Acoustic-based Lip Reading for Mobile Devices: Dataset, Benchmark and A Self Distillation-based Approach

Yafeng Yin, *Member, IEEE,* Zheng Wang, Kang Xia, Lei Xie, *Member, IEEE,* and Sanglu Lu, *Member, IEEE*

Abstract—Speech is a natural communication way between people and a good way for human-computer interaction. However, speech with audible voices often faces the following problems, e.g., being affected by surrounding noises, breaking the quiet environment, leaking privacy, etc. Therefore, silent speech was proposed, especially lip reading, which aims to recognize speech content based on lip movements. In this paper, we utilize inaudible acoustic signals generated from mobile device to sense and recognize lip movements for lip reading. Considering the lack of public dataset in acoustic-based lip reading, we propose and release a large-scale lip-reading dataset LIPCMD with 30000 acoustic-based recordings. To advance the further research in lip reading, we provide benchmark evaluation on LIPCMD, while using traditional machine learning solutions and recent deep learning approaches. To recognize weak acoustic signals as words for lip reading, we propose a self distillation based approach *LipReader*, which distills the probability distribution and attention map in convolutional neural network itself for better classification. Finally, we implement *LipReader* on smartphone and evaluate it on LIPCMD dataset as well as under complex scenarios. Experimental results show that *LipReader* can achieve a good recognition accuracy for lip reading, i.e., 91.58%, while outperforming baseline solutions and existing work.

Index Terms—Acoustic-based Lip Reading, Dataset, Benchmark, Self Distillation.

1 Introduction

As a natural and convenient communication way between people, speech plays an important role in daily communication. Recently, with the advancement of smart devices, speech also makes a good contribution to Human-Computer Interaction (HCI). There have emerged many kinds of speech-driven applications or services, e.g., Siri in iPhone, voice assistant in Google Maps, voice messages in social softwares. When speaking to the device, the device will accordingly provide the queried information, perform the action, deliver the message, etc. In most of the speech-based interactions, the user needs to speak with audible voices, then the device recognizes the speech content based on audios and provides the corresponding services. However, speaking with audible voices often faces the following problems: the audible voices can be easily affected by ambient noises which make it difficult for speech recognition; speaking with audible voices may not be allowed in quiet environments (e.g., library), speaking with audible voices in public may leak the privacy information of user.

To address the above issues caused by audible voices, silent speech was proposed for human-computer interactions. Silent speech conveys the speech content without audible voices. Consequently, we can not recognize speech content with audios. To address this problem, lip reading [1] [2] [3] [4] [5], facial muscle vibration capturing [6], tougue tracking [7], brain-computer interface [8] were proposed for silent-speech recognition. Among the solutions, lip reading is one of the most well-known solutions [5], it aims to recognize the speech content based on lip movements. In regard to lip movements, they were often captured by camera and represented with image sequence or videos. Nevertheless, these vision-based lip reading solutions often have the following limitations: extracting the lip area often requires the whole-face image which may lead to the invasion of privacy, the image quality is easily affected by light conditions and the solutions can hardly work with poor light conditions, the computation overhead of image processing is heavy. When considering the limitations in vision-based solutions, contactless signals [1] [9] [10] [11] [12] [4] [13] can be used. For example, WiFi signals [1], RFID signals [9], and acoustic signals [4] [13] were introduced for lip reading. However, WiFi-based solutions use a fixed WiFi device, RFID-based solutions attach tags around mouth, which may limit the application scenario of lip reading. To provide the service of lip reading anywhere anytime, lip reading based on acoustic signals emitted and collected by mobile devices was proposed in recent years.

However, the amount of research work on acoustic-based lip reading is limited. Specifically, Tan et al. [4] firstly introduced acoustic signals for lip reading and focused on recognizing basic mouth motions. After that, Gao et al. [5] utilized the micro-Doppler effect of acoustic signals and dual microphones of smartphone to recognize 45 words for lip reading. Zhang et al. [14] introduced inaudible acoustic signals modulated by GSM training sequence to sense lip movements, and then extracted channel impulse responses as features for a CNN network to classify 20 commands/words. Zhang et al. [15] introduced multi-frequency inaudible acoustic signals, and then designed a hierarchical convolutional neural network and a multi-task encoder-

Y. Yin, Z. Wang, K. Xia, L. Xie, and S. Lu are with the State Key Laboratory for Novel Software Technology, Nanjing University, China (e-mail: yafeng@nju.edu.cn, {mg1933063, MF21330092}@smail.nju.edu.cn, {lxie, sanglu}@nju.edu.cn). Lei Xie is the corresponding author.

Manuscript received August XX, 2022; revised XX XX, XXXX.

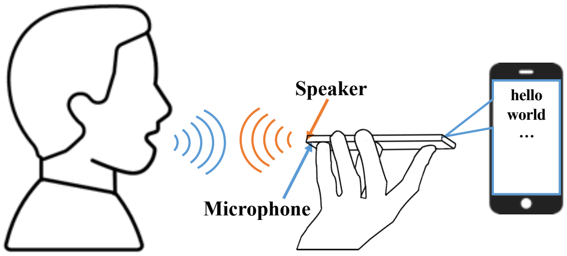


Fig. 1. The speaker of smartphone emits acoustic signals, while the microphone of smartphone collects the reflected signals for lip reading.

decoder network for word-level and sentence-level silent speech recognition, respectively. Overall, the existing work advanced the research of lip reading, but usually had the following limitations. *First*, there are few public lip reading datasets, thus making it difficult for performance comparison and may hinder the deeper research of lip reading on the basis of existing work. *Second*, when only using the weak signals collected from one microphone, it is still unknown how to efficiently perform lip reading with a large number of words and users, where more words/classes often mean worse/lower classification accuracy while more users often mean more differences of acoustic signals in the same class. *Third*, the existing solutions often depended on remote servers for data processing, while difficult to work on resource-limited mobile devices, thus may hinder the widespread use of the solutions.

Considering the above issues, in this paper, we focus on acoustic-based lip reading on mobile device. As shown in Fig. 1, the mobile device (e.g., smartphone) uses the built-in speaker to emit the acoustic signals, while using the built-in microphone to receive the reflected acoustic signals caused by lip movements (i.e., silent speech). The goal is to recognize the reflected acoustic signal caused by lip movement as a word on the mobile device with acceptable time latency. To achieve the above goal, there are four challenges to be addressed in this paper:

(1) Could we provide a public acoustic-based dataset to advance the research on lip reading? Considering the lack of public dataset in acoustic-based lip reading, we collect and release a large-scale lip-reading dataset LIPCMD, which is acquired from 20 users and consisted of 30000 acoustic-based recordings corresponding to 50 commonly used HCI commands (i.e., 50 classes/words). Besides, to further advance the following research on lip reading, we provide the benchmark evaluation on LIPCMD dataset, while using the traditional machine learning based solutions and recent deep learning based approaches.

(2) Only using one microphone to collect the weak acoustic signals of micro lip movements, how to achieve high performance of lip reading? Different from big motions like walking or moving hands, lip movements of silent speech are rather small, thus the reflected signals caused by lip movements are weak. Besides, only using one microphone to collect the reflected acoustic signals may further weaken the received signals. In addition, the information of one dimensional acoustic signal is rather limited. Therefore, it is challenging to use the weak acoustic signals for lip reading. To address this challenge, we first transform the acoustic signals to signal gradient matrix, to highlight the variation of acoustic signals caused by lip movements. Then, we introduce the attention map based self distillation mechanism, which generates attention maps among convolutional neural network (i.e., the backbone network of a deep learning based approach) and distills the importance of features from latter layers to former layers, to enhance the feature representation of weak acoustic signals for better lip reading.

(3) When moving to a large-scale acoustic-based dataset with a large number of words and users, how to achieve a good performance of lip reading? In a large-scale dataset, there are a large number of classes (i.e., types of words) and users. However, the larger number of classes often leads to a worse recognition/classification accuracy. Besides, more users often mean more differences in pronunciations or habits, thus introducing more confusion in recognizing words in the same class. Therefore, it is challenging to achieve a good classification performance for lip reading on a large dataset. To address this challenge, we introduce the probability distribution based self distillation mechanism, which adds intermediate classifiers among convolutional neural network to distill the probability distribution from latter classifier to former classifier, to highlight the class sensitive features of acoustic signals and improve the classification performance. Furthermore, to tolerate the user difference and make *LipReader* work in user-independent way, we also propose a fine-tuning strategy to make the system adapt to new users.

(4) Is it possible to achieve a light-weight and online lip-reading solution working on resource-limited mobile devices? It is known that the resource of mobile device is limited and the computationally-intensive deep learning based approaches can hardly work on mobile device. To provide a light-weight lip-reading solution for mobile device, we design the self-distillation modules, which are only used in training stage while being removed in testing stage, thus can reduce the parameters in proposed deep learning model. Besides, we further compress the trained model and get a small-size model which can be deployed on mobile device. To achieve online lip reading on mobile device with acceptable latency, we introduce the multi-thread scheme to perform Short-Time Fourier Transform (STFT) in parallel for reducing the recognition latency.

We make the following contributions in this paper:

- We collect and release a large-scale acoustic-based word-level dataset LIPCMD for lip reading. Besides, we also provide the benchmark evaluation and extensive analysis on LIPCMD dataset to advance the following research on lip reading.
- Considering the challenges from weak acoustic signals and complexity of a large-scale dataset, we propose a self distillation based approach *LipReader* which distills the attention map and probability distribution in the convolutional neural network itself, to enhance the feature representation and classification performance for lip reading. Besides, we also propose a fine-tuning strategy to make *LipReader* work in user-independent way.
- We implement *LipReader* on Android smartphone for online lip reading and provide a case study to show the usability of *LipReader*. Besides, we conduct extensive experiments on LIPCMD dataset as well

as under complex scenarios to evaluate the performance of *LipReader*. The experiment results show that *LipReader* can achieve a high recognition accuracy (i.e., 91.58%) on lip reading and outperforms the benchmark solutions and existing work.

2 Related Work

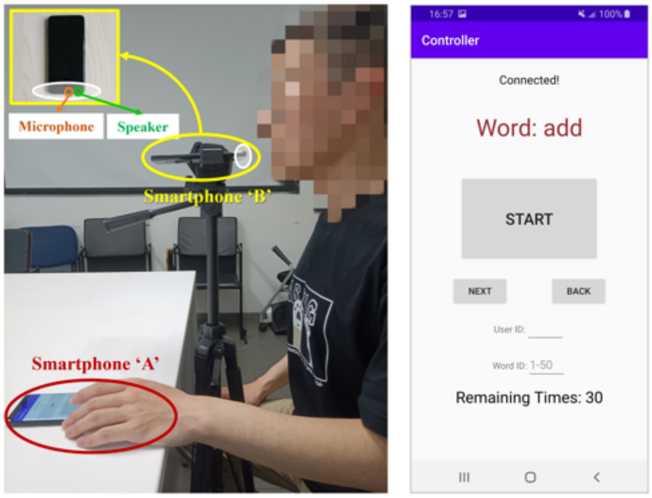
Lip reading aims to recognize the content of silent speech based on lip motions. To achieve this goal, computer vision, WiFi signals, acoustic signals etc were proposed for lip reading, where most of the existing work belonged to vision-based lip reading. In this section, we will introduce the vision-based lip reading which is most popular, the acoustic-based lip reading which is mostly related to this paper, and the public datasets on lip reading.

Vision-based Lip Reading: Vision-based lip reading uses the camera to capture images or videos of silent speech, and then utilizes the image processing techniques to recognize silent-speech content from images/videos. Vision-based lip reading has been studied for a long time. In the early stage, the research work tended to extract handcrafted features from images and then adopted classifiers for lip reading. For example, getting Discrete Cosine Transform (DCT) features [16], Histogram of Directional Gradients (HOG) features [17] from pixels, or using Snake (an active contour model) [18] to extract appearance-based features. Then, using Support Vector Machine (SVM) or Hidden Markov Model (HMM) to recognize the features as meaningful characters. Overall, the early work mainly focused on recognizing alphabet or simple phrase.

Recently, due to the development of deep learning, neural networks [19] [2] [3] [20] [21] [22] were proposed for lip reading. The neural network can be used as a replacement of classical classifier or automatic feature extractor [23]. For example, Wand et al. extracted handcrafted features and adopted Long Short-Term Memory (LSTM) network as a classifier for short phrase recognition [19]. However, in many cases, neural networks were used for automatic feature representation and adopted for lip reading in an end-to-end way. Specifically, to recognize video clips as words for lip reading, Chung et al. introduced Convolutional Neural Network (CNN) and other variants of CNN [2]. To recognize a video as a sentence for lip reading, CNN, LSTM and Connectionist Temporal Classification (CTC) loss were often adopted. For example, Assael et al. proposed a spatiotemporal neural network consisted of CNN and Gated Recurrent Unit (GRU) network, and then utilized CTC loss for sentence-level lip reading for the first time [3]. After that, attention mechanism was introduced to neural networks to further improve lip reading performance. In recent years, in addition to CTC loss, the encoder-decoder architecture was proposed for lip reading. Chung et al. proposed a neural network WLAS based on encoder-decoder architecture, and introduced a novel dual attention mechanism for sentence-level lip reading [20]. Due to the rich information of images/videos and the powerful deep learning based approaches, vision-based lip reading often achieves a good performance. However, the privacy issue and large computation overhead of image processing often hinder vision-based solutions to work on mobile devices, since their data processing was often done on remote server [22].

Acoustic-based Lip Reading: Acoustic-based lip reading was proposed in the last several years, and it was usually designed for mobile or wearable devices. At first, Tan et al. [4] proposed SilentTalk which introduced ultrasonic signals for lip reading on the smartphone by focusing on lip motion recognition. They designed Frequency Shift Detection Model to recognize lip motions corresponding to syllables, and then designed Continuous Lip Reading Model to generate words or short sentences from the recognized lip motions. After that, Gao et al. [5] proposed EchoWhisper, which introduced dual microphones of smartphone to enhance acoustic signals through beamforming, and then utilized the micro-Doppler effect of acoustic signals and a modified MobileNet for lip reading, i.e., recognizing 45 words from 5 subjects. Zhang et al. [14] introduced Endophasia, which used a mobile device to transmit and collect inaudible acoustic signals modulated by GSM training sequence to sense lip movements, and then extracted channel impulse responses as features for a CNN network to classify 20 commands/words. Zhang et al. [15] proposed SoundLip by using a smart device to send and receive multi-frequency inaudible acoustic signals, and then designed a hierarchical convolutional neural network for word-level silent speech recognition while designing a multi-task encoder-decoder network for sentence-level silent speech recognition. Until now, the research of acoustic-based lip reading often has the following limitations, i.e., the large-scale public lip reading dataset was rare, the performance of lip reading under a large number of words and users was unclear, the solutions depended on remote servers while difficult to work on resource-limited mobile devices.

Datasets on Lip Reading: To advance the following research on lip reading, many datasets on lip reading were proposed. However, most of the datasets belong to vision-based lip reading and include word-level, phrase-level and sentence-level datasets. LRW [2] is a typical and large-scale word-level dataset, where each video clip is intercepted from BBC television and corresponds to a word, it totally includes 500 types of words and more than 1000 speakers. MODALITY [24] is a dataset containing both words and phrases, which are commonly-used interaction commands (words and short phrases) in computers, and it totally has 163 types of commands. When moving to sentence-level dataset, GRID [25] is a single-syntax fixed-length sentence-level dataset, where each sentence consists of 6 words which are verbs, color, prepositions, letters, numbers and adverbs from left to right, and it totally has 51 types of words in all sentences. TCD-TIMIT [26] is also a sentence-level dataset, where the sentence is extracted from an audio-only speech recognition database TIMIT [27]. TCD-TIMIT consists of high-quality audios and videos from 62 speakers reading a total of 6913 phonetically-rich sentences. There are rich vision-based datasets on lip reading. However, on acoustic-based lip reading, there is only one public dataset [15], which includes 20 words and 70 sentences collected from 12 subjects. In this paper, we will provide a large-scale word-level dataset which is consisted of 50 words and collected from 20 subjects, aiming to further advance the following research on acoustic-based lip reading.



(a) (b)

Fig. 2. Data collection. (a) Smartphone 'A' is used as a controller which sends instructions, while smartphone 'B' is used as a recorder which emits acoustic signals and records reflected signals based on instructions. (b) The screenshot of controller, which contains the hint of word, the 'START' button, etc.

TABLE 1
Words in LIPCMD dataset

	1 syllable		2 syllables	3 syllables
Word	1. add	16.run	26.alarm	41.appointment
	2. back	17.save	27.begin	42.calendar
	3. check	18.send	28.browser	43.camera
	4. dial	19.set	29.copy	44.delete
	5. end	20.sound	30.edit	45.document
	6. go	21.start	31.export	46.message
	7. help	22.stop	32.forward	47.picture
	8. home	23.time	33.insert	48.reminder
	9. last	24.up	34.music	49.volume
	10.line	25.view	35.paste	50.yesterday
	11.may		36.pause	
	12.move		37.record	
	13.mute		38.select	
	14.nine		39.thursday	
	15.play		40.window	
Count	25		15	10

Note: Among the words, there are three pairs of easily confused words, i.e., {send, set}, {line, nine}, {play, may}.

3 Proposed Dataset

In this section, we propose a large-scale acoustic-based lip reading dataset called LIPCMD. Next, we will introduce the data collection process and describe the dataset in detail.

3.1 Data Collection

To capture the acoustic signals corresponding to lip motions, we use the built-in speaker and bottom microphone of smartphone to emit and receive the inaudible acoustic signals respectively, as shown in Fig. 2a. Specifically, the smartphone used for data collection is Samsung Galaxy S9. The acoustic signal emitted by speaker is 20kHz, while the sampling rate of microphone is set to 44.1kHz. As shown in Fig. 2a, the smartphone 'B' is used to record acoustic signals and is fixed on a tripod in front of the subject, where the distance between the bottom microphone and the subject's mouth ranges in $[2, 4]$cm. In the experiment, the subject silently speaks towards the bottom microphone of smartphone 'B' and we collect the acoustic signals in a quiet meeting room. In regard to the other smartphone 'A', it is used to control the data collection process and runs a controller app shown in Fig. 2b. For the collected acoustic signals, the recorder first removes a fixed-length segment corresponding to the 'beep' at the beginning and then records the remaining acoustic signals in a separate WAV file, which corresponds to one word silently spoken by one subject one time.

3.2 Details of LIPCMD Dataset

The proposed LIPCMD dataset contains 50 commonly-used words in daily life and human-computer interactions, e.g., 'help', 'play', 'pause', which are selected from the words in MODALITY dataset [24]. In regard to MODALITY dataset, it is consisted of 163 classes of words, including digits, months, dates, verbs and nouns used for controlling computer devices. For the 163 words, they can be classified into four groups based on number of syllables, i.e., 64 words with one syllable, 63 words with two syllables, 31 words with three syllables, 5 words with four syllables. When considering the difference between computers and smartphones in HCI and the ratio of words with different syllables, we select 50 words from the 163 words of MODALITY dataset. As shown in Table 1, LIPCMD contains 50 words, where 25 words have one syllable, 15 words have two syllables and 10 words have three syllables. The proportion of words with one syllable, two syllables, three syllables is 50%, 30%, 20%, respectively. Among the selected 50 words, there are three pairs of one-syllable words having similar pronunciations, i.e., 'send' and 'set', 'line' and 'nine', 'play' and 'may', which can be used to demonstrate the difficulty of lip reading on words with similar pronunciations.

To collect the acoustic signals of lip motions corresponding to the 50 words, we recruit 20 participants from our university. The 20 participants include 4 females and 16 males. All the participants are more than 18 years old and can speak English. For each participant, she/he repeatedly speaks each word silently 30 times, as shown in Fig. 2. Therefore, for each type of word, we can collect $30 \times 20 = 600$ samples. For all the 50 types of words in LIPCMD dataset, we can totally collect $600 \times 50 = 30000$ samples. In Fig. 3, we show the durations of samples corresponding to each type of word. We can find that the duration of silently-speaking a word usually lasts for 1.5 seconds to 2 seconds. For different words, there is no linear relationship between the durations and number of syllables. For the same word, the durations can also be different, due to the difference in users. To provide the fairness and convenience of performance evaluation on LIPCMD, we split the dataset into two parts, i.e., training set and testing set. Specifically, for the 30 samples corresponding to one type of word collected from one participant, we randomly select 80% (i.e., 24 samples) of them for training, while the remaining 20% of them are used for testing. In this way, for each type of word, there are 480 samples selected for training, while 120 samples are used for testing. On the whole dataset, there are

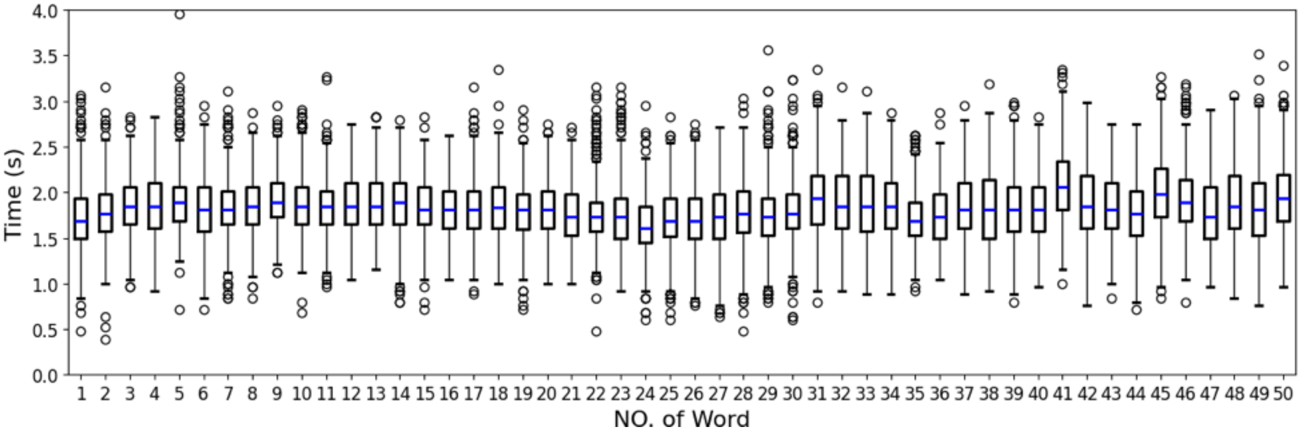


Fig. 3. The duration of each type of word in LIPCMD dataset

TABLE 2
Number of samples in LIPCMD dataset

	Each user		All users	
	Training	**Testing**	**Training**	**Testing**
1 syllable	600	150	12000	3000
2 syllables	360	90	7200	1800
3 syllables	240	60	4800	1200
Confused words	144	36	2880	720
All words	1200	300	24000	6000

24000 samples selected for training, while 6000 samples are used for testing. The statistical information of training and testing samples in LIPCMD is shown in Table 2.

4 Proposed Benchmark Evaluation

As mentioned before, in this paper, the goal of lip reading is to recognize the collected acoustic signal as a word, i.e., word classification. Thus the typical classification methods can be adopted. Therefore, in this section, we provide the benchmark evaluation on LIPCMD dataset through the classical machine learning based classification methods and recent deep learning based classification methods.

4.1 Classical Machine Learning based Methods

To realize word classification for lip reading, the classical machine learning based methods first need to extract features from acoustic signals and then adopt suitable classifiers for classification, as described below.

4.1.1 Feature Extraction

When using acoustic signals for human activity recognition like gesture recognition [28] and daily activity recognition [29], the signals were often transformed into frequency domain. Therefore, in the benchmark evaluation, we introduce Fast Fourier Transform (FFT) [30] to transform the acoustic signals from time domain to frequency domain, and calculate the logarithm of FFT coefficients [31] in the frequency range (i.e., 20kHz ± 40Hz) [4] caused by lip motions. Then, we extract the following features in the frequency range, i.e., mean, variance, Doppler shift, DFT coefficients [30] and PSD coefficients [32]. Finally, the 485 features are concatenated as a one-dimensional frequency feature vector.

TABLE 3
Accuracy of word classification using machine learning based methods

		DT	RF	LR	kNN	SVM
Freq	**1 syllable**	6.21%	**10.62%**	5.85%	8.10%	7.03%
	2 syllables	6.67%	**13.76%**	6.24%	2.22%	8.89%
	3 syllables	7.18%	**12.95%**	6.79%	3.08%	11.41%
	Confused	4.91%	**8.55%**	5.77%	7.48%	7.26%
	All words	6.54%	**12.03%**	6.15%	5.33%	8.46%
Time-Freq	**1 syllable**	14.31%	**34.36%**	23.13%	17.69%	18.62%
	2 syllables	14.62%	**36.50%**	26.75%	9.49%	20.85%
	3 syllables	15.26%	**39.74%**	28.46%	10.51%	23.97%
	Confused	12.61%	**34.83%**	25.21%	11.32%	20.94%
	All words	14.59%	**36.08%**	25.28%	13.79%	20.36%

Note: Freq: Frequency, DT: Decision Tree, RF: Random Forest, LR: Logistic Regression, kNN: k-Nearest Neighbor, SVM: Linear Support Vector Machine. The best performance in each row is shown in bold.

In addition the above frequency-related features, the 2D time-frequency map [33] [28] [34] [35] was also introduced for acoustic-based human activity recognition. Therefore, we perform a Short-Time Fourier Transform (STFT) [36] on the acoustic signals to get the time-frequency map. However, in time-frequency map, the strong signal in main frequency (20kHz) may overwhelm the weak Doppler shift of micro lip motion. Thus we introduce the signal gradient [37], i.e., the difference of data in time-frequency map at two consecutive time points, to extract the variation of signals caused by lip motions. Then, we get a signal gradient matrix. When considering the large number of features in signal gradient matrix, we further introduce Principal Component Analysis (PCA) to extract top 1500 main components from the matrix as features, which are concatenated as a one-dimensional time-frequency feature vector.

4.1.2 Word Classification

With the 1D feature vector (frequency or time-frequency feature vector), we use the following typical classifiers [38] for word classification, i.e., decision tree (DT), random forest (RF), logistic regression (LR), k-Nearest Neighbor

TABLE 4
Accuracy of word classification using deep learning based methods

	AlexNet	VGG16	ResNet18
1 syllable	80.33%	79.60%	**84.77%**
2 syllables	79.39%	81.89%	**86.61%**
3 syllables	81.25%	80.92	**87.83%**
Confused	77.36%	77.22%	**81.86%**
All words	80.23%	80.55%	**85.87%**

Note: The best performance in each setting/row is shown in bold.

(kNN), linear Support Vector Machine (SVM), to provide the benchmark evaluation. In Table 3, we show the recognition/classification accuracy of words on LIPCMD dataset, where '1 Syllable', '2 Syllables', '3 Syllables', 'Confused words', 'All words' respectively mean the words having one syllable, two syllables, three syllables, similar pronunciations, and all the words, as mentioned in Section 3.2. As shown in Table 3, when the features and classifier are fixed, as the number of syllables increases, the recognition accuracy usually increases. In regard to words having similar pronunciations, they are easy to be confused, thus the recognition accuracy is poor. Overall, when using the handcrafted features, the recognition accuracy is rather low. Specifically, when using frequency features, the recognition accuracy is usually smaller than 15%. When using time-frequency features, the recognition accuracy is usually smaller than 40%, whichever the classifier is used. It indicates that it is rather difficult to use traditional machine learning based methods to achieve a good performance for acoustic-based lip reading, more efficient approaches are expected.

4.2 Deep Learning based Methods

Different from the handcrafted features used in classical machine learning based methods, deep learning based methods can automatically extract features from input data and realize word classification in an end-to-end way. Specifically, in the benchmark evaluation, the input to the deep learning methods is the signal gradient matrix, as described in Section 4.1.1. While the adopted deep learning methods are typical convolutional networks i.e., AlexNet [39], VGG16 [40], and ResNet18 [41].

In Table 4, we show the word recognition accuracy using deep learning based methods on LIPCMD dataset. When comparing Table 4 and Table 3, we can find that deep learning based methods greatly improve the recognition accuracy, i.e., close to or larger than 80%. The reason may be that deep learning based methods can automatically extract features from input signal gradient matrix, thus getting more efficient feature representation for each acoustic-based word. According to Table 4, as the number of syllables increases, the recognition accuracy usually increases. In regard to the easily-confused words with similar pronunciations, the recognition accuracy decreases a little. Among the three deep learning based methods, ResNet18 achieves the best performance in all aspects. For all the words on LIPCMD dataset, the recognition accuracy using ResNet18 is 85.87%. It indicates that deep learning based method is possible to achieve a high performance for acoustic-based lip reading.

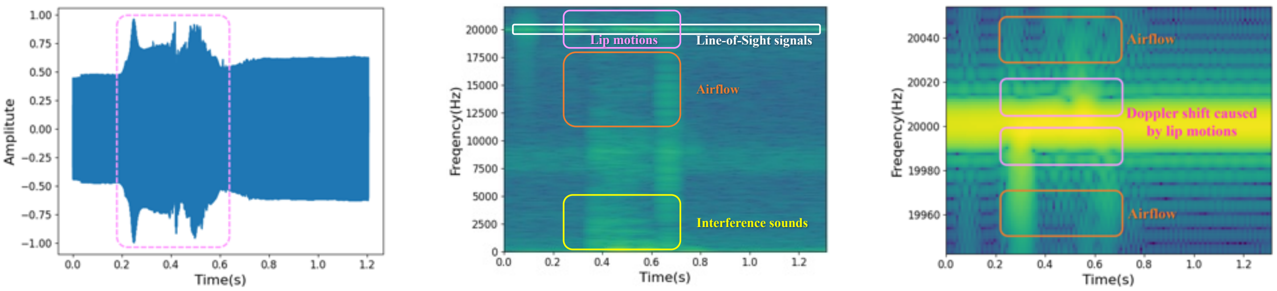
5 The Proposed Self Distillation based Approach for Lip Reading

As described in Section 4, when using the traditional machine learning based methods for lip reading, the recognition performance is rather poor, i.e., less than 40%. When using the typical deep learning based methods, the recognition accuracy is about 80%. It is still a challenging task to achieve a high performance for acoustic-based lip reading on a large-scale dataset. To further improve the recognition performance, we first provide some observations of acoustic-based lip motions to analyze the challenges in lip reading. Then, we provide a self distillation-based deep learning approach *LipReader* to improve the representation of acoustic signals for better lip reading.

5.1 Data Preprocessing and Observations

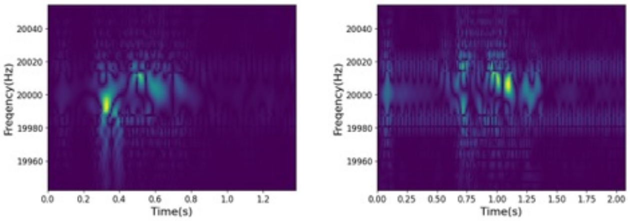
Observation 1. The acoustic signals caused by silent speech are very weak and easily buried by the emitted signals from speaker and other interference signals. Take the word 'add' as an example, in Fig. 4, we show the acoustic signals in time domain and frequency domain. Specifically, Fig. 4a shows the collected acoustic signals for 'add' in time domain, where the pink rectangle corresponds to the duration that the user is silently speaking the word. Compared with the background signals out of the pink rectangle, the variation of acoustic signals caused by silent speech is weak. It is difficult to directly use time-domain signals for lip reading. Therefore, we further introduce short-time Fourier transform [36] to transform the acoustic signals of 'add' in Fig. 4a to time-frequency domain, as shown in Fig. 4b. According to Fig. 4b, when the user silently speaks a word, the collected signals mainly consist of the Line-Of-Sight (LOS) signals from speaker, the lip related signals caused by lip movements, the airflow related signals caused by silent pronunciation, and the low-frequency audible sounds caused by interferences, as shown in Fig. 4b. When considering that the Doppler shift caused by silent speech ranges in [-20, 40]Hz [4], we use the frequency window [f_0-40, f_0+40]Hz to get the silent speech related signals, while filtering the other interference signals, as shown in Fig. 4c. Here, f_0=20kHz means the frequency of emitted signals (main frequency for short) from speaker. According to Fig. 4c, the signals caused by silent speech, i.e., lip motions and airflow related signals, are rather weak, when compared with the main frequency. This is one reason why the recognition accuracy of traditional machine learning based methods is rather low.

Observation 2. Different people show non-negligible difference in lip motions and pronunciations for the same word. According to observation 1, the signals caused by silent speech are rather weak, when compared with the emitted signals from speaker. Therefore, we introduce signal gradient [37] to get the difference of signals at two consecutive time points from the time-frequency map, i.e., $d_t = s_t - s_{t-1}$, to remove the main frequency and extract the signals related to silent speech. Here, s_t and s_{t-1} mean the signal at time t and $t-1$ respectively, while d_t means the difference (i.e., gradient) between s_t and s_{t-1}. In Fig. 5 and Fig. 6, we show the signal gradient matrices corresponding to the word 'add' and 'begin' spoken by different users, respectively. When comparing Fig. 5a and Fig. 6a, we can find that the acoustic



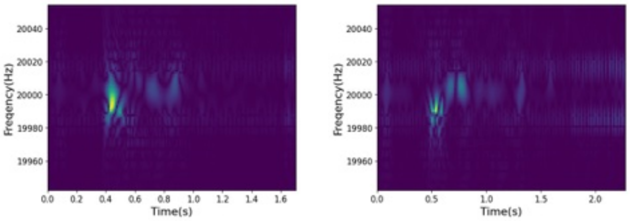
(a) 'add' in time domain (b) 'add' in time-frequency domain (c) 'add' in a window

Fig. 4. Acoustic signals of 'add' in time domain, time-frequency domain when silently speaking the word.



(a) Subject 1 (b) Subject 2

Fig. 5. Signal gradient matrices of silently speaking the word 'add' by two subjects.



(a) Subject 1 (b) Subject 2

Fig. 6. Signal gradient matrices of silently speaking the word 'begin' by two subjects.

signals of different words are different, which indicates that it is possible to use signal gradient matrix to distinguish different words for lip reading. However, when comparing Fig. 5a and Fig. 5b (or Fig. 6a and Fig. 6b), we can find that different people lead to non-negligible difference of signals for the same word. The difference is mainly caused by different user habits, including lip motions, pronunciation, breathing pattern in speaking etc, and it greatly increases the difficulty for word classification (i.e., lip reading). Therefore, the existing research work on lip reading tended to perform the evaluation in user-dependent way.

5.2 Overview of Self Distillation Mechanisms

As described in benchmark evaluation, the traditional machine learning based methods can hardly achieve a good performance for lip reading, due to the poor handcrafted features. While for deep learning based methods, they can automatically extract features and improve the recognition accuracy, especially for ResNet18. However, considering the weak signals caused by silent speech, we further introduce the self distillation technology [42] [43], which is used to teach the neural network to learn from itself, then improve feature representation and classification performance. Specifically, we propose *LipReader* by selecting ResNet18 as the backbone network and introducing two kinds of self distillation mechanisms among ResBlocks of ResNet18, i.e., probability distribution based self distillation and attention map based self distillation, to improve the feature representation and classification performance for lip reading.

5.3 Self Distillation Based on Probability Distribution

In the original ResNet18 [41], there are 18 layers, which mean a convolutional layer, four ResBlocks and a fully-connected layer. After the fully-connected layer, a softmax classifier is used for classification. In regard to the ResBlock, it is consisted of four 3 × 3 convolutional layers, two skip connections, and two addition operations. In *LipReader*, the backbone network is ResNet18, while the input is signal gradient matrix with the size of 68 × 300, as shown in Fig. 7. To further enhance the classification performance, we introduce the Probability Distribution (PD for short) based self distillation mechanism. Specifically, in addition to the exiting classifier after the fourth ResBlock, we also insert the intermediate classifiers after the first, second, and third ResBlock, as the bottom part shown in Fig. 7. Here, the Bottleneck is consisted of 1 × 1, 3 × 3, 1 × 1 convolutional layers, one skip connection and one addition operation, and it is used for changing the dimension of features. In regard to the newly inserted intermediate classifiers, they are only adopted in training phase while being removed in testing phase. In the training phase, the original/last classifier distills the knowledge into the previous three classifiers, to guide the previous three ResBlocks to extract more efficient features for classification. For convenience, the last ResBlock and classifier are treated as a teacher model, while the previous three ResBlocks and three classifiers are treated as a student model.

To distill the knowledge from the teacher model to student model, we design the following Kullback-Leibler (KL) divergence loss. For convenience, we use C to represent the number of classes (i.e., types of words), while using $X = \{x_n\}_{n=1}^{N}$ to represent the N samples in C classes. Besides, we use M to represent the number of classifiers (i.e., $M = 4$), while using Θ_m, $m \in [1, M]$ to represent the ith classifier. Then, when given the sample x_n, we can describe the output probability $q_c^m(x_n)$ after the mth classifier for the cth class with Eq. (1), where $z_c^m(x_n)$ means the output logit of the sample x_n after the mth classifier for the cth class. Here, T represents the temperature [44] in the distillation and it is set to 3 by default.

$$q_c^m(x_n) = \frac{exp(\frac{z_c^m(x_n)}{T})}{\sum_{j=1}^{C} exp(\frac{z_j^m(x_n)}{T})} \quad (1)$$

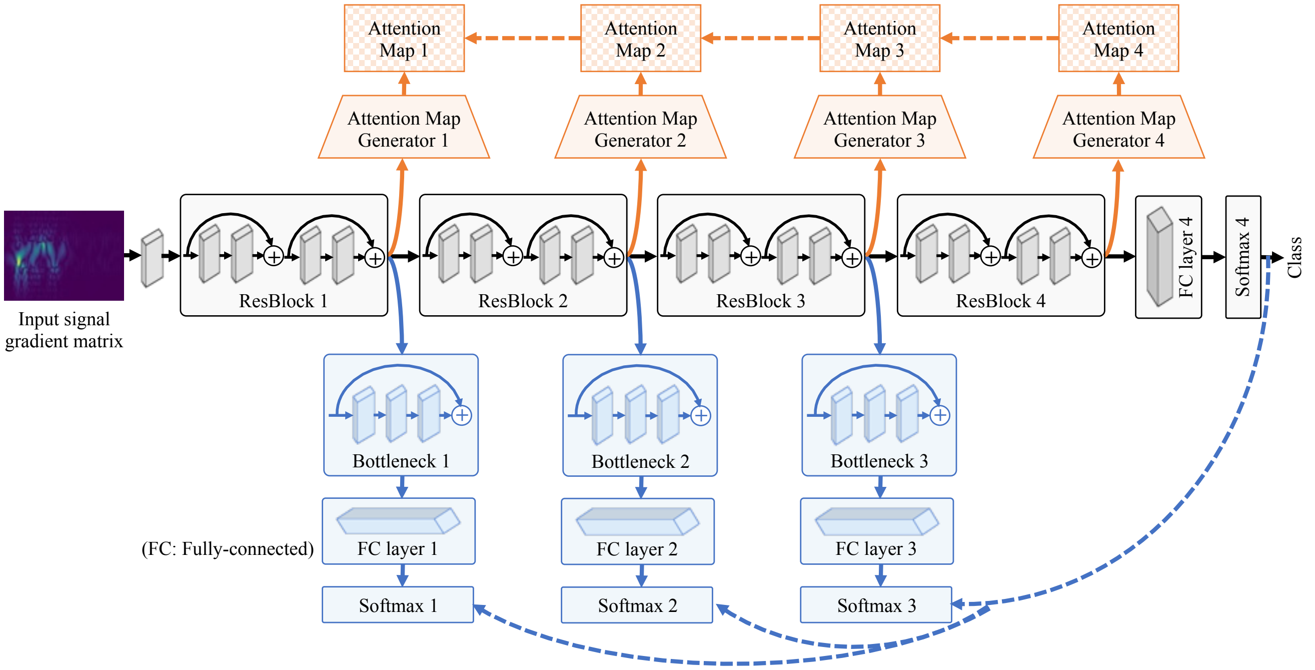


Fig. 7. A self distillation based approach, which includes the backbone network ResNet18, the probability distribution based self distillation and the attention map based self distillation.

With the probability $q_c^m(x_n)$, we can calculate the KL divergence loss $\mathcal{L}_p$ between the teacher model and the student model with Eq. (2). Specifically, $q^m(x_n)$, $m \in [1, M-1]$ means the probability distribution from the previous classifier (i.e., student model), while $q^M(x_n)$ means the probability distribution from the last classifier (i.e., teacher model). When comparing $q^m(x_n)$ and $q^M(x_n)$, i.e., calculating the KL loss between them, we can distill the knowledge from teacher model to student model.

$$\mathcal{L}_p(q^m, q^M) = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} q_c^M(x_n) \cdot log \frac{q_c^M(x_n)}{q_c^m(x_n)} \qquad (2)$$

Since the teacher model has the final softmax classifier which is used for classification, it has the ability to distinguish good (i.e., class-sensitive) features and poor (i.e., class-insensitive) features based on classification results. Therefore, when the teacher model distills the probability distribution in classification to the student model, the student model will learn how to extract good features to improve classification performance.

To verify whether the probability distribution based self distillation mechanism can improve the classification performance, we provide the word recognition accuracy on LIPCMD dataset, while using and NOT using self distillation. In Fig. 8, we show the classification performance of each classifier, while using the samples from 5 randomly-selected users. Usually, whether using or NOT using self distillation mechanism, the deeper classifier can achieve a better performance. However, when using the self distillation mechanism (i.e., the yellow bar), each classifier can further improve the lip reading performance, especially the shallower classifiers. It indicates that each ResBlock has learned from the teacher model to distinguish useful features and unuseful features, then focusing on the useful/good features contributing to higher classification performance. Therefore, the proposed probability distribution based self distillation mechanism can efficiently highlight the class-sensitive features, and improves classification performance.

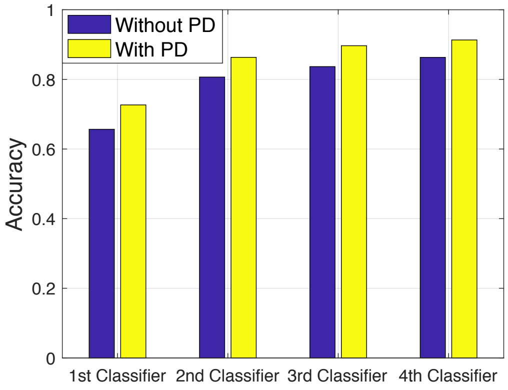


Fig. 8. Classification accuracy of each classifier when using and NOT using probability distribution (PD) based self distillation mechanism

5.4 Self Distillation Based on Attention Map

To further improve the feature representation, we propose the Attention Map (AM for short) based self distillation mechanism, where the attention map is used to describe the importance of extracted features. Specifically, by adopting ResNet18 as the backbone network, we generate an intermediate attention map after the first, second, third and fourth ResBlock, respectively, as the top part shown in Fig. 7. The newly generated intermediate attention maps are only adopted in training phase while removed in testing phase. In the training phase, the latter attention map distills the knowledge (i.e., importance of different features) to the previous and adjacent attention map, to guide the previous

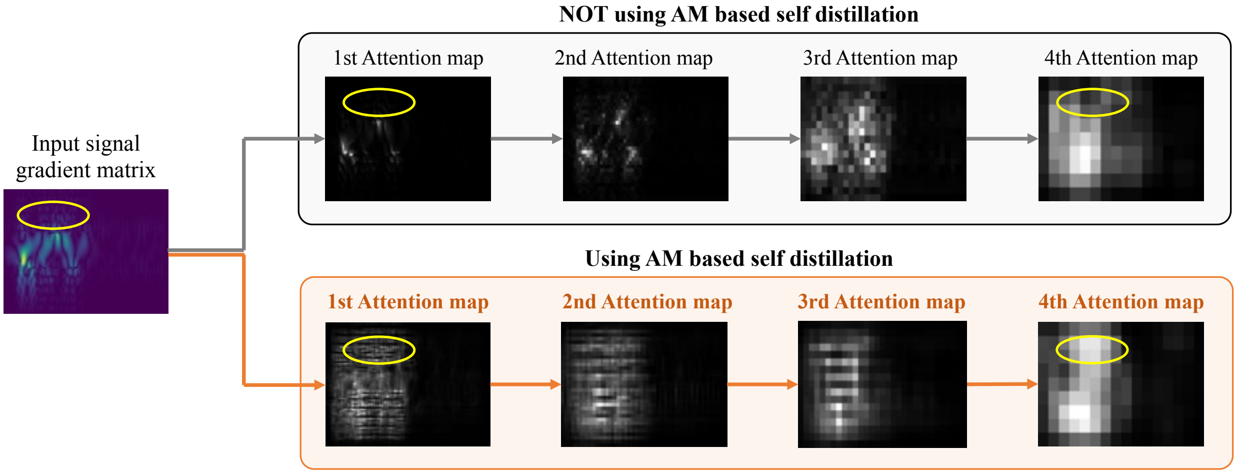


Fig. 9. Attention map after each ResBlock when using and NOT using attention map based self distillation mechanism.

ResBlock to focus on the important/efficient features. For convenience, the latter ResBlock and attention map are treated as a teacher model, while the previous and adjacent ResBlock and attention map are treated as a student model.

To distill the knowledge from teacher model to student model, the key is to get the attention map. In *LipReader*, we use the output from each ResBlock to generate the attention map. Specifically, we use O_i, $i \in [1,4]$ to represent the output (i.e., intermediate feature maps) of the ith ResBlock. Here, $O_i \in \mathbb{R}^{h_i \times w_i \times l_i}$, where h_i, w_i and l_i mean the height, the width and the number of channels (i.e., number of feature maps) of O_i, respectively. For convenience, we use $O_i(x,y,z)$ to represent the element in the xth row and the yth column of the zth feature map, where $x \in [1,h_i]$, $y \in [1,w_i]$ and $z \in [1,l_i]$. Then, we use the activation-based mapping function [45] to generate the attention map A_i from O_i with Eq. (3). Here, $A_i \in \mathbb{R}^{h_i \times w_i}$, while $A_i(x,y)$ means the element in the xth row and the yth column, $x \in [1,h_i]$, $y \in [1,w_i]$.

$$A_i(x,y) = \sum_{z=1}^{l_i} |O_i(x,y,z)|^2 \tag{3}$$

With the attention map A_i, $i \in [1,4]$ after the ith ResBlock, we utilize the block-wise distillation loss $\mathcal{L}_a(A_i, A_{i+1})$ to distill the knowledge from the latter attention map A_{i+1} to the previous and adjacent attention map A_i with Eq. (4). Here, the function $\mathcal{P}(\cdot)$ means the upsampling [46] operation, which is used to align the attention maps with different sizes. The function $\mathcal{L}_2(\cdot,\cdot)$ is used to calculate L2 loss. Specifically, when representing A_i and $\mathcal{P}(A_{i+1})$ with U and V, we can calculate $\mathcal{L}_2(U,V) = \sum_{x=1}^{h}\sum_{y=1}^{w}\frac{(U(x,y)-V(x,y))^2}{h \cdot w}$, where $U(x,y)$, $V(x,y)$ mean the element in the xth row and the yth column of U and V, respectively.

$$\mathcal{L}_a(A_i, A_{i+1}) = \mathcal{L}_2(A_i, \mathcal{P}(A_{i+1})) \tag{4}$$

Usually, the deeper layers of neural network can extract better features, while the shallower layers can hardly extract good features. Therefore, by distilling the latter attention map to the previous attention map, the previous layers can learn the importance of different features and focus on extracting the important and meaningful features.

To verify whether attention map based self distillation mechanism can improve the feature representation, we respectively show the attention maps after the first, second, third, fourth ResBlock, while using and NOT using attention map (AM) based self distillation mechanism during training. As shown in Fig. 9, the leftmost map means the input signal gradient matrix corresponding to word ‘add’, while the gray-scale images represent the attention maps outputted after each ResBlock. Specifically, in gray-scale images, the white region corresponds to the features paid more attention (i.e., important features) by the ResBlock. When comparing the gray-scale images in the top row and that in the bottom row, especially for the regions marked with yellow ellipses, we can find that when using self distillation, the extracted features in the shallower layers and deeper layers have better consistency. That is to say, using attention map based self distillation can help the shallower layers learn the importance of features from the latter layers and quickly focus on the important features, thus improve feature representation.

5.5 Model Training

To apply the probability distribution based self distillation and attention map based self distillation on ResNet18, we design three kinds of losses. Specifically, the first loss $\mathcal{L}_c$ is designed for the last classifier and used for word classification. For convenience, we use $y_{n,c}$ to represent the label (i.e., true class) of sample x_n, where $n \in [1,N]$ and $c \in [1,C]$. If the label of x_n belongs to the cth class, then $y_{n,c} = 1$. Otherwise, $y_{n,c} = 0$. After that, we can calculate the cross entropy loss $\mathcal{L}_c$ with Eq. (5), where $q_c^M(x_n)$ means the probability that the sample x_n is classified as the cth class by the Mth/last classifier.

$$\mathcal{L}_c = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{n,c} \cdot log(q_c^M(x_n)) \tag{5}$$

The second loss $\mathcal{L}_p$ is designed for probability distribution based self distillation, aiming to improve the classification performance of intermediate classifiers and highlight the

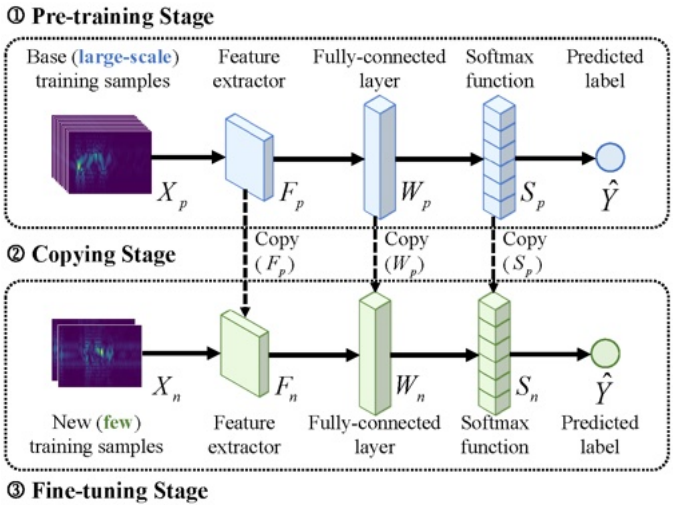


Fig. 10. The fine-tuning strategy used for domain adaptation

class-sensitive features. $\mathcal{L}_p$ a Kullback-Leibler divergence loss and computed with Eq. (6), where $\mathcal{L}_p(q^m, q^M)$ is calculated with Eq. (2).

$$\mathcal{L}_p = \sum_{m=1}^{M-1} \mathcal{L}_p(q^m, q^M) \tag{6}$$

The third loss $\mathcal{L}_a$ is designed for attention map based self distillation, aiming to make the shallow layers learn the importance of features and focus on important features, thus improving the feature representation of each word. $\mathcal{L}_a$ is computed with Eq. (7), where $\mathcal{L}_a(A_i, A_{i+1})$ is calculated with Eq. (4).

$$\mathcal{L}_a = \sum_{i=1}^{M-1} \mathcal{L}_a(A_i, A_{i+1}) \tag{7}$$

Finally, we combine the three losses and define the overall loss $\mathcal{L}$ with Eq. (8) for model training, where α and λ are two hyper-parameters to balance the three kinds of losses. In this paper, we set $\alpha = 0.1$ and $\lambda = 10^{-6}$ by default.

$$\mathcal{L} = (1 - \alpha) \cdot \mathcal{L}_c + \alpha \cdot \mathcal{L}_p + \lambda \cdot \mathcal{L}_a \tag{8}$$

6 Fine-Tuning for Domain Adaptation

According to Observation 2, the non-negligible difference among users may confuse the recognition of same word, thus the existing work tended to work in user-dependent way while could hardly work in user-independent way. To address this issue, we propose a fine-tuning strategy to make *LipReader* adapt to different domains, e.g., different users. As shown in Fig. 10, the fine-tuning strategy has three stages, i.e., pre-training stage, copying stage, and fine-tuning stage. Firstly, we utilize large-scale training samples from one or more domains to train a base model, which has a good ability of feature extraction and word classification for seen/known domains while may not work well for unseen/unknown domains. Secondly, we copy the components (i.e., feature extractor F_p, fully-connected layer W_p, softmax function S_p) of the above pre-trained model to generate a duplicated model, which is used to accelerate the convergence speed of the following model fine-tuning. Thirdly, we only use a few training samples from the new/unseen domain to fine tune the duplicated model and get a new model, which has the same structure but different parameters F_n, W_n, S_n compared with the pre-trained model. Here, the fine-tuned new model can adapt to new/unseen users, thus can work in user-independent way. It is worth noting that the fine-tuning strategy does not need to be adopted everytime. In fact, when the model needs to work in different domains, e.g., for new users, the fine-tuning strategy will be adopted. Otherwise, the strategy will not be adopted. Besides, as a domain adaptation strategy, the fine-tuning strategy can not only be used for user adaptation, but also for device adaptation, environment adaptation, placement adaptation, and so on.

7 Performance Evaluation

To evaluate the performance of proposed solution *LipReader* on lip reading, we introduce the experiment setting and conduct extensive experiments on LIPCMD dataset. Firstly, we show the average word recognition/classification accuracy, and analyze the performance from the aspect of words and users. Secondly, we perform the ablation study to test the efficiency of designed self distillation components of *LipReader*, i.e., probability distribution based self distillation and attention map based self distillation. Thirdly, we evaluate how the training size, device placement, and complex scenarios affect the performance of lip reading. Finally, we compare our proposed *LipReader* with the existing lip-reading methods. It is worth noting that unless otherwise specified, *LipReader* works in user-dependent way by default.

7.1 Experiment Setting

Before conducting the following experiments, we first introduce the detailed settings of *LipReader*, including the input data, model parameters, model training and model implementation. For the input signal gradient matrix to neural network, the height is set to 68, while the width is set to 300 by default. If the duration of a sample is short, i.e., the width of signal gradient matrix is smaller than 300, we will pad the matrix with zeros in the left side and right side symmetrically until the width achieves 300. Besides, to get enough input data (i.e., signal gradient matrix) for model training, we introduce data augmentation. Specifically, we use time-dimensional cropping, data masking and size scaling strategies for data augmentation. Here, time-dimensional cropping strategy randomly crops the data in left (or right) columns, and then pads the matrix with zeros in right (or left) side. Data masking strategy randomly sets the data in some rows or columns to zero. Size scaling strategy treats the input matrix as an image, and then stretches or compresses the matrix/image in horizontal or vertical direction around the center of matrix/image. In this way, we can increase the training size to 30 times of the original one. For the model parameters, the backbone network is ResNet18. Considering that the channel of input signal gradient matrix is 1, we modify the channel of the first layer in ResNet18 to be 1, while keeping the others same with ResNet18. For model training, we adopt the

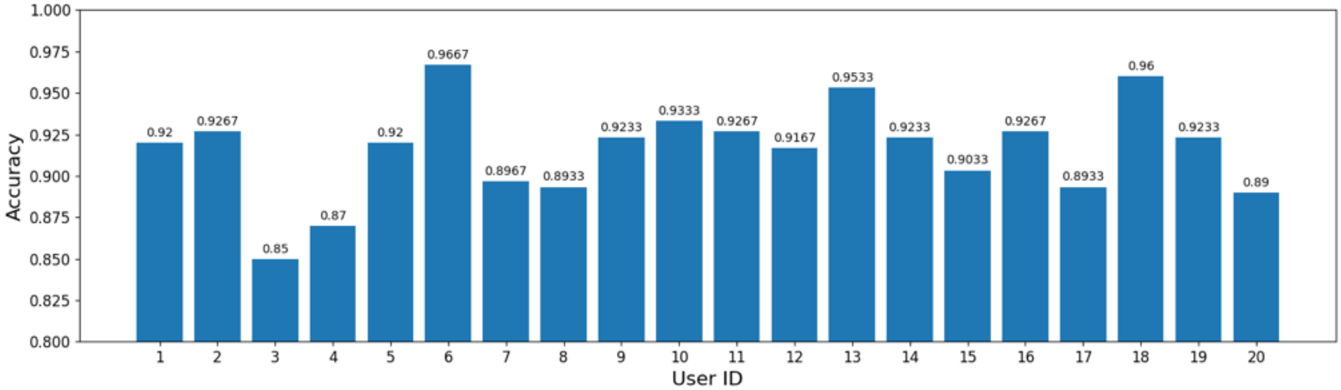


Fig. 11. The accuracy of *LipReader* vs. different users

TABLE 5
Recognition accuracy of words in different categories

	LipReader
1 syllable	90.97%
2 syllables	91.67%
3 syllables	93.00%
Confused words	88.61%
All words	91.58%

Adam optimizer [47]. For model implementation, *LipReader* is implemented with PyTorch 1.5.1 and trained for 200 epochs on two NVIDIA Tesla V100 GPUs.

7.2 Accuracy on Lip Reading

In this subsection, we first evaluate the performance of *LipReader* on lip reading in terms of word recognition/classification accuracy. To keep the consistency with benchmark evaluation, we respectively show the recognition accuracy of words with one syllable, two syllables and three syllables, as well as confused words and all words. As shown in Table 5, as the number of syllables increases, the recognition accuracy increases. For the three-syllable words, our *LipReader* can even achieve the accuracy of 93.00%. When moving to the confused words, which have similar pronunciation, the recognition accuracy drops to 88.61%. However, on average, the recognition accuracy of all words on LIPCMD dataset can achieve 91.58%. When compared with the benchmark evaluation, which is shown in Table 3 and Table 4, our proposed *LipReader* can apparently improve the performance of lip reading. It indicates the efficiency of *LipReader* on acoustic-based lip reading.

7.2.1 Analysis about User Difference

When considering the user difference, we also analyze the word recognition accuracy for each user. Specifically, for each user, we average the recognition accuracy of all words. As shown in Fig. 11, for most of users, the word recognition accuracy is close to or larger than 90%. Take the sixth user as an example, the recognition accuracy can achieve 96.67%. While for some user, e.g., the third user, the recognition accuracy is a little low, i.e., 85%. The difference is mainly caused by the different speech habits of users, e.g., different amplitudes of lip motions, different pronunciation manners, different speaking speeds. Usually, when the user speaks with a large lip motion, clear pronunciation and normal speed, *LipReader* can achieve a good recognition accuracy. When the user tends to speak with a very small lip motion or a very fast speed, the word recognition accuracy will decrease. However, on average, we can achieve a good recognition accuracy on 50 types of words, i.e., 91.58%, even the users have different habits in speech.

7.2.2 Analysis about Word Difference

When considering the difference in words, we further analyze the recognition accuracy for each type of word. As shown in Fig. 12, we show the confusion matrix of word recognition accuracy on LIPCMD dataset, which consists of 50 types of words. In Fig. 12, the element in the ith row and the jth column means the probability that the ith word is recognized as the jth word in Table 1. Therefore, the diagonal elements correspond to the recognition accuracy of each word, i.e., the probability that the word is correctly recognized, while the other/off-diagonal elements correspond to the recognition error rate, i.e., the probability that the word is wrongly recognized. According to Fig. 12, our *LipReader* can achieve a good performance on acoustic-based lip reading. For each word, the recognition accuracy is larger than 84%. For some words, e.g., the 22th word 'stop' and the 24th word 'up', the recognition accuracy can achieve 98%. While for some word, e.g., the 18th word 'send', the recognition accuracy is a little low, e.g., 84%. The difference on recognition accuracy mainly comes from the number of syllables and manner of pronunciation in a word, and the similar pronunciations among words. However, on average, the word recognition accuracy of our proposed *LipReader* can achieve 91.58%, and it is obviously superior than the solutions in benchmark evaluation where the best recognition accuracy is 85.87%.

7.3 Ablation Study

To evaluate the contributions of designed self distillation mechanisms in *LipReader*, we perform the following ablation study. Specifically, we respectively remove the Probability Distribution (PD for short) based self distillation component, Attention Map (AM for short) based self distillation component, and both of them from *LipReader*, and then test the recognition accuracy on LIPCMD dataset. According to

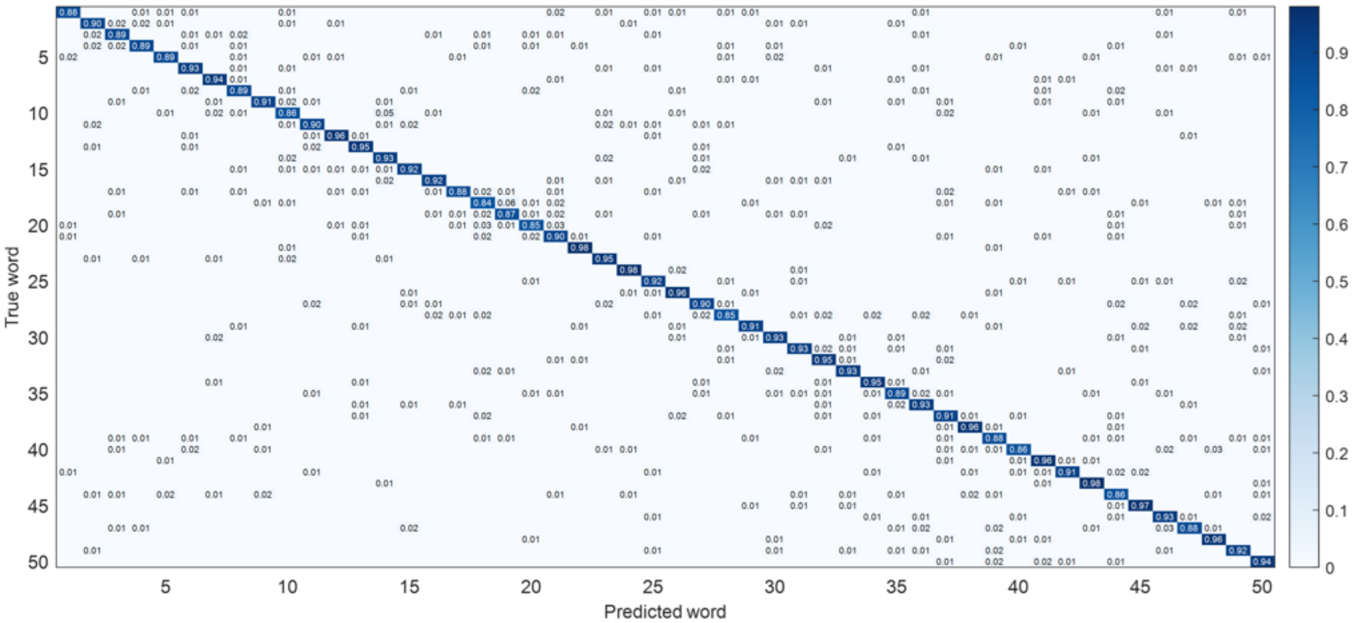


Fig. 12. Confusion matrix of word recognition accuracy on LIPCMD dataset, where the NO. of word can be found in Table 1

TABLE 6
Ablation study on self distillation mechanisms

LipReader	Without PD	Without AM	Without PD+AM
91.58%	89.05%	87.35%	85.87%

Note: PD: Probability Distribution based self distillation, PA: Attention Map based self distillation

Table 6, when removing the probability distribution based self distillation mechanism (i.e., 'Without PD' in the second column), attention map based self distillation mechanism (i.e., 'Without AM' in the third column), the recognition accuracy decreases by 2.53%, 4.23%, respectively. When removing both of the two self distillation mechanisms (i.e., 'Without PD+AM' in the fourth column), the recognition accuracy decreases by 5.71%. It indicates that each of the above self distillation mechanisms contributes to a higher recognition performance. Overall, by making full use of the intermediate information (i.e., probability distribution, attention map) in neural network, self distillation mechanisms can make the neural network improve feature representation of silent speech and achieve a good performance of lip reading.

7.4 Effect of Training Size

To evaluate how the training size affects the word recognition performance. We change the number of training samples of each word. In LIPCMD dataset, for each person, there are 30 samples for each word, where 24 samples (i.e., 80%) are used for training while the remaining 6 samples (i.e., 20%) are used for testing. This is a default setting. However, in the following experiment, for each person, we change the number of training samples of each word from 9 (i.e., 30%) to 24 (i.e., 80%), where the step length is 3 (i.e., 10%). When the number of training samples is fixed, we average the recognition accuracy of all words. As shown in Fig. 13, when the training size is small (e.g., 9 training samples), the recognition accuracy is rather low, i.e., 70.83%. This is because training with a few samples can lead to the problem of overfitting and hardly get a good neural model for lip reading. As the training size increases, the word recognition accuracy increases. When the number of training samples achieves 24, our *LipReader* can achieve a good recognition accuracy of 91.58%. Therefore, while considering the recognition performance and enough test samples on LIPCMD dataset, we randomly select 80% of samples for training and use the remaining 20% of samples for testing.

7.5 Effect of Device Placement

Effect of distance between mouth and microphone: In this experiment, we evaluate *LipReader* by varying the distance between user's mouth and smartphone's bottom microphone. Firstly, given a fixed distance (i.e., 2cm), we invite one volunteer to collect 24 samples for each type of word, and get 24 × 50 = 1200 samples for model training. Secondly, we invite the same volunteer to collect 6 samples for each type of word at different distances, which range in $[1, 6]$cm and change by 1cm, and then get $6 \times 50 = 300$ test samples under each distance. After that, we use the previous trained model to evaluate lip reading performance on test samples at each distance. As the blue line shown in Fig. 14, when the testing samples and training samples are collected at the same distance (i.e., 2cm), the recognition accuracy achieves the highest (i.e., 92.33%). Otherwise, the recognition accuracy decreases. When we train the model with samples collected at other distances, i.e., 3cm or 4cm, the above phenomenon still exists, as the black line and red line shown in Fig. 14. It indicates that keeping the same distance in training and testing can guarantee a good performance, while testing with unseen signals from different distances may decrease the performance.

To make *LipReader* adapt to new/unseen distances, we can adopt the fine-tuning strategy in Section 6. Specifically, at a new distance, we invite the same volunteer to additionally collect 6 samples for each type of word. Then, we

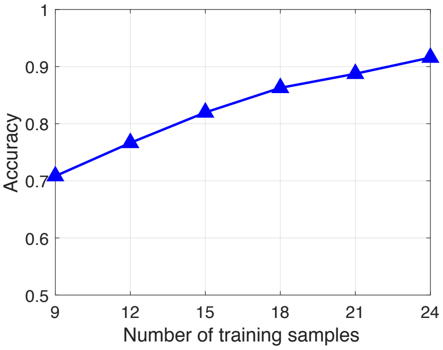


Fig. 13. Word recognition accuracy vs. training size

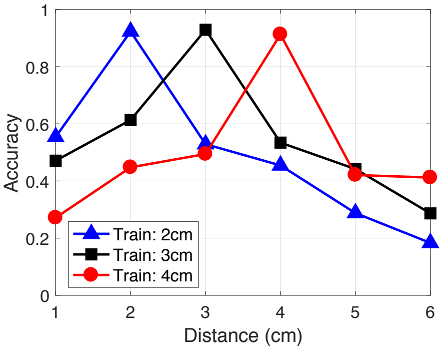


Fig. 14. Word recognition accuracy vs. distance between user and microphone

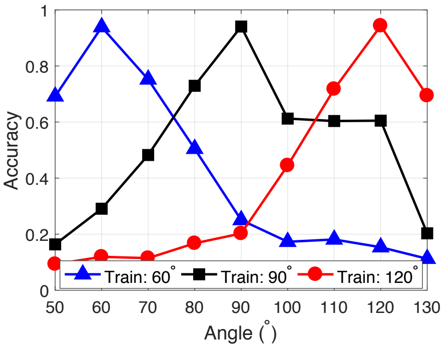


Fig. 15. Word recognition accuracy vs. angle of smartphone

TABLE 7
Recognition accuracy of *LipReader* at different distances

Train: 3cm	Test: 5cm	
	Without fine-tuning	With fine-tuning
	44.33%	**91.33%**

TABLE 8
Recognition accuracy of *LipReader* at different angles

Train: 90°	Test: 120°	
	Without fine-tuning	With fine-tuning
	60.67%	**92.67%**

use these samples to fine tune the pre-trained model, and use the fine-tuned model for testing. As shown in Table 7, after fine tuning, the word recognition accuracy at the new distance (i.e., 5cm) increases from 44.33% to 91.33%. It indicates that *LipReader* can adapt to new distances with fine tuning and achieve a good performance for lip reading.

Effect of angle between face's plane and smartphone's surface: In this experiment, we evaluate *LipReader* by varying the angle between face's plane and smartphone's surface. Firstly, given a fixed angle (i.e., 90°), we invite one volunteer to collect 24 samples for each type of word, and get $24 \times 50 = 1200$ samples for model training. Secondly, we invite the same volunteer to collect 6 samples for each type of word at different angles, which range in $[50, 130]^{\circ}$ and change by 10°, and then get $6 \times 50 = 300$ test samples at each angle. After that, we use the previous trained model to evaluate *LipReader* on test samples from different angles. As the blue line shown in Fig. 15, when the testing samples and training samples are collected at the same angle (i.e., 90°), the recognition accuracy achieves the highest (i.e., 95%). Otherwise, the recognition accuracy decreases. When we train the model with samples collected at other angles, i.e., 60° or 120°, the above phenomenon still exists, as the black line and red line shown in Fig. 15. It indicates that keeping the same angle in training and testing can guarantee a good performance, while testing with unseen signals from different angles may decrease the performance.

To make *LipReader* adapt to new/unseen angles, we can adopt the fine-tuning strategy in Section 6. Specifically, at a new angle, we invite the same volunteer to additionally collect 6 samples for each type of word. Then, we use these samples to fine tune the pre-trained model, and use the fine-tuned model for testing. As shown in Table 8, after fine tuning, the word recognition accuracy at the new angle (i.e., 120°) increases from 60.67% to 92.67%. It indicates that *LipReader* can adapt to new angles with fine tuning and achieve a good performance for lip reading.

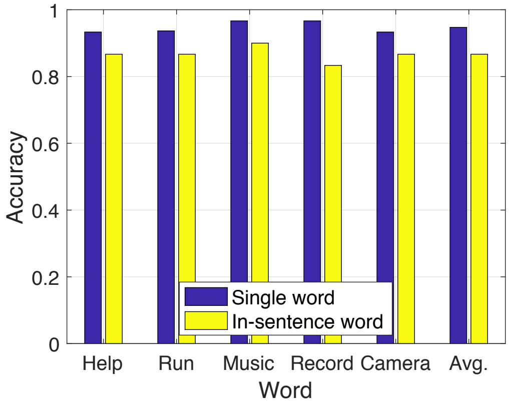


Fig. 16. Recognition accuracy of words collected in different ways

7.6 Effect of Complex Scenarios

Effect of word source: In the proposed dataset, each word is spoken separately. However, in fact, a word is often spoken in a sentence. To test the robustness of *LipReader* in real scenarios, we invite one volunteer to speak 5 randomly-selected words (i.e., 'help', 'run', 'music', 'record' and 'camera'). First, the volunteer speaks each word 30 times separately and we get 30 samples for each single word. Second, the volunteer speaks each word in sentences. Specifically, there are 5 sentences and each sentence contains one selected word. The volunteer speaks each sentence 30 times, and we manually segment the signal corresponding to the selected word based on recorded video. Then, we can get 30 samples for each selected word. After that, we adopt the trained model from the same volunteer, to recognize the words collected separately and those collected from sentences, respectively. As shown in Fig. 16, the recognition accuracy of words in sentences is a little lower than that of single words. This may be caused by the little difference in acoustic signal between the word in a sentence and the single word, since the words in a sentence may affect each other in pronunciation.

TABLE 9
Recognition accuracy of *LipReader* in different sessions

Single session	Multiple sessions
96.67%	96.00%

TABLE 10
Recognition accuracy of *LipReader* with user movement

Still	Walking	Jogging
96.67%	80.33%	56.67%

Effect of multiple sessions: In the proposed dataset, the signals corresponding to the same word are collected in a single session. However, in a real scenario, the user can speak the same word at different time, e.g., different days. To verify whether *LipReader* can achieve high performance of lip reading in different sessions, we invite one volunteer to speak each word 30 times in five days, i.e., speaking each word 6 times in a day and repeating the process in the following four days. Then, we select the samples collected in four of these days for training, while using the remaining samples for testing. As shown Table 9, the word recognition accuracy under different sessions achieves 96.00%, which is comparable to the recognition accuracy (i.e., 96.67%) under single session. It indicates that speaking in different sessions has little effect on lip reading performance. The reason may be that user habit changes little in different sessions.

Effect of user movement: In previous experiments, when the user silently-speaks a word, she/he keeps still, i.e., not moving. In fact, when the user uses *LipReader*, she/he can walk here and there. Thus in the experiment, we test the performance of *LipReader*, when the user speaks and walks at the same time. Specifically, we invite one volunteer to speak each word 6 times, and get $6 \times 50 = 300$ samples for testing. After that, we adopt the trained model corresponding to the same volunteer for evaluation. As shown in Table 10, speaking while walking may lead to the decrease of word recognition accuracy. However, when the user moves with a slow speed, e.g., in walking state, the word recognition accuracy can still achieves 80.33%.

Effect of devices: Considering the difference of devices, we also evaluate *LipReader* on other device, i.e., Xiaomi Redmi K40 Pro. Specifically, when using the new device, we invite one volunteer to collect 6 samples for each type of word, and get 6×50=300 test samples. Then we use the trained model corresponding to the same volunteer for evaluation. As shown in Table 11, the word recognition accuracy is 72.33%. The decrease of performance may be caused by the difference of signals, since the model is trained with signals from Samsung S9 smartphone while testing with signals from Redmi K40 smartphone. To reduce the effect of different devices, we introduce the fine-tuning strategy presented in Section 6, to make *LipReader* adapt to new device. Specifically, in a new device, the volunteer additionally collects 6 samples for each type of word. Then, we use these collected samples to fine-tune the pre-trained model, and use the fine-tuned model for testing. As shown in Table 11, after fine-tuning, the lip reading performance increases to 90.67%, i.e., *LipReader* can adapt to new device.

TABLE 11
Recognition accuracy of *LipReader* on different devices

Train: Samsung S9	Test: Redmi K40	
	Without fine-tuning	With fine-tuning
	72.33%	**90.67%**

TABLE 12
Recognition accuracy of *LipReader* under different environments

Train: Office	Test: Mall	
	Without fine-tuning	With fine-tuning
	64.00%	**84.67%**
	Test: Street	
	Without fine-tuning	With fine-tuning
	48.33%	**85.33%**

Effect of environments: Considering the noises from environments, we evaluate *LipReader* under new environments, i.e., mall and street. Specifically, in a new environment, the volunteer collects 6 samples for each type of word, and gets $6 \times 50 = 300$ testing samples. After that, we adopt the trained model in office (see Fig. 2) to evaluate *LipReader* on test samples in mall or street. As shown in Table 12, the different environments in training and testing lead to the decrease of lip reading performance. To reduce the effect of different environments, we introduce the fine-tuning strategy presented in Section 6, to make *LipReader* adapt to new environments. Specifically, under each new environment, the volunteer additionally collects 6 samples for each type of word. Then, we use these collected samples to fine-tune the pre-trained model, and use the fine-tuned model for testing. As shown in Table 12, after fine-tuning, the lip reading performance increases to 84.67% and 85.33% in mall and street, respectively. That is to say, *LipReader* can adapt to new environment with fine tuning.

7.7 Comparison with Previous Work

7.7.1 User-dependent Lip Reading

As described in Observation 2 of Section 5.1, the existing work often evaluates the performance of lip reading in user-dependent way. Therefore, in this subsection, we compare the proposed *LipReader* with EchoWhisper [5] and Endophasia [14] in user-dependent way. Here, EchoWhisper [5] and Endophasia [14] focused on word-level lip reading by acoustic signals, as mentioned in Section 2. Besides, EchoWhisper [5] enhanced the emitted acoustic signals by utilizing two microphones, Endophasia [14] modulated the emitted acoustic signals with GSM training sequence, while our *LipRerader* only adopts one microphone to receive continuous wave. When considering the lack of emitted signals (i.e., raw data) from EchoWhisper and Endophasia, we make the comparison on our LIPCMD dataset. Specifically, for each sample on LIPCMD dataset, we provide the same

TABLE 13
Comparison of LipReader and other approaches on LIPCMD dataset

	LipReader	EchoWhisper	Endophasia
1 syllable	**90.97%**	79.07%	84.77%
2 syllables	**91.67%**	78.67%	86.61%
3 syllables	**93.00%**	81.75%	87.83%
Confused words	**88.61%**	74.31%	81.86%
All words	**91.58%**	79.48%	85.93%

input, i.e., signal gradient matrix, to *LipReader*, EchoWhisper and Endophasia for recognition/classification. After that, we compare the recognition accuracy of *LipReader*, EchoWhisper and Endophasia on one-syllable words, two-syllable words, three-syllable words, confused words, and all words, respectively. As shown in Table 13, whatever the words are, our *LipReader* can achieve a better recognition performance. Take ‘all words’ as an example, the recognition accuracy of *LipReader* is 91.58%, which outperforms that of EchoWhisper and Endophasia by 12.10% and 5.65%, respectively. It indicates that our *LipReader* can achieve a good performance of acoustic-based lip reading and outperforms the previous work in terms of user-dependent performance.

7.7.2 User-independent Lip Reading

To adapt to new users, we propose a fine-tuning strategy to allow *LipReader* to work in user-independent way, as described in Section 6. Therefore, in this experiment, we also evaluate the performance of *LipReader* in user-independent way, and compare it with Endophasia [14] which adopted Few-Shot Adversarial Domain Adaptation (FADA) method for user adaptation. Specifically, we randomly invite 10 volunteers to participate in the experiment. Everytime, we select the samples from 9 volunteers (i.e., training users) for model training. In regard to the other volunteer (i.e., test user), we adopt a part of samples (i.e., 0% - 80%) of each word for fine-tuning in *LipReader* or re-training in Endophasia, while using the remaining samples (i.e., 20%) for testing. Each volunteer will be selected for testing once, and we average the results of different volunteers. As shown in Fig. 17, we use blue, yellow bar to represent the recognition performance of *LipReader*, Endophasia, respectively. When no samples of the test user are adopted (for fine-tuning or re-training), the recognition accuracy is low. However, when using only a small number of samples (e.g., 6 samples of each word) from test user, the recognition accuracy increases a lot, i.e., 83.67% for *LipReader* and 68.67% for Endophasia. As the number of samples from test user increases, the recognition accuracy also increases. Usually, our *LipReader* can achieve a better performance than Endophasia. In addition, the fine-tuning cost of *LipReader* is less than the re-training cost of Endophasia, since *LipReader* only needs a few samples from test user to fine tune the pre-trained model, while Endophasia needs both a few samples from test user and all samples from training users to re-train the whole model. It indicates that the proposed fine-tuning strategy can make *LipReader* adapt to new user well with a small cost, i.e., only using a few samples from new user.

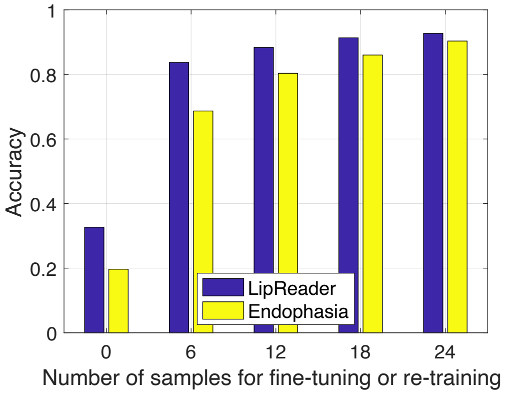


Fig. 17. Comparison of *LipReader* and Endophasia in user-independent way

8 Case Study

In addition to evaluating the recognition performance of *LipReader* on LIPCMD dataset, we also implement *LipReader* on Samsung Galaxy S9 smartphone. First, we implement *LipReader* with PyTorch and perform the model training offline. Then, we use the ‘PyTorch Mobile’ [48] workflow to transform the trained model to a lightweight model (i.e., 42.6 MB) which can be deployed on smartphone. Besides, to reduce the time latency of lip reading, we introduce the multi-thread strategy to calculate the time-consuming Short-Time Fourier Transform (STFT) in parallel. Specifically, since Samsung S9 smartphone has 8 cores, we totally adopt 8 threads, including one main thread and other seven parallel threads for STFT. As shown in Fig. 18, in STFT, the main thread first divides time-domain acoustic signals into equal-length segments. Then, the eight threads perform STFT on divided segments in parallel. After that, the main thread merges STFT results from segments to get the final STFT result. In this way, we can achieve online lip reading for mobile devices. It is worth noting that the deployed model on smartphone belongs to a user-dependent model, since a smartphone is usually used by a specific user.

In Fig. 19, we show a typical usage of *LipReader* on smartphone. When the user presses the ‘START’ button, *LipReader* immediately emits the acoustic signals. Then, the smartphone records the acoustic signals corresponding to silent speech, as the blue signals shown the right part of Fig. 19. After that, when the user releases the ‘START’ button, *LipReader* stops recording and processes acoustic signals to get the signal gradient matrix, as the color map shown in the right part of Fig. 19. Finally, the signal gradient matrix is input to the trained neural model, which outputs the recognized result (i.e., ‘add’). It is worth noting that the signals/images shown in Fig. 19 are used for illustration and not shown in the real application.

When running *LipReader* on Samsung S9 smartphone, we also evaluate the time latency and power consumption. For time latency, it means the duration from releasing ‘START’ button (i.e., end of silent speech) to outputting the recognition result. Totally, the time latency is 635ms, including 206ms used for processing acoustic signals and 429ms used for recognizing the word. When compared with the existing work on acoustic-based lip reading, we are the first to achieve online word recognition in a mobile device

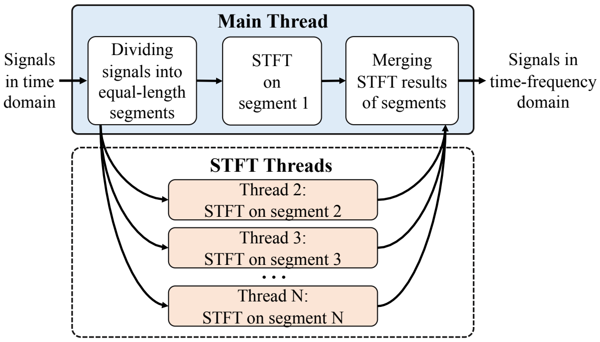

Main Thread
Signals in time domain
Dividing signals into equal-length segments
STFT on segment 1
Merging STFT results of segments
Signals in time-frequency domain
STFT Threads
Thread 2: STFT on segment 2
Thread 3: STFT on segment 3
• • •
Thread N: STFT on segment N

Fig. 18. Multi-thread strategy adopted to perform STFT in parallel

with acceptable time latency. For power consumption, we use BatteryHistorian [49] for measurement and the average power consumption is 505.1 ± 26.3mW. In the measurement, we first measure power consumption P_0 when only keeping screen on, then we measure power consumption P_1 when running *LipReader*, which includes emitting acoustic signals, processing signals and recognizing the word. After that, we get the power consumption $\Delta P = P_1 - P_0$ caused by *LipReader*. In the experiment, we randomly invite 5 users where everyone silently speaks 50 words, and then average the power consumption in each test.

9 Discussion

Microphone and acoustic signal: We use one embedded microphone of smartphone and adopt sinusoidal signals without modulation for lip reading, aiming to make the approach work on device with weak sensing ability and computing power. However, considering that current smartphones are often embedded with two microphones, using more microphones to get enhanced acoustic signals for lip reading can also be a good solution. Besides, applying suitable modulations on sinusoidal signals may provide better acoustic signals for lip reading. To make *LipReader* work with the enhanced or modulated signals, it is necessary to appropriately preprocess the signals. In future, we will make further research on these aspects.

Model fine-tuning: In Section 6, we propose a fine-tuning strategy to update the pre-trained model, to make the model adapt to new users. The model is pre-trained on a server and the fine-tuning strategy is also performed on the server. After fine tuning, the updated model on the server will be sent back to mobile device for lip reading. However, in some cases, the user may have no access to the server or be not willing to upload the acoustic signals of lip reading. To address this issue, the fine-tuning strategy which can update the pre-trained model on mobile device is expected. We will make the research in future.

Unexpected disturbances: The unexpected disturbances caused by placement of device, user movement, device type, and environment may affect word recognition accuracy. In these cases, to achieve high performance of lip reading, we can adopt the fine-tuning strategy, which requires to collect a few samples in new scenarios. To further remove the labor cost of collecting samples for fine tuning, in future, we will try to use adversarial learning to reduce the effect of disturbances.

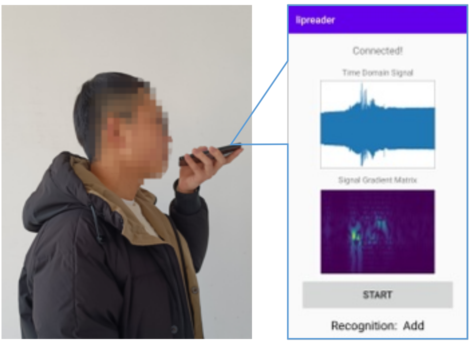


Fig. 19. A typical case of using *LipReader* for lip reading

Phrase-level or sentence-level lip reading: This paper focuses on word-level lip reading, where the acoustic signal of each word is collected separately. This can be different from phrase-level or sentence-level lip reading, where the acoustic signals of consecutive words may be close to or affect each other. Usually, if there is a suitable pause between consecutive words, we can segment each word based on pauses, and then adopt *LipReader* to recognize extracted words for phrase-level or sentence-level lip reading in an indirect way. However, if the user speaks a phrase or sentence quickly, i.e., word segmentation is difficult, more efficient solutions will be expected.

10 Conclusion

In this paper, we utilize the acoustic signals emitted from smartphone for lip reading. Considering the lack of public dataset in acoustic-based lip reading, we propose and release a lip-reading dataset LIPCMD. Besides, we provide benchmark evaluation on the dataset. To improve the performance of lip reading, we propose a self distillation based approach *LipReader*, which distills the attention map and probability distribution in the convolutional neural network itself to improve feature representation and classification performance. The extensive experimental results show that *LipReader* can achieve a good recognition accuracy of 91.58% for lip reading and outperforms the benchmark solutions and previous work.

Acknowledgments

This work is supported by Jiangsu Provincial Key Research and Development Program under Grant BE2020001-4; National Natural Science Foundation of China under Grant Nos. 62172208, 62272216, 61832008. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization. Lei Xie is the corresponding author.

References

[1] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with wi-fi!" *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2907–2920, 2016.

[2] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 87–103.

[3] Y. M. Assael, B. Shillingford, S. Whiteson, and N. De Freitas, "Lipnet: End-to-end sentence-level lipreading," *arXiv preprint arXiv:1611.01599*, 2016.

[4] J. Tan, C.-T. Nguyen, and X. Wang, “Silenttalk: Lip reading through ultrasonic sensing on mobile phones,” in *IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[5] Y. Gao, Y. Jin, J. Li, S. Choi, and Z. Jin, “Echowhisper: Exploring an acoustic-based silent speech interface for smartphone users,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–27, 2020.

[6] A. Kapur, S. Kapur, and P. Maes, “Alterego: A personalized wearable silent speech interface,” in *The 23rd International conference on intelligent user interfaces*, 2018, pp. 43–53.

[7] M. Kim, N. Sebkhi, B. Cao, M. Ghovanloo, and J. Wang, “Preliminary test of a wireless magnetic tongue tracking system for silent speech interface,” in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2018, pp. 1–4.

[8] B. Min, J. Kim, H.-j. Park, and B. Lee, “Vowel imagery decoding toward silent speech bci using extreme learning machine with electroencephalogram,” *BioMed research international*, vol. 2016, 2016.

[9] S. Zhang, Z. Ma, K. Lu, X. Liu, J. Liu, S. Guo, A. Y. Zomaya, J. Zhang, and J. Wang, “Hearme: Accurate and real-time lip reading based on commercial rfid devices,” *IEEE Transactions on Mobile Computing*, 2022.

[10] J. Wang, C. Pan, H. Jin, V. Singh, Y. Jain, J. I. Hong, C. Majidi, and S. Kumar, “Rfid tattoo: A wireless platform for speech recognition,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1–24, 2019.

[11] X. Liu, J. Yin, S. Zhang, K. Li, and S. Guo, “Receive only necessary: Efficient tag category identification in large-scale rfid systems,” *IEEE Transactions on Mobile Computing*, 2021.

[12] X. Liu, X. Chen, Q. Yang, S. Zhang, S. Guo, J. Luo, and K. Li, “More than scheduling: Novel and efficient coordination algorithms for multiple readers in rfid systems,” *IEEE Transactions on Mobile Computing*, 2022.

[13] C. Cai, R. Zheng, and J. Luo, “Ubiquitous acoustic sensing on commodity iot devices: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 432–454, 2022.

[14] Y. Zhang, W.-H. Huang, C.-Y. Yang, W.-P. Wang, Y.-C. Chen, C.-W. You, D.-Y. Huang, G. Xue, and J. Yu, “Endophasia: Utilizing acoustic-based imaging for issuing contact-free silent speech commands,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–26, 2020.

[15] Q. Zhang, D. Wang, R. Zhao, and Y. Yu, “Soundlip: Enabling word and sentence-level lip interaction for smart devices,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–28, 2021.

[16] P. Sujatha and M. R. Krishnan, “Lip feature extraction for visual speech recognition using hidden markov model,” in *International Conference on Computing, Communication and Applications*. IEEE, 2012, pp. 1–5.

[17] A. Rekik, A. Ben-Hamadou, and W. Mahdi, “A new visual speech recognition approach for rgb-d cameras,” in *International conference image analysis and recognition*. Springer, 2014, pp. 21–28.

[18] X. Ma, L. Yan, and Q. Zhong, “Lip feature extraction based on improved jumping-snake model,” in *The 35th Chinese Control Conference (CCC)*. IEEE, 2016, pp. 6928–6933.

[19] M. Wand, J. Koutník, and J. Schmidhuber, “Lipreading with long short-term memory,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6115–6119.

[20] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, “Lip reading sentences in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3444–3453.

[21] T. Afouras, J. S. Chung, and A. Zisserman, “Deep lip reading: a comparison of models and an online application,” *arXiv preprint arXiv:1806.06053*, 2018.

[22] K. Sun, C. Yu, W. Shi, L. Liu, and Y. Shi, “Lip-interact: Improving mobile device interaction with silent speech commands,” in *The 31st Annual ACM Symposium on User Interface Software and Technology*, 2018, pp. 581–593.

[23] Z. Ma, S. Zhang, J. Liu, X. Liu, W. Wang, J. Wang, and S. Guo, “Rf-siamese: Approaching accurate rfid gesture recognition with one sample,” *IEEE Transactions on Mobile Computing*, 2022.

[24] A. Czyzewski, B. Kostek, P. Bratoszewski, J. Kotus, and M. Szykulski, “An audio-visual corpus for multimodal automatic speech recognition,” *Journal of Intelligent Information Systems*, vol. 49, no. 2, pp. 167–192, 2017.

[25] M. Cooke, J. Barker, S. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[26] N. Harte and E. Gillen, “Tcd-timit: An audio-visual corpus of continuous speech,” *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 603–615, 2015.

[27] L. F. Lamel, R. H. Kassel, and S. Seneff, “Speech database development: Design and analysis of the acoustic-phonetic corpus,” in *Speech Input/Output Assessment and Speech Databases*, 1989.

[28] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan, “Audiogest: enabling fine-grained hand gesture detection by decoding echo signal,” in *ACM international joint conference on pervasive and ubiquitous computing*, 2016, pp. 474–485.

[29] B. Fu, D. V. Gangatharan, A. Kuijper, F. Kirchbuchner, and A. Braun, “Exercise monitoring on consumer smart phones using ultrasonic sensing,” in *The 4th international Workshop on Sensor-based Activity Recognition and Interaction*, 2017, pp. 1–6.

[30] P. Heckbert, “Fourier transforms and the fast fourier transform (fft) algorithm,” *Computer Graphics*, vol. 2, pp. 15–463, 1995.

[31] T. Gong, H. Cho, B. Lee, and S.-J. Lee, “Knocker: Vibroacoustic-based object recognition with smartphones,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–21, 2019.

[32] M. Chen, P. Yang, J. Xiong, M. Zhang, Y. Lee, C. Xiang, and C. Tian, “Your table can be an input panel: Acoustic-based device-free interaction recognition,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–21, 2019.

[33] H. Watanabe and T. Terada, “Improving ultrasound-based gesture recognition using a partially shielded single microphone,” in *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, 2018, pp. 9–16.

[34] S. Gupta, D. Morris, S. Patel, and D. Tan, “Soundwave: using the doppler effect to sense gestures,” in *The SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 1911–1914.

[35] X. Xu, J. Yu, Y. Chen, Y. Zhu, S. Qian, and M. Li, “Leveraging audio signals for early recognition of inattentive driving with smartphones,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1553–1567, 2017.

[36] J. B. Allen and L. R. Rabiner, “A unified approach to short-time fourier analysis and synthesis,” *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[37] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, Y. Liu, and M. Li, “Lippass: Lip reading-based user authentication on smartphones leveraging acoustic signals,” in *IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1466–1474.

[38] F. Ott, M. Wehbi, T. Hamann, J. Barth, B. Eskofier, and C. Mutschler, “The onhw dataset: Online handwriting recognition from imu-enhanced ballpoint pens with machine learning,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–20, 2020.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[40] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[41] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, “Be your own teacher: Improve the performance of convolutional neural networks via self distillation,” in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3713–3722.

[43] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, “Learning lightweight lane detection cnns by self attention distillation,” in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1013–1021.

[44] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[45] S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” *arXiv preprint arXiv:1612.03928*, 2016.

[46] “Upsampling,” 2021. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.functional.upsample.html?highlight=upsample#torch.nn.functional.upsample

[47] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[48] "Pytorch mobile," 2021. [Online]. Available: https://pytorch.org/mobile/android/
[49] "Battery historian," 2016. [Online]. Available: https://github.com/google/battery-historian

Yafeng Yin received her B.E. degree in network engineering from Nanjing University of Science and Technology, and received her Ph.D. degree in computer science from Nanjing University, China in 2011 and 2017, respectively. She is currently an associate researcher in the Department of Computer Science and Technology at Nanjing University. She has published papers in IEEE Transactions on Mobile Computing, IEEE Transactions on Computers, ACM UbiComp, ACM MM, IEEE INFOCOM, etc. Her research interests include mobile sensing, wearable computing, etc.

Zheng Wang received his B.E. degree in Computer Science and Technology from Nanjing University of Science and Technology, China in 2019. He is currently a third year graduate student in the Department of Computer Science and Technology at Nanjing University. His research interests include mobile sensing, and wearable computing.

Kang Xia received his B.E. degree in Computer Science and Technology from Jiangsu University of Science and Technology, China in 2020. He is currently a second year graduate student in the Department of Computer Science and Technology at Nanjing University. His research interests include mobile sensing and action recognition.

Lei Xie received his B.S. and Ph.D. degrees from Nanjing University, China in 2004 and 2010, respectively, all in computer science. He is currently a professor in the Department of Computer Science and Technology at Nanjing University. He has published over 100 papers in IEEE Transactions on Mobile Computing, ACM/IEEE Transactions on Networking, IEEE Transactions on Parallel and Distributed Systems, ACM Transactions on Sensor Networks, ACM MobiCom, ACM UbiComp, ACM MobiHoc, IEEE INFOCOM, IEEE ICNP, IEEE ICDCS, etc.

Sanglu Lu received the BS, MS, and PhD degrees from Nanjing University, China, in 1992, 1995, and 1997, respectively, all in computer science. She is currently a professor in the Department of Computer Science and Technology at Nanjing University. Her research interests include distributed computing and pervasive computing. She is a member of the IEEE.