6. Appendix

6.1. Implementation Details

While videos in Kinetics are 10 seconds long, we randomly sample either 1-second (30 frames), or 2-second (60 frames) clips from the 30fps videos. For the R(2+1)-D-18 visual encoder, the dimensions of the $res5$ feature map before spatial pooling is 512 x T x 7 x 7 for a 112 x 112 resolution video, where $T = 4$ for 30-frame (1 second) input, and $T = 8$ for 60-frame (2 second) input. After spatial pooling, we use either average pooling or a transformer as the temporal pooling function for the visual encoder, but always use average pooling for the audio encoder. The transformer's layers dimensionality are set to 512-D. Both encoders produce a fixed-dimensional representation vectors after temporal aggregation (512-D). Both vectors are then passed through two fully-connected layers with intermediate size of 512 to produce 256-D embedding vectors $\boldsymbol{z}$ as in [106]. We use these embeddings in our loss eq. (7) and train our model for 100 epochs. For the visual component of the video, we use a 30 frame RGB clip as input, at 30 fps covering 1 second. The video clip has a spatial resolution of 112×112 pixels. For input data augmentation, we apply random crops, horizontal flips, Gaussian blur and color jittering, all clip-wise consistent, following the protocol of SimCLR [24], and we ablate multiple settings for spatial and temporal feature cropping sizes. For the audio input, we extract a 1-second log-mel spectrogram of dimension 257×199 starting at the same time as the visual component. We also apply volume jittering to increase the robustness of our audio features. We optimize this model using SGD with momentum 0.9, weight decay 10^{-5} and learning rate 0.64, with a warm-up period of 10 epochs. For NCE contrastive learning, the temperature τ is set as 0.1 for cross-modal loss, and 0.5 for the within-modal loss. We use a mini-batch size of 8 on each of our 64 GPUs giving an effective batch size of 512 for distributed training. In our ablations, we evaluate the learned representation by fine-tuning the visual encoder on fold 1 of the HMDB-51 [77] action recognition dataset.

6.1.1 State-of-the-Art Experiment Details

For our state-of-the-art model, we train for 100 epochs, using R(2+1)-D-18 visual encoder with transformer temporal attention pooling, and Resnet-9 for audio encoder. We use 60 frames as input, and feature-crop augmentation (space: $2\times6^2+4\times4^2$ & time: $2\times3+1\times2$).

6.2. Transformer Architecture Details

We use a 2-layer transformer, with 4 attention heads, and hidden dimension 512. The input to the transformer is the spatially averaged output of the last convolutional layer of R(2+1)D-18 video backbone. The transformer contextualizes features across time to output a fixed feature length representation of dimension 512, which is then passed to MLP head for contrastive learning. While transformers generally benefit from being optimized with Adam [147], we adhere to using SGD for simplicity. We also do not observe any stability issues, likely because the transformer is quite shallow.

Method	**Pretraining**	**Acc%**	
		DCASE	ESC50
Autoencoder [12]	-	-	39.9
Random Forest [109]	-	-	44.3
Piczak ConvNet [108]	-	-	64.5
RNH [116]	-	72	-
Ensemble [121]	-	77	-
ConvRBM [117]	-	-	86.5
AVTS [74]	K400	91	76.7
XDC [6]	K400	–	78.0
AVID [97]	K400	93	79.1
ACC [90]	K400	–	79.2
Ours: STiCA	K400	**94**	**81.1**
SoundNet [12]	SNet	88	74.2
L3-Net [7]	SNet	93	79.3
AVTS [74]	SNet	94	82.3
DMC [60]	SNet	–	82.6
AVTS [74]	AS	93	80.6
XDC [6]	AS	–	85.8
MMV [5]	AS	–	86.1
AVID [97]	AS	96	89.2
GDT [106]	AS	**98**	88.5
ACC [90]	AS	–	**90.8**
Human [109]	–	–	81.3

Table 6: Audio classification. Downstream task accuracies on standard audio classification benchmarks on DCASE2014 and ESC50. Dataset abbreviations **A**udio**S**et, **K**inetics**400**, **S**ound**Net**,

7. Additional experiments

7.1. Audio Classification

For completeness, we also present audio classification results on ESC-50 [108] and DCASE-2014 [122]. ESC-50 [109] is an environmental sound classification dataset which has 2K sound clips of 50 different audio classes. ESC-50 has 5 train/test splits of size 1.6K/400 respectively. DCASE2014 [122] is an acoustic scenes and event classification dataset which has 100 training and 100 testing sound clips spanning 10 different audio classes. We demon-

Method	**Architecture**	**Dataset**	**Top-1 Acc%** HMDB	UCF
RotNet3D [65]	S3D	K600	24.8	47.7
CBT [125]	S3D+BERT	K600	29.5	54.0
MemDPC [53]	R-2D3D	K400	30.5	54.1
AVSF [143]	AVSF	K400	44.1	77.4
CoCLR [54]	S3D	K400	46.1	74.5
Ours: STiCA	R(2+1)D-18	K400	**48.2**	**77.0**
MIL-NCE [93]	S3D	HT	53.1	82.7
XDC [6]	R(2+1)D-18	IG65M	56.0	85.3
MMV [5]	R(2+1)D-18	AS	60.0	83.9
ELo [110]	R(2+1)D-50	Y8M	64.5	–

Table 7: Comparison to state-of-the-art. Transfer learning results on UCF-101 and HMDB-51 when video backbone is frozen.

strate competitive performance relative to the state-of-the-art, despite training on a much smaller and less audio-rich Kinetics-400 dataset. We extract 10 equally spaced 2-second sub-clips from each full audio sample of ESC-50 [109] and 60 1-second sub-clips from each full sample of DCASE2014 [122]. We save the activations that result from the audio encoder to quickly train the linear classifiers. We use activations after the last convolutional layer of the ResNet-9 and apply a max pooling with kernelsize (1,3) and stride of (1,2) without padding to the output. For both datasets, we then optimize a L2 regularized linear layer with batch size 512 using the Adam optimizer [72] with learning rate 1x10^{-4}, weight-decay set to 5x10^{-4} and the default parameters. The classification score for each audio sample is computed by averaging the sub-clip scores in the sample, and then predicting the class with the highest score. The mean top-1 accuracy is then taken across all audio clips and averaged across all official folds.

7.2. Linear probing results

In Tab. 7, we compute the linear classification results of our model compared to other recent methods. We find that our best model has competitive 3-fold linear evaluation results of 48.2% on HMDB-51 and 77.0% on UCF-101.

7.3. Supervised training on K-400

Here we experiment with training supervisedly on Kinetics-400 and observing the effect of using feature cropping (with the configuration 2 medium and 2 small latent space crops). The experimental results are given in Tab. 8 We find that even though our method is designed for contrastive cross-modal pretraining, using feature crops can help in training in a supervised manner too.

Fm-Crop	HMDB-51 Top-1 Acc.
✗	67.6
✓	69.0

Table 8: Supervised Training. We train the R(2+1)D+Transformer architecture supervisedly on Kinetics-400 with and without feature crops enabled.

7.4. Audio-Visual Heatmap Visualizations

In Fig. 3, we show examples that our model truly learned some spatial correspondence between a region and audio. We have done this by visualizing the strength of the dot-product of the visual feature map (without pooling) with the audio feature vector.

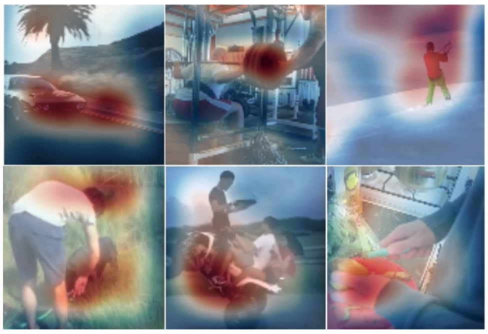

Figure 3: Heatmap visualizations. Heatmaps are obtained by removing the spatial pooling layer and visualizing the strength of the dot-product of the audio feature vector with the video feature-map as in [8]. Here, we show selected samples from Kinetics-400 training set of the resulting heatmaps along with the middle frame of the video.

7.5. Preventing Shortcut Learning with Feature Crops.

Noise contrastive learning works better when you can reduce the mutual information between the input pairs [131] as its harder for the network to cheat. This can be achieved by taking multiple spatial crops of images in the input space and independently applying different augmentations, such as color jittering and Gaussian blurring, to the cropped inputs. However, as mentioned above, taking more than 2 crops in input space is both memory and computationally infeasible for multi-modal video data. Crops in feature space, on the other hand, allows us to take multiple crops for noise contrastive learning. However, since CNNs have large receptive fields that easily cover the full frame, there may be shortcut learning with feature crops as information may leak between the crops from same feature map. To alleviate this, we take feature crops from two originally augmented video clips, allowing us to make NCE comparisons *across* modalities and individual augmentations (such as color jitter), leading to a beneficial reduction in mutual information. Furthermore, while the theoretical receptive fields of units in later layers are indeed very large, units tend to be sensitive to an effective area which is significantly smaller than the theoretical receptive field [89, 153], further reducing the mutual information between inputs for noise contrastive learning.