audio encoder on audio classification task using DCASE Stowell et al. (2015) and ESC50 Piczak (2015b) datasets as shown in Table 9. It can be seen that our approach achieves competitive results with other methods.

A.5 Audio-Video Retrieval

Implementation details. In Section 4.3, we present a new downstream task, audio-video retrieval, to evaluate the cross-modality ability of the learned representations. We use the validation set of K-sounds as the evaluation dataset. Two settings are considered, audio-to-video retrieval and video-to-audio retrieval. For each setting, we uniformly sampled 10 clips of both audio and video from one sample. Then the sampled clips are fed into the video and audio encoders to extract representations from the last pooling layer (pool5). Based on the extracted representations, cosine distances are computed. Finally, the retrieval performance is evaluated by querying top-k nearest neighbors from

Table 9: **Audio classification.** Audio representations are evaluated on standard audio classification benchmarks using DCASE Stowell et al. (2015) and ESC50 Piczak (2015b).

Method	Pretraining	Acc%	
		DCASE	ESC50
Autoencoder (Aytar et al., 2016)	-	-	39.9
Random Forest (Piczak, 2015b)	-	-	44.3
Piczak ConvNet (Piczak, 2015a)	-	-	64.5
RNH (Roma et al., 2013)	-	72	-
Ensemble (Stowell et al., 2015)	-	77	-
AVTS (Korbar et al., 2018)	K400	91	76.7
XDC (Alwassel et al., 2019)	K400	–	78.0
AVID (Morgado et al., 2021b)	K400	93	79.1
ACC (Ma et al., 2020)	K400	–	79.2
SvaCLR (Ours)	K400	**94**	**79.5**

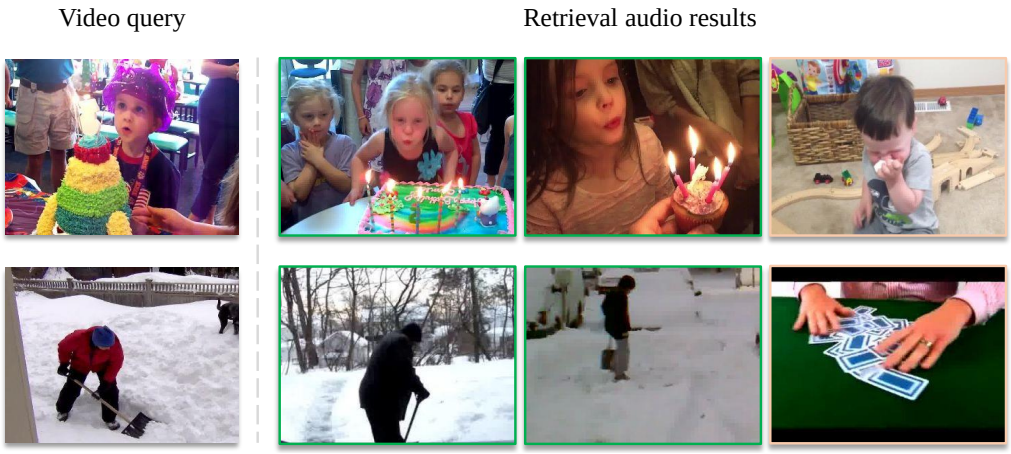

Figure 9: **Qualitative video-to-audio retrieval results.** The correctly retrieved results are marked in green while the failure cases are in orange. Better visualization with color.

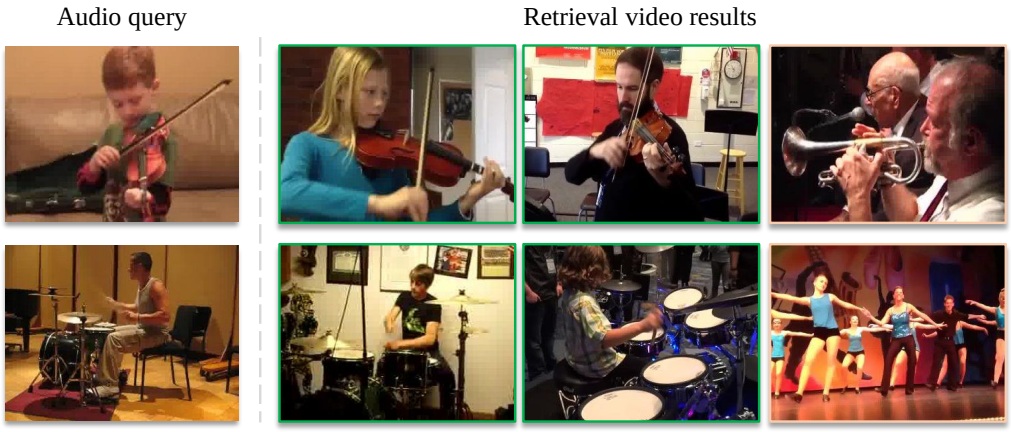

Figure 10: **Qualitative audio-to-video retrieval results.** The correctly retrieved results are marked in green while the failure cases are in orange. Better visualization with color.

the other modality samples based on cosine distances. Here, we consider k to be 1, 5, 10, 20. If the test clip class label is within the top-k retrieval results, it is considered to be successfully retrieved.

Visualizations. Figures 9 and 10 show qualitative retrieval results. We investigate some failure cases of the retrieval results and we find that networks can be confused by samples with similar sounds and thus retrieve those samples. For example, "playing drums" is confused with "tap dancing", and "shoveling snow" is confused with "shuffling cards".

A.6 CLASSES OF KINETICS-SOUNDS DATASET

Note that the original Kinetics-Sounds dataset described in (Arandjelovic & Zisserman, 2017) consists of 34 action classes, where a few classes got removed currently. Therefore, after discussing with the authors, we retain the following 30 classes: *blowing nose, bowling, chopping wood, ripping paper, shuffling cards, singing, tapping pen, blowing out candles, dribbling basketball, laughing, mowing lawn, shoveling snow, tap dancing, tapping guitar, tickling, playing accordion, playing bagpipes, playing bass guitar, playing clarinet, playing drums, playing guitar, playing harmonica, playing keyboard, playing organ, playing piano, playing saxophone, playing trombone, playing trumpet, playing violin, playing xylophone.*