Efficient Active Speaker Detection

Wenyi Jiang*, Zilin Guo*, Ruijie Zhang*, Ningrui Zhou*
Georgia Institute of Technology
Shenzhen, China
{wjiang311, zguo370, rzhang641, nzhou67}@gatech.edu

Abstract

*Nowadays, many models have achieved great performance on Audio-Visual Diarization tasks. However, this usually means huge parameter numbers and longer training time. These two factors significantly limit applications in real industrial scenarios. Thus, in this work, we aim at addressing these two problems with only a small cost on performance. For this model, we replace the audio encoder with **SincNet**, the extraction part in the visual encoder with **VGG16**, and the attention layer with **Audio-visual Cross-External-Attention**, which accelerates the speed of training hugely and decreases the complexity of models. Finally, we discuss and analyze corresponding **data augmentation** and **visual encoder** replacement results. Our model finally achieved **84.06** mAP on the Amazon AVD dataset, but its training time is just **25** % of the benchmark. The demo video and code are available in supplementary materials.*

1. Introduction

Audio-Visual Diarization (AVD) is a task combining computer vision and audio. In a video clip, there are many people speaking. If we can distinguish between the voices, we can record the diary automatically or concentrate on the talker. In this task, the bounding boxes of persons have been provided. We want to utilize the visual information and audio clip to localize which person spoke and when they spoke. Improving the former work's accuracy and speeding up the training processes are also what we want to realize. In this paper, we try to realize the goal and find out the answer to the above question. We did ablative experiments to decouple different parts of the model. Also, we try to make the former ASD identification technique more useful and accurate.

Former works use the motion of the lip to detect speakers and cluster the audio clips. They normally match the bounding boxes and voice by linear assignments or Bayes approaches. These approaches are limited when the lip is occluded. Many recent works try to combine both picture and voice information. Also, many attempts have been made to take the most advantage of the constant videos. Min used graphs to help the model get the long temporal context. [9] Tao, in his paper, describes a method of cross-attention for multi-modality interaction.[13]

Our work aims to lightweight the former work and speeds up the training processes, which can help the whole model be affordable for mobile devices. The related industry and military, which need to detect special people who are talking secretly, will think our work useful.

The AVA-ActiveSpeaker-dataset here are part of Hollywood movies[11]. It contains 29,723, 8,015, and 21,361 video utterances in the training, validation, and test sets, respectively. The video utterances range from 1 to 10 seconds and are provided as face tracks. We follow the official evaluation tool and report the performance in terms of mean average precision (mAP). There are several challenges involved in the AVA-ActiveSpeaker dataset. The language is diverse, and the frame per second (fps) of the movies varies. Furthermore, a significant number of videos have blurry images and noisy audio. It also contains many old movies with dubbed dialogues. All these factors make it hard to synchronize the audio-visual signals accurately.

2. Related Works

There are some previous works related to our research, including audio classification,...

video Video classification needs to detect which persons' lips have the features of talking. And take consideration context is also needed because talking is a constant behavior. VGG[8] is a traditional network used to get features of input images and with more parameters. Simonyan adds ResNet into CNNs and tries to solve problems with the deeper networks that will lead to worse performance[12]. ResNeXt uses repeating blocks to modify ResNet and shows better performance on ILSVRC 2016 than ResNet.[14]

audio Audio classification is a task that distinguishes the different kinds of voice, ranging from people's voices, animal voice, noise, music, and so on. Given a clip of voice, previous works [1, 5] utilize CNN and Transformer to capture the feature of voice. The main challenge in audio classification is that of the background noise. The main voice can be accompanied by other noise voices, which lower the model's performance. Another problem is that the Model in previous works are too heavy to be deployed on an mobile device. So it's not suitable for edge computing. The SincNet[10] mitigate this problem by detecting the lowest and highest frequencies rather than all frequencies. In this way the SincNet has much fewer parameters which make it available for application on mobile device.

3. Approach

ff

3.1. Pipeline

Following the previous works[10, 2, 3], we established the pipeline as Figure 1. We proposed an end to end model with the audio encoder, visual encoder and the Cross-Externel-Attention. We extracted the audio into WAV file from the origin video. And We cropped the face bboxs of each frame and resized them to 112*112. Took the audio file and cropped image as input, both the audio encoder and visual encoder will generate the embedding of each frame with 128 dimensions. The audio feature help us classify whether there is someone speaking. And the visual feature aims at capturing the motion of speaker's lips. We update the visual and audio feature by each other separately using an attention layer. Our pipeline keeps the video's temporal information. In this way, the two domain features are able to gain useful information through the other domain's context. We concatenated them together, and fed to a linear layer to classify every frame into 2 classes (speaking or not speaking).

Our implementation is modified from the TalkNet[10] implementation on GitHub(https://github.com/TaoRuijie/TalkNet-ASD). For the baseline, we use the ResNet34 with SE modules as the audio encoder and ResNet18 with the V-TCN module as the visual encoder. We managed to improve the baseline by using the other backbones to extract features, adding more data augmentations, and designing an attention mechanism to fully use the temporal information. The TalkNet repo mainly helps us with the basic training process like tackling the datasets' format, defining an optimizer, and measuring our model's performance. We have completely designed the new model structure by ourselves.

3.2. Audio Encoder

Previous works usually use ResNet as the audio Encoder and MFCC as the input. However, though ResNet has a strong ability to analyze the audio, it is heavy for a mobile phone. The combination of MFCC and ResNet is time-consuming and asks high computational resources. In this way, we replace it with a SincNet. SincNet has about 1/5 parameters of ResNet34 and does not require the MFCC as the preprocessing. SincNet is well-designed to deal with audio-related work and reported slightly higher performance than ResNet. As shown in Figure 2, it has a To further speed up the inference process, we implemented the Depthwise Separable Convolution(DSConv) to cut off the size of the Model. DSConv can decouple the goal of weighting different channels and learning from spatial information.

3.3. Visual Encoder

The structure of the visual encoder is shown in Figure 3 Previous work used ResNet18 for feature extraction. In order to realize accurate improvement, I want to use ResNeXt18 to exchange ResNet18. ResNeXt has a similar number of parameters but higher accuracy than ResNet. Its networks' architecture can also help speed up the training phase under certain conditions. So I think it may increase the former work's performance by replacing the pipeline's visual feature extension part with ResNext. Also, the parameters ResNeXt uses are smaller than ResNet. The differences between ResNet and ResNeXt's blocks are described in Figure 4. However, after I made the attempt, the results after the same epoches are worse than ResNet18. The original idea didn't work. I think its structure may be too complex and I tried VGG16 replacing Resnet18 to find out if a simple model would work better. And the VGG16 achieves a result just a little worse than ResNet18 after fewer epochs. Although it does not make results better, it can speed up the network to get the final model using less time.

3.4. Data Augmentation

Lots of previous works have proven data augmentation plays an extremely significant role in image-related training. In this work, we add many data augmentation mechanisms like image shifting, Gaussian blur noise, unsharpen mask, and box blur noise as data augmentation strategies to help model training. Image shifting is shifting our original image to the upper left position by small pixels. Gaussian blur noise is to process every pixel by Gaussian function which will reduce image details. Unsharpening mask is to make the image less blur but may lose some original details. Box blur is to take average neighborhood pixel values for every pixel in a small box range. These mechanisms will help the model to learn a more general representation of images and show more robustness on the testing dataset.

3.5. Cross-External-Attention

In [13], the audio-visual cross-attention they implemented are complicated, and we cut their attention layer including cross-attention and self-attention as the baseline. Inspired by [6], we design three different attention layers,

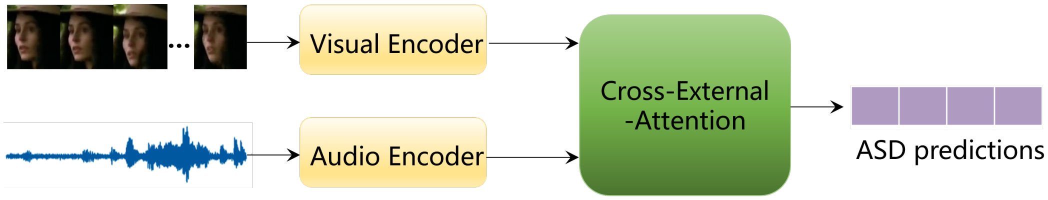


Figure 1. The pipeline of our proposed solution. The video's temporal relation is preserved to match the visual and audio feature. The predictions are the predicted classes for each frame.

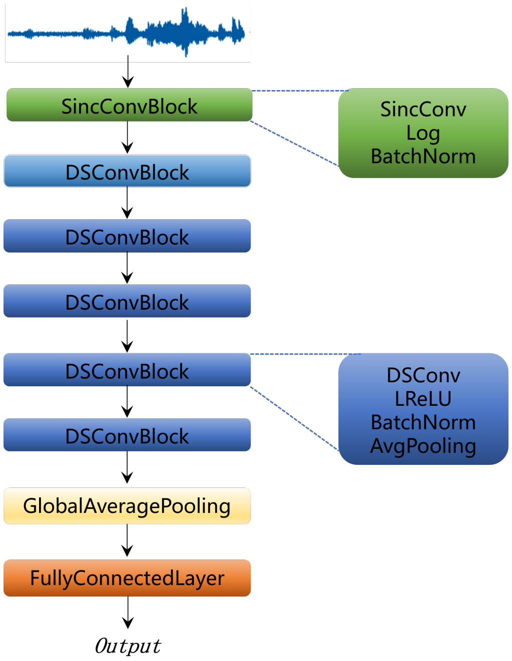


Figure 2. The architecture of SincNet.

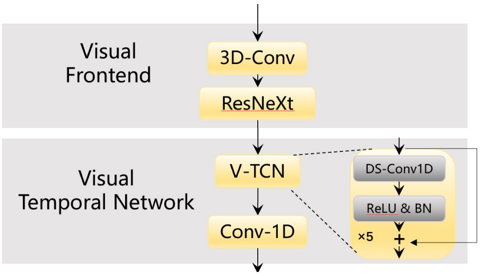


Figure 3. The architecture of visual encoder.

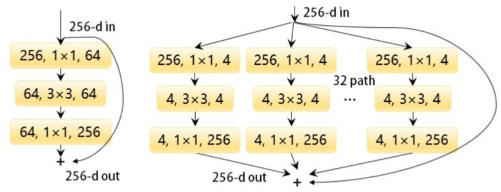
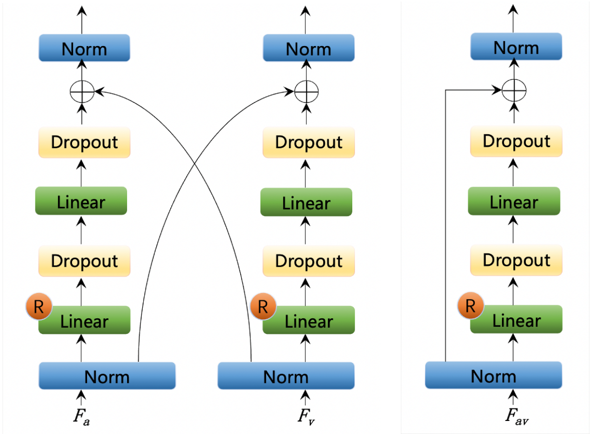


Figure 4. Resnet block(left) vs ResNeXt block(right)

train them and select the best one as our attention layer. As Figure 5 (a) shows, in cross-attention, the features of audio embeddings F_a firstly are normalized and pass a linear layer with ReLU activation function, subsequently, they are dropout followed by a linear layer and a dropout layer, and added by the normalization of the features of visual embeddings F_v, eventually normalized to obtain audio cross-attention features $F_{v \to a}$. Similarly, visual cross-attention features and audio (visual) self-attention can be generated.

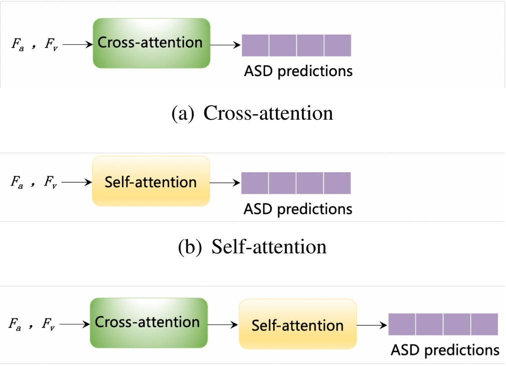
The main difference between cross-external-attention (CEA) and normal cross-attention (NCA) is that instead of using Eq.(1), CEA captures the long-range dependency by using two external and shared memories which are two linear layers and two normalization layers. Based on the architectures of Figure 5, we design tree different attention layers, as shown in Figure 6. After ablation experiments, the cross-attention layer outperforms the others and is implemented as the attention part of our model.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{V}})V \qquad (1)$$



(a) Cross-attention (b) Self-attention

Figure 5. (a) The attention layer in the cross-attention network. Considering the audio embeddings F_a as the source, and the visual feature F_v as the target, we generate audio attention feature $F_{a \to v}$ as the output. Similarly, we generate visual attention feature $F_{v \to a}$. (b) The attention layer in the self-attention network.



(a) Cross-attention

(b) Self-attention

(c) Cross-self-attention

Figure 6. Tree different attention layers.

4. Experiments and Results

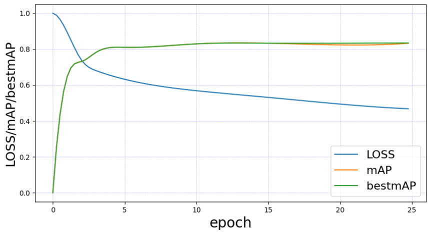


Figure 7. Figure of Training Process. This figure presents the Loss, mAP and bestmAP's curves during the training. We can find that the model did not overfit and reached an convergence.

Shown as Figure 7, Our model finished training in 6 hours with 25 epochs on an RTX5000. We measure our model's performance with its accuracy and speed. For the model's accuracy, we chose mean Average Precision(mAP) which can evaluate the model's performance on recognition. For the speed part, we evaluated our model with the amount of parameters and floating-point operations(FLOP). Our goal is to trade off between the model's effectiveness and speed. We aimed at proposing a model which is both light and accurate. More precisely, our goal is reaching 83% on mAP and reducing the FLOP by at least 30%. To achieve this goal, we focus on reducing the computational resources on audio encoder. A light model may suffer from lower accuracy. To compensate the accuracy loss on audio encoder, we tried to improve the effectiveness of visual encoder and bring attention mechanism.

4.1. Effectiveness

As shown in Table 1, our work has a slightly lower effectiveness than state-of-the-arts results. However it still satisfy the requirement of application in AVD task.(We have an demo video to show it's useful for detecting the active speaker.) Our work is suitable for the deployment on a mobile device because it is light.

Table 1. ActiveSpeaker Model's overall effectiveness performance. Our work is slightly lower than the SOTA. The TalkNET's result is according to our implementation without attention and other 2 works are directly borrowed from their paper.

Algorithms	mAP (%)
Chung et al. [3]	**87.8**
TalkNET[10]	87.6
MAAS-LAN[2]	85.1
Ours	84.06

Table 2. Visual Encoder's mAPs and loss on different feature extension after 5 epochs.

Model	BestmAP(%)	Loss
ResNet18	82.59	0.60700
ResNeXt18	60.60	0.83779
VGG16	**84.77**	**0.59249**

Why does the newly introduced features weaken the effectiveness? We mainly attributed it to our implementation. Because we used an open-source code to establish our work, our hands are tied to modify how the datasets are loaded. The clips are organized in tracks which has different lengths, ranging from 1 seconds to 10 seconds, it made the audio encoder hard to extract a consistent feature to represent the given audio clip. We managed to mitigate this problem by group the data by different window sizes, which means that group a number(1 to 16) of frames together for our audio encoder. Unfortunately, we can not solve all the correlated problems among different modules.

4.2. The Efficiency of ResNext18 and VGG16

We compare replacing the visual feature extension network with ResNeXt18 and VGG16. As shown in Table 2. After the same epochs, VGG16 get better results than the other two nets earlier. After the first 5 epochs, VGG16 gets the best mAPs(84.77%) and the lowest loss(0.59249), which is similar to the ResNet18's final results after 25 epochs. The ResNeXt18 got the worse mAPs(60.60%) and the highest loss(0.83779). It does not work on our task. Therefore, finally, we choose VGG16 to replace ResNet18 as the visual feature extension part in our codes.

4.3. The Efficiency of SincNet

We compared our audio encoder with other two typical structure ResNet and Transformer. As shown in Table 3. SincNet has a lower magnitude complexity than others. Transformer has 85.3M parameters and consumes heavy computational resources. The ResNet34 with SE modules contains 1.4M parameters comparing with the 0.12M parameters in SincNet. We shrank the overall parameters by about 45%. And the overall Flop was reduced to the half. The efficiency of our audio encoder structure strongly speed up our training and inference. On our RTX5000, An epoch training can be finished in 15 minutes, comparing to 1 hour using ResNet34 instead.

Table 3. Audio Encoder's parameter and FLOP. The FLOP is calculated for encoding a length of 2 seconds video clip.

Algorithms	Parameters	FLOP
AST[4]	85.3M	1.3T
TalkNET[10]	1.4M	8.7G
Ours	121.0K	86.2M

4.4. Ablation study of attention mechanism

We train baseline and these three different models (Figure 6) and gain their scores in Figure 8 and Table 4. It can be seen that both cross-attention and cross-self-attention have good performance. Although the bestmAP and loss of cross-self-attention are slightly better than those of cross-attention, cross-attention has faster rate of convergence since it only needs 16 epochs to reach the bestmAP (90.63%). Therefore, cross-attention and cross-self-attention are implemented in the model we improved respectively to find out which attention mechanism is more appropriate. Table 5 confirms the efficiency of cross-attention (Figure 5. (a)) and we decide to use it as our attention layer.

Table 4. Scores of tree different attention models and baseline. All the models are trained for 25 epochs.

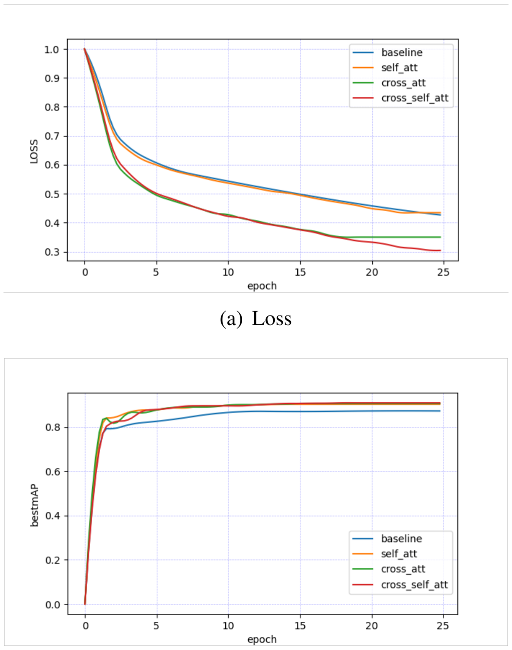
Model	Epoch	BestmAP(%)	Loss
Baseline	20	87.27	0.458299
Cross-attention	**16**	**90.63**	**0.370400**
Self-attention	18	90.28	0.467746
Cross-self-attention	**18**	**90.94**	**0.346203**

Table 5. Our net Overall effectiveness performance of two attention mechanisms.

Attention mechanism	mAP(%)
Cross-attention	**84.06**
Cross-self-attention	83.46

4.5. Ablation study of Data Augmentation

In the ablation study of data augmentation, we add many common mechanisms used in [7]. These include image shifting, Gaussian blur noise, unsharpen mask, and box blur noise. We compare original models with modified models that include new data augmentation mechanisms. Results are shown in Table 6. As we observe from experiments, the newly added data augmentation mechanism doesn't improve the performance of models and even underperforms compared with the benchmark. There are two possible reasons behind it. On the one hand, the original feature extractor has extracted and cropped images very well. It can extract actors' faces accurately and transforms them into the input. Thus, more mechanisms make no sense. On the other hand, our new data augmentation mechanism changes the potential data distribution of the model, so performance on the test dataset is not good.



(a) Loss

(b) BestmAP

Figure 8. Scores curves of tree different attention models and baseline.

Table 6. Data augmentation performance ablation experiments between TalkNet and ours

Epochs	5	10	20
TalkNET[10]	82.59	86.59	87.27
Ours	84.82	86.09	86.93

5. Conclusion

In this essay, we let Audio-Visual Diarization related model more applicable to real industrial scenarios and solve this problem from three perspectives. In the model structure, we replace the audio encoder with SincNet, the visual encoder's feature extraction model with VGG16, and the original attention mechanism with Cross-External-Attention. These two changes significantly decrease the training workload. Meanwhile, we also add new data augmentation mechanisms to improve the performance of the model in order to limit the previous costs on performance. Experiment results show they don't work, but we think they still improve the robustness of the model in a more complex and real-world application.

6. Work Division

Please add a section on the delegation of work among team members at the end of the report, in the form of a table and paragraph description. This and references do **NOT** count towards your page limit. An example has been provided in Table 7.

References

[1] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021. 2

[2] Juan León Alcázar, Fabian Caba, Ali K Thabet, and Bernard Ghanem. Maas: Multi-modal assignation for active speaker detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 265–274, 2021. 2, 4

[3] Joon Son Chung. Naver at activitynet challenge 2019–task b active speaker detection (ava). *arXiv preprint arXiv:1906.10555*, 2019. 2, 4

[4] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021. 5

[5] Yuan Gong, Yu-An Chung, and James Glass. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3292–3306, 2021. 2

[6] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 5

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv: Computer Vision and Pattern Recognition*, 2015. 1

[9] Kyle Min, Sourya Roy, Subarna Tripathi, Tanaya Guha, and Somdeb Majumdar. Learning long-term spatial-temporal graphs for active speaker detection. 2022. 1

[10] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, 2018. 2, 4, 5

[11] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew C. Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, and Caroline Pantofaru. Ava active speaker: An audio-visual dataset for active speaker detection. *international conference on acoustics speech and signal processing*, 2020. 1

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. 1

[13] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3927–3935, 2021. 1, 2

[14] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *computer vision and pattern recognition*, 2016. 1

Table 7. Contributions of team members.

Student Name	Contributed Aspects	Details
Ruijie Zhang	Implementation and Analysis	Implemented and analyzed the effect of three different visual feature extraction models' effectiveness in the visual encoder part, see it in visualEncoder.py. Find the model which can help the model converge faster and speed up the training process. Analyzed the efficiency of VGG16 compared to other visual models. See Section 3.3 and Section 4.2.
Ningrui Zhou	Implementation and Analysis	Implemented seven different data augmentation ways. From them, select the 4 best data augmentation mechanisms. See load_visual function in dataloader.py. Conducts 11 different experiments to evaluate their contribution to the model. Analyzed corresponding results and figure out reasons. See Sections 3.4 and 4.5
Zilin Guo	Implementation and Analysis	Implemented Audio-visual Cross-External-Attention, see it in attentionLayer.py. Analyzed the effect of different attention architectures and selected the best one as the attention layer of our model, see Section 3.5 and Section 4.4. Plot all the figures in this article.
Wenyi Jiang	Implementation and Visualization	Updated the audio encoder with SincNet to speed up the training process, see sincnet.py in models. Modified the data loader to generate temporal group data, see load_audio function in dataloader.py. Visualized the results with a demo video, see demo.mp4 in supplementary files. Analyzed the efficiency of SincNet comparing to other audio encoder, see Section 3.2.Audio Encoder and 4.3.The Efficiency of SincNet.