# ProxiMic: Convenient Voice Activation via Close-to-Mic Speech Detected by a Single Microphone


Yue Qin
Department of Computer Science and Technology, Tsinghua University, Beijing, China
Key Laboratory of Pervasive Computing, Ministry of Education, China
qiny19@mails.tsinghua.edu.cn

Chun Yu†
Department of Computer Science and Technology, Tsinghua University, Beijing, China
Key Laboratory of Pervasive Computing, Ministry of Education, China
chunyu@tsinghua.edu.cn

Zhaoheng Li
Department of Computer Science and Technology, Tsinghua University, Beijing, China
lizhaoha17@mails.tsinghua.edu.cn

Mingyuan Zhong
Computer Science & Engineering, University of Washington, Seattle, WA, USA
myzhong@cs.washington.edu

Yukang Yan
Department of Computer Science and Technology, Tsinghua University, Beijing, China
Key Laboratory of Pervasive Computing, Ministry of Education, China
yyk15@mails.tsinghua.edu.cn

Yuanchun Shi
Department of Computer Science and Technology, Tsinghua University, Beijing, China
Key Laboratory of Pervasive Computing, Ministry of Education, China
shiyc@tsinghua.edu.cn


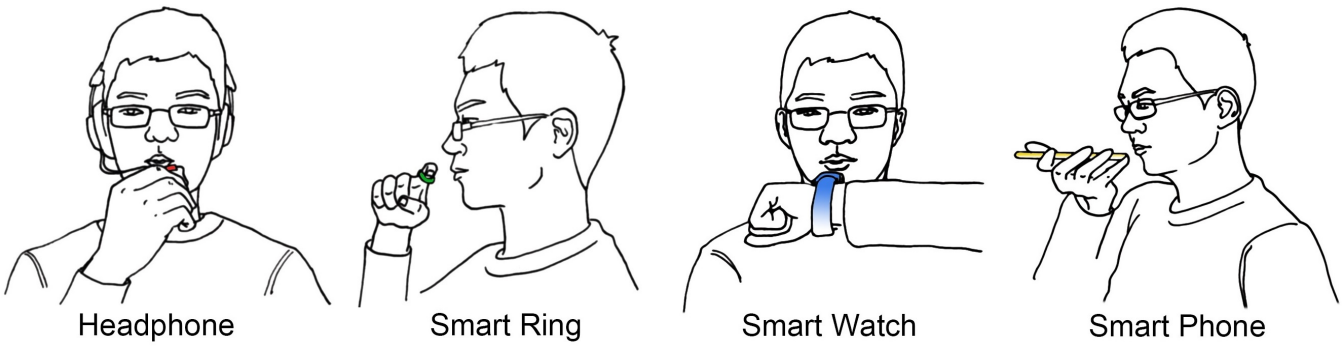


**Figure 1: Users perform voice input using ProxiMic on a variety of devices, keeping the microphones close to their mouths.**


## ABSTRACT

Wake-up-free techniques (e.g., Raise-to-Speak) are important for improving the voice input experience. We present ProxiMic, a close-to-mic (within 5 cm) speech sensing technique using only one microphone. With ProxiMic, a user keeps a microphone-embedded device close to the mouth and speaks directly to the device without wake-up phrases or button presses. To detect close-to-mic speech, we use the feature from pop noise observed when a user speaks and blows air onto the microphone. Sound input is first passed through a low-pass adaptive threshold filter, then analyzed by a CNN which detects subtle close-to-mic features (mainly pop noise). Our two-stage algorithm can achieve 94.1% activation recall, 12.3 False Accepts per Week per User (FAWU) with 68 KB memory size, which can run at 352 fps on the smartphone. The user study shows that ProxiMic is efficient, user-friendly, and practical.




*CHI '21, May 8–13, 2021, Yokohama, Japan*



ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445687



## CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Interaction techniques*; Sound-based input / output.

## KEYWORDS

voice input, sensing technique, activity recognition

**ACM Reference Format:**
Yue Qin, Chun Yu†, Zhaoheng Li, Mingyuan Zhong, Yukang Yan, and Yuanchun Shi. 2021. ProxiMic: Convenient Voice Activation via Close-to-Mic Speech Detected by a Single Microphone. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411764.3445687

# 1 INTRODUCTION

Voice interaction is convenient, efficient, and intuitive, and therefore users complete a variety of tasks via voice input, including communication, text entry, and issuing commands [20, 23, 30, 31]. Despite its wide adoption, voice input still faces three challenges. First, the voice wake-up process can be lengthy. Second, such interaction can be tedious and annoying especially during multiple rounds of voice input. Third, users risk exposing their privacy as their voices may be overheard or eavesdropped by a third party.

Previous work has attempted to address each of the individual challenges above. Raise to Speak[52], PrivateTalk [47] and ProxiTalk [49] use a variety of sensors and device-related gestures to shorten the wake-up time for specific devices. However, these techniques are limited in their application by their requirements for multiple sensors or specific device form factors. For multiple rounds of dialogue, voice input without repeated wake-up and interruption while listening and speaking at any time is still a challenging problem. Some existing works attempt to remedy this by using semantic understanding or sentiment analysis [27, 48], but it still cannot be completely solved due to the complexity of natural language. To address privacy issues, some prior work addressed how to perform completely silent speech, which means speaking but no sound is made [10, 18, 19]. However, due to the complex device requirements and limited distinguishable phrase set it supported, silent speech has not yet been popularized.

We propose ProxiMic, a novel and low-cost input method that provides a solution to the above three challenges simultaneously by accurately detecting close-to-mic (within 5 cm) human voice. As shown in Figure 1, with ProxiMic, a user can position any microphone-embedded device close to the mouth, start speaking directly without wake-up phrases. By recognizing close-to-mic speech, ProxiMic can conveniently support multiple rounds of dialogue and interruption while speaking. In addition, ProxiMic supports whispering (speaking without vibrating the vocal-fold; airflow only), which can protect privacy.

To detect close-to-mic speech, we use a Convolutional Neural Network (CNN) to capture subtle close-to-mic features, especially consonants with pop noise. Pop noise caused by air being blown onto the microphone while speaking (e.g., most English words containing "b,c,d,f,j,k,l,p,q,r,s,t,v,w,x,y,z" generate such airflow, and this airflow will generate a surge of amplitude of the audio signal). In addition, considering the high amplitude of pop noise, we design an Adaptive Amplitude Threshold Trigger (AATT) to filter out daily noise to reduce CNN calculation. The AATT+CNN two-stage detection pipeline allows us to recognize close-to-mic speeches with high accuracy, low power consumption, and low memory utilization.

We conducted three studies to test the performance and usability of ProxiMic.

In Study 1, we created data sets of four audio types from 102 users on 55 devices and 49 different environments. We tested the performance of ProxiMic to reject false positives and recognize close-to-mic speeches on this data set. Key performance metrics for ProxiMic are comparable to that of Raise to Speak [52], with 94.1% activate recall and 12.3 False Accepts per Week per User (FAWU).

In study 2, we conducted a preliminary user study to test the influence of different factors. A white box analysis of the interpretability of our CNN model shows that the model has indeed learned the joint features of pop-noise and human voice. We also conducted an ASR accuracy test, which verified that pop-noise did not interfere with speech recognition. Then we tested different form factors to analyze the generalizability of the algorithm. In terms of privacy, we verified that close-to-mic whispering which can be effectively recognized by ProxiMic is a private and effective voice input method, which can hardly be heard by other eavesdroppers.

Finally, in study 3, we conducted a comparison user study to evaluate the user experience of ProxiMic. Results show that users consider ProxiMic to be efficient, user-friendly, and practical compared to baseline activation methods.

In sum, there are three main contributions in this paper:

(1) We proposed a novel wake-up-free method that doesn't require special gestures and complex sensors and can be deployed on various forms of handheld and wearable devices.
(2) We specifically designed a two-stage algorithm for close-to-mic speech recognition. By utilizing subtle close-to-mic features (mainly pop noise), ProxiMic provides a low-power, feasible, and practicable solution to voice activation.
(3) We evaluated various boundary performances and user feedback of ProxiMic, which provide guidance for the deployment of real applications.

# 2 RELATED WORK

## 2.1 Activating Voice Input

Using sensors to detect voice input events has been well studied. In the early years, voice activity detection (VAD) algorithms have been developed to detect the presence of human speech [32, 41, 43]. The use of sound signals can detect the occurrence of various events [9, 28, 38]. Thanks to the robustness of Key Word Spotting (KWS) technology, the wake-up phrase as a significant event is used as the conventional activation method of voice assistants [12, 22, 39, 50]. However, because the wake-up phrase still faces problems such as cumbersome interaction, privacy, and security, the interest of researchers in recent years has gradually been attracted by various activation methods. Gaze wake-up is used for devices with relatively fixed positions, such as in-vehicles, smart speakers, etc. The user wakes up the voice input by looking at the device when speaking [26, 33]. PrivateTalk detects the hand pose of covering the mouth from one side to activate voice input for bluetooth earbuds [47]. FaceSight [46] deployed a camera on glasses to recognize the cover-mouth gesture to activate voice input in the AR scenario. Apple introduced a *Raise to Speak* feature to support wake-up-free activation for smartwatches [1, 52] which requires 4 inches of vertical wrist movement. ProxiTalk [49] comprehensively uses the signals of the camera, IMU, and two microphones to implement the robust voice activation system for smartphone. Different from all

the above, ProxiMic uses only one microphone and close-to-mic speech to perform voice input without wake-up phrase.

## 2.2 Privacy and Security of Voice Input

Inconvenience of privacy risks is the important concern with voice input. Especially in public places, users are not inclined to use voice input [8]. Therefore, silent voice interface has become a direction of research. In order to achieve silent speech, researchers have proposed many methods such as Brain-Computer-Interface (BCI) [5, 29], and electromyography (EMG) [7, 18]. Sun et al.'s Lip-Interact [42] repurposes the front camera of smartphone to capture the user's mouth movements and recognize 44 issued commands with an end-to-end deep learning model. AlterEgo [18] allows a user to silently converse with a computing device without any voice or any discernible movements by facial neuromuscular input. SottoVoce [19] uses a skin-contact ultrasonic sensor at the larynx to recognize tongue movements to achieve silent speech. EchoWhisper [11] leverages the Doppler shift of reflection of near-ultrasound sound waves caused by the mouth and tongue movements to recognize 45 words. SilentVoice [10] uses a device close to the mouth and adopts ingressive speech for silent voice input to ensure privacy, and recognizes the voice content only by the airflow sound generated by inhalation. Different from the above-mentioned methods, ProxiMic doesn't seek completely silent speech. We hope to use an acceptable ultra-small volume from whispering to realize the voice interaction with a large word set for various everyday devices.

## 2.3 Close-to-Mic Speech Detection

The core of ProxiMic is to determine whether the source of the voice signal is close enough to the device. Shiota et al. [36, 37] used pop noise as a feature in voice liveness detection (VLD), which can be used to classify whether a given segment of human speech was spoken by a real person. We use the feature of pop noise too and further push it to voice activation techniques. We focus on the robustness of complex environments with low power consumption. Some works use mic-array to localize the sound source [6, 21, 34, 35, 44, 45]. But for handheld and wearable devices, deploying the microphone array seems to be expensive. For the setting of two microphones, Volume Difference and Time Difference is the mainly methods to estimate the vocal distance [2, 3, 14]. ProxiTalk [49] has studied the distance classification task of smartphones with dual microphones. Because of the significant volume difference, it is usually easy to determine which one of the microphones is the sound source close to, but determining whether the sound source is close-to-mic (within 5 cm) is also challenging. Due to its simple hardware and easy deployment in embedded devices, the distance measurement method based on a single microphone has gradually attracted the interest of researchers in recent years [13]. In this work, we focus on single microphone, which can be applicable to various devices. We also believe that for the existing multi-microphone systems or devices with multi-sensors (e.g., IMU, camera, and proximity), we can get better close-to-mic detection performance by additionally using the unique features we utilized and the two-stage algorithm we presented.

# 3 SIGNAL ANALYSIS OF CLOSE-TO-MIC SPEECH

We conducted a pilot study to understand the characteristics of close-to-mic speech. We specifically looked for features that showed potential to be easily computed on performance- and energy- constrained devices, while achieving high accuracy at the same time. We found two promising features, namely sound amplitude and spectrogram characteristics of pop noise, which we detail below.

## 3.1 Collecting Audio Samples

We recruited three participants from the university campus for this study. For each participant, we conducted the experiment in eight different environments and we asked them to speak close to an on-device microphone while we collected data for their close-to-mic speech. Nowadays, almost all of the microphones of wearable devices and handheld devices are Electret Condenser Microphone (ECM) or Micro-Electro-Mechanical System (MEMS) which have similar acoustic characteristics, so in this study, we choose the internal microphone (MEMS) of a Huawei P30 smartphone for recording voice command.

## 3.2 Sound Amplitude Characteristics

By plotting the waveforms of recorded speech, we observe higher amplitudes when a person is speaking than that of background noise at all distances, but this is especially prominent for speech recorded at 2 cm from the microphone (Figure 2). This is because of the proximity of the sound source, as well as the presence of pop noise. Pop noise is produced when the weak airflow is generated by speaking causes strong vibrations in the microphone diaphragm nearby, which translates to high amplitudes in the audio signal.

We then compare the amplitudes of speech with environmental noises (Figure 3). Again, close-to-mic speech within 5 cm produced higher amplitudes than that of all other noisy environments we sampled.

## 3.3 Spectrogram Characteristics

Spectrogram is the spectrum of frequencies of a signal as it varies with time, and it is an effective method for analyzing sound components and timbre [16]. We plot the spectrogram for three types of speech recorded in the hospital hall (Figure 4). The green to yellow background color is mostly environmental noise.

On the left is a reference sample containing normal voice recorded at a distance of 30 cm, where pop noise doesn't appear. We note that vocal patterns can be observed mainly between the 200–800 Hz frequency range (colored in orange and red), and does not extend below 100 Hz (green).

Pop noise can be clearly heard on the rest of the speech samples, and shows clear distinction from the 30 cm reference sample. Both samples captured at 2 cm show strong pop noise features, which are the vertical spikes in the spectrogram extending from 0 to around 2000 Hz. In the close-up crops of the low-frequency range, the difference between the reference sample and close-to-mic speech is particularly prominent at around 50 Hz, as close-to-mic speech has more energy in the low-frequency region. This is consistent with Shiota et al.'s observation [37].

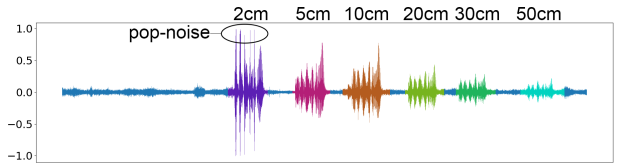


**Figure 2: An user read [ˈdʒiŋˈtiænˈtiænˈtʃiːˈzenˈməˈjʌŋ] which means "how is the weather today" at six different distances with normal volume in a noisy hospital hall. Sound amplitude is normalized between -1 and 1.**

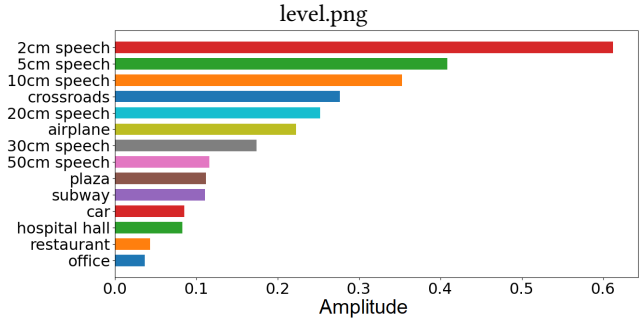


**Figure 3: The 90$^{th}$ percentile amplitude with 10 ms of maximum filter in different settings, including speech. The amplitudes of close-to-mic speech at 2 cm and 5 cm are higher than that of the loudest environment noises recorded.**

When compared with normal voice at 2 cm, whispering, which can be considered as pop noise and airflow only speech, is relatively easy to distinguish due to its lack of 200–400 Hz vocals. This left the red vertical spikes narrower on the bottom, with wider gaps in between.

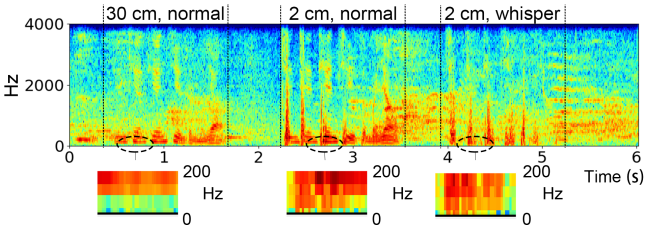


**Figure 4: The spectrogram of [ˈdʒiŋˈtiænˈtiænˈtʃiːˈzenˈməˈjʌŋ] (means "how is the weather today") at three different distances in a noisy hospital hall. From left to right, the indicated segments were recorded at 30 cm with normal voice (no pop noise), at 2 cm with normal voice (with strong pop noise), and at 2 cm with whispering (only pop noise and airflow), respectively. 0–200 Hz close-ups are provided below to illustrate low-frequency pop noise characteristics.**

## 3.4 Design Implication

In light of the results from our feature analysis, we adopt a two-stage approach to detect close-to-mic speech both accurately and efficiently.

The first stage detector is based on the amplitude threshold. Due to the high amplitude of close-to-mic speech, we identify potential close-to-mic voice based on Adaptive Amplitude Threshold Trigger (AATT) to effectively filter out low energy noise with minimal calculation required. Due to the low-frequency characteristics of pop noise, we use a low-pass filter to enhance the performance of AATT. If the amplitude of audio signal exceeds the dynamic threshold of AATT, ProxiMic will extract a one-second audio clip and hand this raw signal input over to the second stage detector for refined detection.

For the second stage detection, since pop noise has obvious spectrogram characteristics, we use the spectrogram-based Convolutional Neural Network (CNN) for refined detection. Our white-box analysis below verifies that CNN indeed recognized mainly pop noise.

All of the features we observed extended well below 4 kHz. In order to reduce power consumption and the usage of computing resources, we limit our sampling rate to 8 kHz for the rest of this paper.

# 4 THE PROXIMIC DETECTOR

We introduce the specific parameters and components of the two-stage algorithm here.

## 4.1 Adaptive Amplitude Threshold Trigger

Adaptive Amplitude Threshold Trigger (AATT) identifies potential close-to-mic speech by maintaining a threshold that matches the noise level. According to pop-noise's strong low-frequency characteristics, the signal first passes through a first-order low-pass filter with a cutoff frequency at 50 Hz. If the amplitude of audio signal exceeds the dynamic threshold $T$, ProxiMic will extract a one-second audio clip and hand this raw signal input over to the CNN for refined detection. Figure 5 shows an example of the dynamic threshold and the high amplitude of close-to-mic speech signal (2 cm) in a noisy environment.

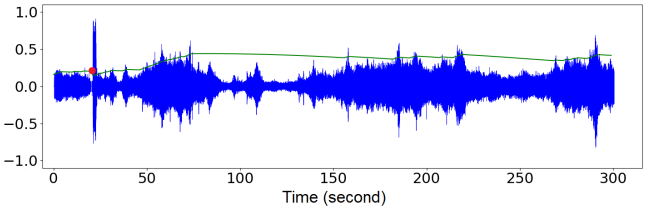


**Figure 5: Adaptive thresholding applied to a 300-second sound sample recorded at a busy crossroad, where the noise of cars and trucks can be heard constantly. A close-to-mic speech (2 cm) is marked by a red dot. The green line represents the adaptive amplitude threshold. In 77.5% of the time, background noise amplitude remains below the amplitude threshold.**

According to the output of CNN, we adjust the adaptive threshold $T$ according to Equation 1. According to our experimental results below, we set the parameters $T_{low}$, $T_{high}$, $A_{low}$, $\alpha$, $\beta$, and $\gamma$ to 0.17, 0.8, 0.995, 0.01, 0.05, and $5 * 10^{-5}$ respectively. This set of parameters allows the threshold to remain stable for one minute

in a noisy environment and return to the lowest level for up to 5 minutes in a quiet environment. In particular, to reduce the amount of calculation, we always refresh the threshold $T$ with a one-second cycle instead of updating at 8 kHz.

$$T_{k+1} = \begin{cases} T_{low}, & \text{accepted by CNN} \\ \min(A_k(T_k + \alpha), T_{high}), & \text{rejected by CNN} \\ \max(A_k T_k, T_{low}), & \text{reject by AATT} \end{cases} \quad (1)$$

$$A_{k+1} = \begin{cases} A_{low}, & \text{accepted by CNN} \\ A_k + \beta(1 - A_k), & \text{rejected by CNN} \\ \max((1-\gamma)A_k, A_{low}), & \text{reject by AATT} \end{cases} \quad (2)$$

## 4.2 CNN-Based Spectrogram Detector

In the second stage, we employ spectrogram features to determine if a sound snippet is close-to-mic speech through a CNN classifier. Whenever an audio segment passes the amplitude threshold in the first stage, the CNN calculates features from a one-second audio snippet centered around that audio segment, which is then passed on to the CNN classifier. This leads to a 0.5s latency required for CNN detection. Figure 6 shows the structure of the CNN-based spectrogram detector model.

The CNN detector first extracts an $80 \times 201$ two-dimensional time-frequency spectrogram of the one-second input signal (at 8 kHz sample rate) by Short-Time Fourier Transform (STFT). The STFT window size is set to 20 ms, the hop length is set to 5 ms. We apply the logarithmic transformation to the two-dimensional time-frequency map generated by STFT.

To provide sufficient resolution at low frequencies, and to reduce the computational overhead and improve the generalization ability of the model, we utilize a low-frequency enhanced triangular filter bank, which produces a $20 \times 201$ feature map. The center of the triangular filter equals to the STFT center at the frequency lower than 250 Hz, and at the frequency higher than 250 Hz, the filter center is equidistantly distributed according to Mel Frequency.

The $20 \times 201$ feature map is the input to our CNN classifier. The first layer of the CNN is a 1D convolution layer with 50 filters of size 3. The convolutions are along the temporal dimension, creating a feature map of size 201 for each filter. The feature map is then passed through a second similar convolutional layer, and a 100-dimensional feature vector is obtained using global maximum pooling. The final layers are fully connected layers with size 20 and a softmax layer. We applied batch normalization [17] to the output of each convolution layer. The CNN model uses the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The model is trained for 10 epochs with a batch size of 64 and constant learning rate of 0.001. The total number of parameters of the CNN model is 20K, which takes 40 KB of disk space.

The CNN-based spectrogram detector requires 3 ms of calculation (352 fps) on a Huawei P30 device with HiSilicon Kirin 980 CPU, and the runtime memory footprint of the model is approximately 68 KB.

# 5 STUDY 1: ALGORITHM EVALUATION

We collected three audio data sets in order to facilitate training and evaluation of ProxiMic. They are the voice commands data set $\mathbb{D}_1$, the comprehensive environmental background noise data set $\mathbb{D}_2$, and the extended everyday sound data set $\mathbb{D}_3$.

## 5.1 Participants

We recruited 102 participants for creating data set $\mathbb{D}_1$. 56 participants were female and 46 were male, and the mean age was 25.9 years old (SD = 8.1), ranging from 13 to 50. All participants speak Mandarin Chinese fluently and regularly use smartphones. For obtaining data sets $\mathbb{D}_2$ and $\mathbb{D}_3$, we did not recruit participants and recorded it ourselves.

## 5.2 Apparatus

For creating data set $\mathbb{D}_1$, We asked the participants to bring their personal smartphones to the study. In total, the users brought 55 different phone models from various manufacturers. With the participants' permission, we recorded all audio samples using their smartphones' built-in sound recording application. All applications recorded stereo (dual-channel) raw audio signals without additional speech enhancement algorithms, such as multi-channel beamforming, noise suppression as so on. We extracted the mono (single-channel) raw signal from the main microphone, which was located on the bottom of all smartphones. We normalized the gain difference between devices for both training and testing, which ensures the direct superposition of noise and speech is correct.

We created a voice interaction command corpus by referring to and summarizing the recommended command list of common voice assistants. In total, our corpus included 476 sentences of 25 types of interactive commands. Because of the participant demographics, the corpus was designed in Mandarin Chinese.

For creating data set $\mathbb{D}_2$ and $\mathbb{D}_3$, we use Huawei P30, Honor V20 smartphone to record the background noise audio and use UMIK-1 microphone to record the environmental noise level (dB) of different scenes.

## 5.3 Design and Procedure

For data set $\mathbb{D}_1$, we included close-to-mic speech at different loudness and close-to-mic whispers. Additionally, we included farther-away but louder speech, which could be confused as close-to-mic speech based on amplitudes alone.

We conducted our collection of data set $\mathbb{D}_1$ in an acoustically quiet lab setting. For each participant, we randomly shuffled the corpus, and then divided it into four parts, each containing 119 sentences. The experiment consisted four sessions corresponding to the four parts. Each session was randomly assigned one of the following four conditions: *within 5 cm-loud*, *within 5 cm-soft*, *within 5 cm-whisper*, and *30 cm loud*. For *within 5 cm* condition, we asked participants to keep their mouth within 5 cm of the microphone. *loud* means speaking at a normal and comfortable volume, *soft* means deliberately lowering the volume and speaking softly and quietly (vocal-fold vibration), and *whisper* means keeping their vocal-fold silent (not vibrating and only airflow). For the *30 cm* condition, we asked the participant to keep the microphone at least 30 cm away from their mouth and to record normal and loud voices. Out

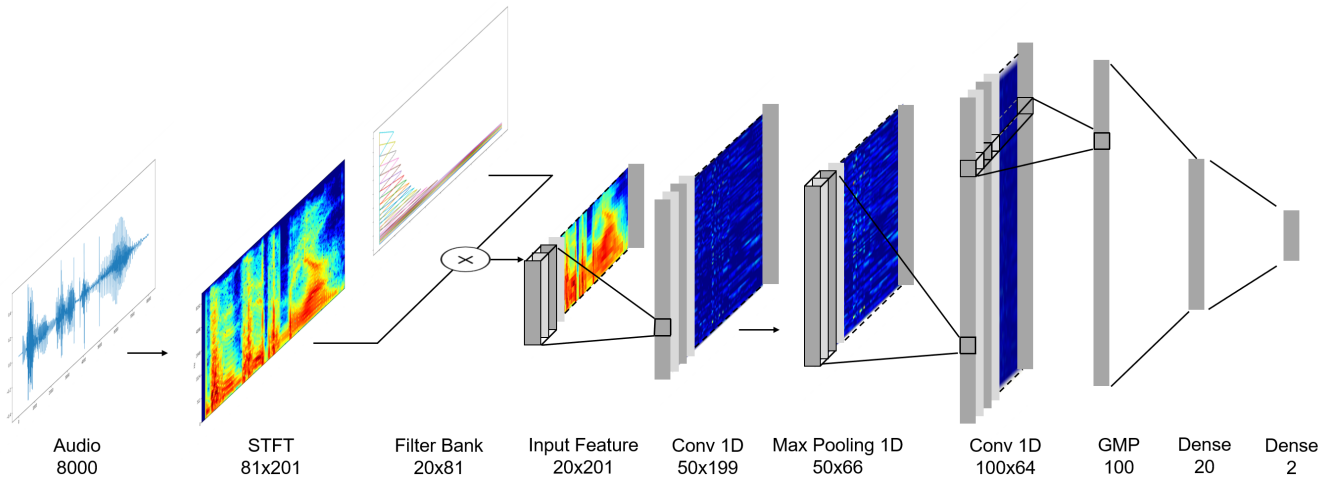


**Figure 6: Structure of CNN-based spectrogram detector.**

of the 119 sentences of *30 cm* condition, we specifically asked the participant to shout at the same volume as "say hello to friends 20 meters away" for 20 of the sentences, chosen randomly. Each session lasted around 15 minutes, and participants were allowed to rest between the sessions. The experiment lasted about 70 minutes in total. Data set $\mathbb{D}_1$, collected using the above protocol, contains 48552 Mandarin Chinese sentences, totaling 43.5 hours. The mean of recording duration for each command is 2.99 seconds (SD = 0.95 s).

For data set $\mathbb{D}_2$, we recorded background sounds in 49 different environments for model training. These environments include airplanes (78 dB), subways (70 dB), on car (66 dB), canteens (58 dB), square (50 dB), office (45 dB), and so on. Each recording is more than 5 minutes. The total duration for all recordings is 15 hours.

For data set $\mathbb{D}_3$, one of the researchers recorded audio continuously in their everyday life with a Huawei Honor V20 smartphone. The recording application was running in the background and the user uses the smartphone and lives normally, such as eating, watching movies, chat with others, playing games, typing on the keyboard, taking the subway, going to the mall, riding a bicycle, etc.. This data set includes 161 hours of long-term recordings and doesn't include any close-to-mic speech. In the data set, 10.3-hours noise is stronger than 60 dB, 21.7-hours noise is stronger than 50 dB. We use data set $\mathbb{D}_3$ to evaluate the False Accepts per Week per User (FAWU) of the two-stage algorithm in real life.

## 5.4 Model Training

Since the background noise and the user's voice are usually additionally superimposed, for model training, we superimposed samples from data sets $\mathbb{D}_1$ and $\mathbb{D}_2$ to acquire a larger data augmentation space. We randomly superimposed speech ($\mathbb{D}_1$) and environmental background sounds ($\mathbb{D}_2$) segments through a Poisson process with $\lambda = 0.1$. To augment training data set, we modulated the amplitude of each sound segment from the data sets by a random amplification factor of between 0.5–2.0 before superimposing. The episode was divided into one-second frames for model training. The frame is marked as a positive example if and only if it contained more than 0.5 seconds of close-to-mic speech signal.

We randomly selected 25% of the data as a verification set, and performed 4-fold cross validation. Data from any same user or same environment will not appear in both the training set and the verification set. We used the same protocol for the rest of this Evaluation section.

## 5.5 Real-world Performance Testing

In order to fully evaluate the contribution of each module of the algorithm, we split the algorithm into four parts: (1) Only use AATT without low-pass filter and CNN (AATT only), (2) AATT using low-pass filtering without CNN (Lowpass+AATT), (3) Only use CNN without AATT (CNN only), (4) Complete two-stage algorithm (Lowpass+AATT+CNN). We choose four scenarios: quiet office (45 dB), noisy canteen (58 dB), subway (70 dB) and daily recording (161 hours data set $\mathbb{D}_3$) to evaluate the performance of the two-stage algorithm in specific scenarios and average conditions. For three specific scenarios, we used Huawei P30 to record background noise for 15 minutes each. The SNR is around 37 dB, 24 dB, and 12 dB for the three scenes respectively. We perform the algorithm on the environmental background sound without any close-to-mic speech to calculate FAWU, and add close-to-mic speech ($\mathbb{D}_1$) with equal probability to test the average recall rate. It's worth noting that we estimate FAWU by assuming all times of the week are in the corresponding scenario, but no one can be on the subway 24 hours. FAWU is just a performance indicator in this study.

Results are shown in Table 1. We can find that the quieter the environment, the better the effect of AATT. At the same time, in quiet environment, low-pass filter can effectively reduce the false accepts of AATT. The CNN and AATT present complementary advantages and performs well in the 161 hours daily data set. For relative low recall rate of 78.6% in subway (70 dB), one main reason can be that the quiet speeches in the data set are highly overwhelmed by the environmental noise and hard to recognize. It is reasonable that the volume of voice should be increased appropriately to obtain a better signal-to-noise ratio in the noisy subway.

**Table 1: Recall and FAWU of different settings. '/' means there is no false accepts during the whole episode.**

| | office (45dB) | | canteen (58dB) | | subway (70dB) | | daily (161 hours) | |
|---|---|---|---|---|---|---|---|---|
| | Recall | FAWU | Recall | FAWU | Recall | FAWU | Recall | FAWU |
| AATT only | 100% | 1468.9 | 100% | 9064.0 | 100% | 28579.4 | 99.3% | 2780.1 |
| Lowpass+AATT | 100% | / | 100% | 3296.0 | 99.5% | 70362.1 | 99.4% | 1880.4 |
| CNN only | 98.6% | 3930.5 | 96.2% | 604.6 | 78.9% | 241.8 | 94.7% | 860.4 |
| Lowpass+AATT+CNN | 98.6% | / | 96.2% | / | 78.6% | / | 94.1% | 12.3 |

## 6 STUDY 2: UNDERSTANDING PROXIMIC

In order to understand the performance and boundaries of ProxiMic, we conducted four analysis: (1) white-box analysis of CNN, (2) device variation test, (3) automatic speech recognition (ASR) accuracy test with close-to-mic speech, (4) privacy of ProxiMic.

### 6.1 Interpretability of the CNN

In order to verify whether the CNN really captured the pop-noise and other subtle close-to-mic features which humans can distinguish clearly, we collected an outdoor recording and calculated the Saliency Map[40] and Occlusion Sensitivity Map[51]. As shown in Figure 7, the recording contains four Chinese words (sounds like [ˈtiænˈtʃiːˈtiænˈtʃiː]) means "weather, weather". The first two words are recorded at 30 cm (without pop-noise). The last two words are recorded at 2 cm (with pop-noise at 0.6s and 0.8s).

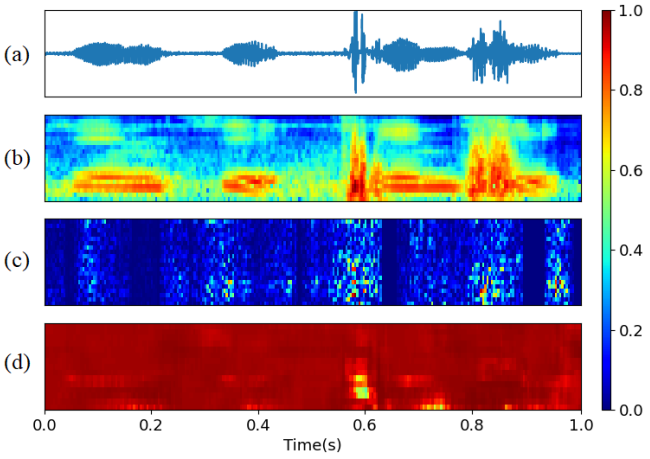


**Figure 7: Explainable analysis of CNN. (a) Original signal. (b) The input of CNN. (c) Saliency Map of CNN input. (d) Occlusion Sensitivity Map of CNN input.**

Sensitivity Map responds to the areas that CNN is most concerned about. Figure 7 shows that the most sensitive part of CNN is the pop-noise part of the third and fourth words (2 cm). There are also some bright spots at the beginning of the second word and the first word (30 cm), which means that if there is a pop-noise shape (similar to the beginning of the third word), the probability of activation will increase significantly, in other words, high-energy noise at other locations will not significantly increase the probability of activation. In other words, it is effective for only pop-noise and consonants to occur simultaneously.

For the Occlusion Sensitivity Map, we use a $5 \times 10$ gray rectangle to cover a part of the spectrogram input and calculate the activation probability. We can find from Figure 7 that only when the pop-noise part of the third word is covered, the activation probability is reduced to 45.8%. This is consistent with the intuition and training data that CNN captures pop-noise and activates only if there are close-to-mic features longer than around 0.5 seconds.

### 6.2 Performance Analysis on Different Form Factors

In Study 1, we evaluated 55 types of ECM and MEMS microphones of smartphone, and the result shows that ProxiMic is robust for smartphone. In order to clarify the impact of more types of hardware packaging on ProxiMic and the generalization ability of CNN, we conducted this multi-device study. Unlike the smartphone microphone which has a wind tunnel with a diameter of about 1 mm, we found three other types of devices for evaluation. (1) TicWatch2 smart watch, which has a microphone on the side and has a similar air duct structure. (2) B&O H6 wired headphone. Its microphone is on the headphone cable and is completely wrapped by a plastic shell. (3) Aigo R6625 recording pen, which has an ECM with a diameter of about 7 mm at the top and is wrapped by a 5 mm thick windproof sponge.

**Table 2: Recall rate on different devices**

| Devices | phone | watch | headphone | pen |
|---|---|---|---|---|
| Recall | 98.6% | 98.9% | 78.6% | 85.1% |

We collected 119 close-to-mic speech (within 5 cm) from smartphones, smart watch, headphones and recorder pen in quiet environment respectively. Table 2 shows the four recall rate of devices. The recall of the smartphone and the smartwatch is similar because of the same structure. The result of the watch indicates that for any wearable device in the future, as long as the microphone structure is the same as the smartphone, it can deploy ProxiMic directly. The headphone's microphone and voice recorder pen filtered out around half of the pop noise. However, the vibration noise and slight airflow caused by wind impacting the plastic shell are still obvious in the sense of hearing. Although the structure and characteristics are different, the model trained on the smartphone data sets also shows a strong generalization ability for headphones and voice recorder pens. We believe that collecting more data for special device structures can effectively solve the problem of low recall.

### 6.3 ASR Accuracy

Automatic Speech Recognition (ASR), also known as Speech To Text (STT), is a technology that can transform the one-dimensional

speech input into a series of word tokens, and it is an important task for realizing voice-based natural language understanding and human-computer interaction. In order to understand the impact of pop-noise on voice quality, testing the translation accuracy of ASR is an intuitive method. In this study, we used the Baidu Phrase Recognition Standard Edition interface [4] to perform ASR on a total of 48552 pieces of voice commands from study 1 (data set $\mathbb{D}_1$ without background noise), and calculated the Word Error Rates (WER) on the ASR results (Table 3).

**Table 3: Word error rate of 48552 sentences**

| Scenes | 5cm loud | 5cm soft | 5cm whisper | 30cm loud |
|---|---|---|---|---|
| WER | 2.10% | 2.24% | 7.83% | 2.30% |

Baidu ASR is one of the well-known state-of-the-art ASR systems, which shows the leading level of ASR technology in Mandarin Chinese. We can see that the Word Error Rate (WER) of "within 5 cm soft", "within 5 cm loud" and "30 cm loud" are similar and excellent. This result shows that the strong pop-noise has little effect on the machine's understanding of human speech. In addition, thanks to technological progress, within 5 cm whispering can also be accurately translated by state-of-the-art ASR. The WER of within 5 cm whispering can meet the requirements of the scene of daily conversation with the voice assistant. This allows us to use whispering with ProxiMic to protect privacy.

## 6.4 Privacy Study

The close-to-mic speech brings a high Signal-to-Noise Ratio (SNR), which allows the user to talk to the device at a much lower volume than normal. In this special interactive scenario, we try to understand how much the user's voice input would be eavesdropped. We recruited 6 participants from the campus, ask them to talk to ProxiMic on smartphone in quiet offices (~40 dB), cafes (~50 dB), and canteens (~60 dB) respectively. We let everyone input voice commands from study 1 (data set $\mathbb{D}_1$) to smartphone in a comfortable small voice, and try to avoid being heard by others. When one person speaks, others sit on chairs at different distances, eavesdrop carefully and write down what they hear. We use the iFlytek ASR system, one of the well-known state-of-the-art ASR systems for Mandarin Chinese, which is also integrated into the ProxiMic Android application, to convert user's speech into text. When the speaker's words are not completely understood by smartphone, the speaker is asked to speak again until the ASR is completely accurate. So the WER of ASR is 0% for each sentence. Each user speaks 50 different sentences in each scene, and the result is shown in Figure 8 and Figure 9.

Because ProxiMic supports whispering, almost all voice commands are made in whispering way. This makes all sentences almost impossible to understand at 1 meter, and familiar words are only vague guesses. The users said that close-to-mic whispering is completely acceptable in a slightly noisy environment or when the words spoken are not so private.

While the user is speaking, we additionally set up a UMIK-1 measuring microphone 50 cm in front of the speaker and recorded the whole process. After the experiment, we played back what the user said, and we couldn't hear any user's whispering voice from the microphone recording. This shows that ProxiMic also has a good defense effect against ordinary microphone eavesdropping.

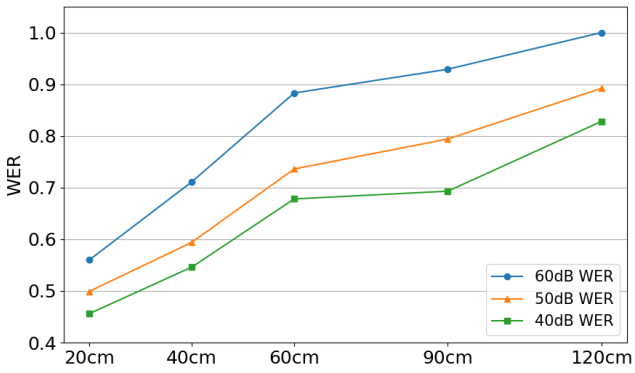


**Figure 8: Word Error Rate (WER) of eavesdroppers in different distances and scenes.**

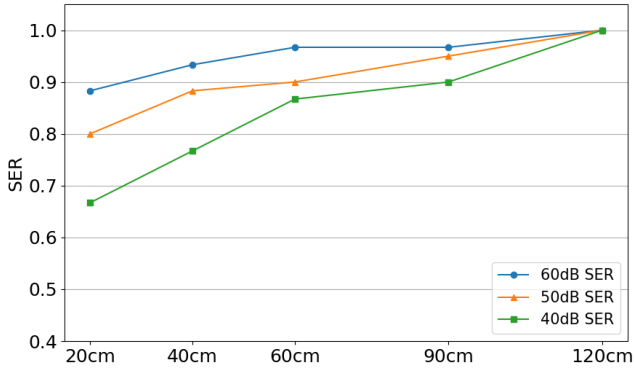


**Figure 9: Sentence Error Rate (SER) of eavesdroppers in different distances and scenes.**

# 7 STUDY 3: USABILITY STUDY OF PROXIMIC

In this study, we compare ProxiMic to other activation methods to evaluate its efficiency and usability. We choose three widely-used activation methods of wake-up phrase (Keywords), virtual key press (GUI) and physical key press (Button) for comparison. We recorded *Activation Time* for each task, and asked the participants to provide subjective ratings for each task.

## 7.1 Activation Methods

We compared the following four activation methods for voice input, which are the main methods to activate voice input on various devices.

- **ProxiMic**: The participant brings the device to the mouth to activate voice input. The algorithm on the devices recognize the close-to-mic speech locally.
- **Keywords**: The participant speaks "XiaoYi-XiaoYi" to trigger the activation. The wake-up phrase is detected by the built-in keyword spotting algorithm of the device.

- **GUI**: The participant finds the voice assistant application and presses the virtual key on the touchscreen as the activation.
- **Button**: The participant presses the physical button on the side of the device for one second as the activation.

## 7.2 Participants

We recruited 6 participants (2 female and 4 male; mean age = 43.3, SD = 16.48) in this experiment. Five of them had prior experience of voice input on smartphones.

## 7.3 Apparatus

We developed a voice assistant applications on smartphone to apply ProxiMic. We used the following devices for this study: a Huawei P30 with a power button located on the right, and a main microphone on the bottom; a Mobvoi TicWatch 2 with a microphone on the side pointing to the palm and a button next to the microphone to trigger the built-in voice assistant; a B&O H6 headphone (connected to the Huawei P30) with one microphone on the headphone cable, which can be easily re-positioned by hand to within 5 cm in front of the mouth. When connected to the headphone, the smart phone switches to use voice input from the headphone.

## 7.4 Design

We conducted a within-subject study with two independent factors as *Activation Method* and *Device Type*. We tested four activation methods on the smartphone to compare the performance of ProxiMic with the widely-used baselines. In addition, we evaluated ProxiMic on two more devices to test the influence of device difference. Using one activation method on one device, the participant completed 15 rounds of voice input tasks as one session. The order of *Activation Method* and *Device Type* were randomized for each participant. In total, each participant performed 6 sessions $\times$ 15 tasks = 90 sentences input tasks.

We chose the widely-used voice commands as the voice input task in each session. We counted the time duration between when the test started and when the first word of the sentence was spoken by the participant as the efficiency metric. We asked participants to fill in the NASA-TLX [15] questionnaire and answer 6 supplementary questions in seven-point Likert scale as the metrics to evaluate user experience.

## 7.5 Procedure

The experimenter first introduced the voice input task and activation methods to the participants. Two minutes was given to participants to practice and familiarize themselves with the task and activation methods. Then, participants completed 6 sessions of voice input. They were asked to perform the inputs as fast as possible. For each task, we asked the participants to hold the microphone to around 2 cm from their mouth, and rest the arm on a desk after finishing each task. Participants were asked to repeat a task if the voice input was not successful. We only record the time of each successful operation. After a session, each participant was allowed a 2-minute break. Finally, we asked participants to completed a NASA-TLX questionnaire with 6 supplementary questions. The experiment lasted around 30 minutes for each participant.

## 7.6 Results

We ran RM-ANOVA on the activation time with post-hoc T-tests, Friedman tests on subjective scores with post-hoc Wilcoxon signed rank tests.

*7.6.1 Activation Efficiency.* RM-ANOVA results showed significant effects of both *Activation Method* ($F_{3,15} = 25.633, p < .001$) and *Device Type* ($F_{2,10} = 7.339, p = .01$) on *Activation Time*. Post-hoc tests showed that ProxiMic (mean=1.14, SD=.08) was significantly faster than all other methods, outperforming Keywords by 51.7% (p<.05), GUI by 61.5% (p<.01), Button by 42.4% (p<.001) (Table 4). For *Device Types*, post-hoc tests found significant difference between smart watch and smart phone (p<.05), with the former 29.8% faster (Table 5).

**Table 4: *Activation Time* (second) for different *Methods***

| | Keywords | GUI | Button | ProxiMic |
|---|---|---|---|---|
| mean | 2.36 | 2.96 | 1.98 | 1.14 |
| SD | 0.26 | 0.21 | 0.10 | 0.08 |

**Table 5: *Activation Time* (second) for different *Device Types***

| | Phone | Watch | Headphone |
|---|---|---|---|
| mean | 1.14 | 0.80 | 1.09 |
| SD | 0.08 | 0.04 | 0.08 |

*7.6.2 User Experience.* The overall results are shown in Figure 10. Friedman tests show that *Activation Method* makes a significant influence on all index ($p < .05$). Post-hoc tests show that ProxiMic provides significantly lower effort, easier to use, more privacy activation than Keywords; lower mental demand, lower temporal demand, more overall performance,lower effort, easier to use, broader applications and more want to use activation than GUI; broader applications activation than Button ($p < .05$ in all cases). ProxiMic also provides significantly lower frustration, easier to continuous input then other three methods ($p < .05$).

In the experiment, three participants (P2, P5, P6) reported that "Wake-up phrase(s) were difficult to recognize", P5 reported that "Touching the ear and pulling the headphone line is very easy, fast and comfortable". These comments are consistent with users' subjective feedback that ProxiMic is efficient, user-friendly and practical.

# 8 DISCUSSION

We discuss opportunities and application scenarios of ProxiMic, as well as limitations and future work.

## 8.1 Interaction Enabled by Whisper Detection

Since ProxiMic supports whispering — a tone that is only perceptible when speaking within close proximity to the microphone — we attempted to further distinguish whispering and normal speaking, and design separate interactive feedback. Whispering may suggest that a user is in a situation where speaking loudly is inappropriate, or that the voice data are privacy-sensitive. Therefore, voice-based

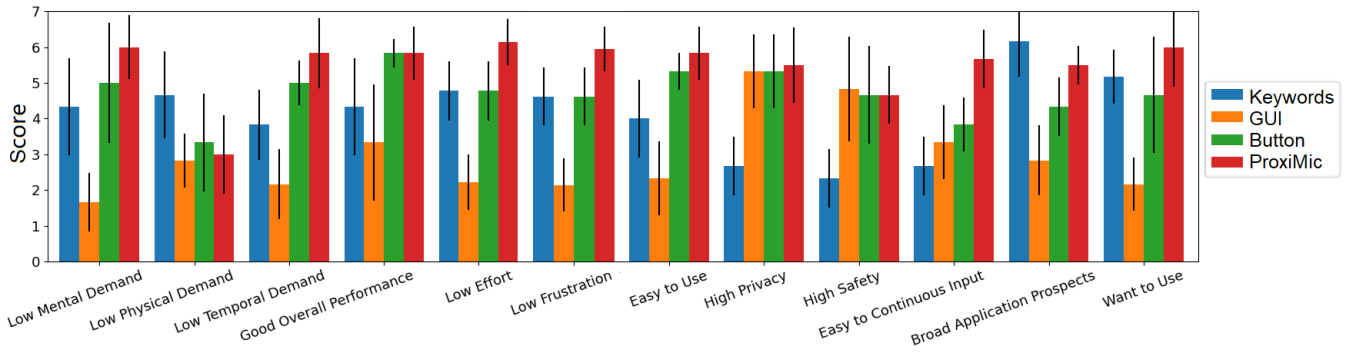


**Figure 10: User's subjective feedback on the four ways. The score range is from 1 to 7, and 7 represents the most positive evaluation. The standard deviation is marked in the figure.**

systems can change how they provide feedback when a whisper is detected, e.g., responding in text or vibration only. We employed the ProxiMic CNN structure to distinguish between sound from the *within 5 cm whisper* and the *within 5 cm soft/loud* data sets from study 1. After retraining on the two data sets, the ProxiMic CNN achieved a classification accuracy of 95.7% without modifying the model design. This demonstrates that the structure of our CNN is extensible and can be easily extended to more categories. This shows that it is promising and feasible to realize tone-context wake-up for ProxiMic. The user study of tone-context-based feedback in the close-to-mic scenario can be used as one of the directions for future work.

## 8.2 Language dependence and Robustness of ProxiMic

We mainly utilize pop noise to recognize close-to-mic speech. Pop noise is a common feature in the words in different languages. We take English words as an example. Most words containing "b,c,d,f,j,k,l,p,q,r,s,t,v,w,x,y,z" generate clear "pop noise" airflow. We invited 4 additional users to read random English sentences from "the MacKenzie phrase set" [24] (50 sentences per user in quiet scenario). 96% of sentences are recognized as close-to-mic speech by ProxiMic (with Chinese training corpus). However, the exact proportion of the words that can trigger Proximic is different in different languages, and can be used as future work. In addition, the feature of pop noise can prevent voice attacks to a certain extent (as discussed in Shiota et al.'s work[36, 37]). To show our model can be applied to deter direct replay attacks, we tested ProxiMic's ability in identifying replayed voice recordings with strong pop-noise from a speaker. Results show that 99.92% of the replayed frames were rejected by our CNN because the vibration caused by the speaker and the resulting impact of the airflow are vastly different. The two-step algorithm using sound amplitude and spectrogram characteristics shows a certain degree of attack prevention performance. This can be an interesting direction for future work.

## 8.3 Potential Applications of ProxiMic

We believe that the ability of recognizing close-to-mic speech has broad application potential. In recent years, with the development of Internet of Things and ubiquitous computing, we can see the improvement of sensing capabilities of various devices. PenSight [25] demonstrates the great potential of gesture interaction for digital tablet pen. If we add an additional microphone to the top of the pen, we can potentially enhance gesture input with the rich semantics of voice input. Ubiquitous computing also extends the boundaries of smart terminals such as smartphone and smart TV. Consider a button with an embedded microphone pinned on the neckline: a user can simply lower their head to speaker into the device, which can as an input of other smart devices. In addition, ProxiMic can turn most of the existing handheld and wearable voice input devices into personal voice input devices: by recognizing close-to-mic speech, we have the opportunity of separating the voice of the device holder (with pop noise) from the voice of others (without pop noise).

## 8.4 Subtle Close-to-Mic Features

Our two-step algorithm is designed according to the characteristics of pop noise, and we found that CNN indeed responds to the frequency spectrogram of pop noise most of the time. But for some words without obvious pop noise, ProxiMic can still partly distinguish whether it is close-to-mic speach. Although the CNN is hard to be explained, we think that the reason can be other subtle close-to-mic features which we didn't use explicitly. We decided to list the noteworthy subtle close-to-mic features which can be noticed by the human ear in the experiment here. (1) With close-to-mic speech, the sound of air rubbing against the mouth is faintly audible. This sound is similar to the high-frequency part of whispering which will be significantly reduced as the distance becomes longer. (2) When indoors, due to the presence of wall reflections, farther-away sounds (over 30 cm) will be superimposed with muddy reverberation (often perceived as a sense of space). This direct-to-reverberant ratio may be used as a potential feature to distinguish close-to-mic speech in indoor environments. (3) The difference in volume between far and near is more significant for outdoor environments because there is no indoor reflection. (4) In addition, the timbre and volume of speech also have a certain correlation, which means that if a person speaks in a low voice, but has a huge volume received by microphone, it should be considered as a potential close-to-mic speech. We expect that in addition to CNN's automatic feature extraction, these subtle features can be effectively used in future algorithms.

## 8.5 Limitation and Future Work

In addition to the future work mentioned above, there are some limitations of ProxiMic which can be improved in future work.

First, we use smartphone microphones (ECM or MEMS) for training and testing. Although the multi-device performance is acceptable, we think that more data and testing in more package types are necessary for real-world applications.

Second, although result shows that the privacy of ProxiMic is acceptable, a general ASR system was used in our privacy study. In the future, we can try to train a domain specific ASR system used to recognizing within 5 cm whispering which can effectively utilize pop-noise and exhaled airflow. Analogous to Fukumoto's ingressive-speech based SilentVoice [10], we can try to use ProxiMic to provide ultra-small volume voice input for smartphones and smartwatches.

Third, the pop-noise-based detector relies on the user talking direct into the microphone and the device can effectively capture pop noise (e.g., no mechanical pop filter). In this case, using more features or hardware to designing a close-to-mic detector which does not completely rely on pop noise is a future work.

Finally, semantic-based off-device False Trigger Mitigation (FTM) systems can be used to further reduce the FAWU. In the future, we can try to integrate the FTM system into ProxiMic to understand how much false alarms from ProxiMic can be solved by FTM system.

## 9 CONCLUSION

ProxiMic is a novel sensing technique that uses pop noise and close-to-mic features to activate voice input by a single microphone. The evaluation results show that ProxiMic has reached 94.1% activate recall, 12.3 FAWU with 68 KB memory size, which can run at 352 fps on smartphone. ProxiMic can be used in a variety of devices such as mobile phones, watches, headphones, etc. Experiments show that ProxiMic is robust to the environment and devices, and will not affect the performance of subsequent algorithms such as ASR. Users agree that ProxiMic is efficient, user-friendly and practical. With the continuous popularity of voice input devices, we expect that ProxiMic can potentially be applied in a wide variety of use cases.

## ACKNOWLEDGMENTS

This work is supported by National Key Research and Development Plan under Grant No. 2016YFB1001200 and No. 2019AAA0105200, the Natural Science Foundation of China under Grant No. 61672314, and also by Beijing Key Lab of Networked Multimedia, the Institute for Guo Qiang of Tsinghua University, Institute for Artificial Intelligence of Tsinghua University (THUAI), and Beijing Academy of Artificial Intelligence (BAAI).

## REFERENCES

[1] Apple. 2020. Use Siri on all your Apple devices - Apple Support. Website. https://support.apple.com/en-us/HT204389#apple-watch.
[2] Sylvain Argentieri, Patrick Danes, and Philippe Souères. 2015. A survey on sound source localization in robotics: From binaural to array processing methods. *Computer Speech & Language* 34, 1 (2015), 87–112.
[3] S Argentieri, A Portello, M Bernard, P Danes, and B Gas. 2013. Binaural systems in robotics. In *The technology of binaural listening*. Springer-Verlag, Berlin, Germany, 225–253.
[4] Baidu. 2020. Baidu ASR. http://ai.baidu.com/tech/speech/asr.
[5] Jonathan S Brumberg, Alfonso Nieto-Castanon, Philip R Kennedy, and Frank H Guenther. 2010. Brain–computer interfaces for speech communication. *Speech communication* 52, 4 (2010), 367–379.
[6] Joe C Chen, Kung Yao, and Ralph E Hudson. 2002. Source localization and beamforming. *IEEE Signal Processing Magazine* 19, 2 (2002), 30–39.
[7] Yunbin Deng, James T. Heaton, and Geoffrey S. Meltzner. 2014. Towards a practical silent speech recognition system. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie (Eds.). ISCA, Baixas, France, 1164–1168. http://www.isca-speech.org/archive/interspeech_2014/i14_1164.html
[8] Aarthi Easwara Moorthy and Kim-Phuong L Vu. 2015. Privacy concerns for use of voice activated personal assistant in the public space. *International Journal of Human-Computer Interaction* 31, 4 (2015), 307–335.
[9] Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. 2015. Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters* 65 (2015), 22–28.
[10] Masaaki Fukumoto. 2018. SilentVoice: Unnoticeable Voice Input by Ingressive Speech. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. ACM, New York, NY, USA, 237–246. https://doi.org/10.1145/3242587.3242603
[11] Yang Gao, Yincheng Jin, Jiyang Li, Seokmin Choi, and Zhanpeng Jin. 2020. EchoWhisper: Exploring an Acoustic-based Silent Speech Interface for Smartphone Users. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–27.
[12] Fengpei Ge and Yonghong Yan. 2017. Deep neural network based wake-up-word speech recognition with two-stage detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Piscataway, NJ, USA, 2761–2765. https://doi.org/10.1109/ICASSP.2017.7952659
[13] Eleftheria Georganti, Tobias May, Steven van de Par, Aki Harma, and John Mourjopoulos. 2011. Speaker distance detection using a single microphone. *IEEE transactions on audio, speech, and language processing* 19, 7 (2011), 1949–1961.
[14] Eleftheria Georganti, Tobias May, Steven Van De Par, and John Mourjopoulos. 2013. Sound source distance estimation in rooms based on statistical properties of binaural signals. *IEEE transactions on audio, speech, and language processing* 21, 8 (2013), 1727–1741.
[15] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index) : Results of Empirical and Theoretical Research. *Advances in Psychology* 52, 6 (1988), 139–183.
[16] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. 2017. CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*. IEEE, Piscataway, NJ, USA, 131–135. https://doi.org/10.1109/ICASSP.2017.7952132
[17] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 448–456. http://proceedings.mlr.press/v37/ioffe15.html
[18] Arnav Kapur, Shreyas Kapur, and Pattie Maes. 2018. AlterEgo: A Personalized Wearable Silent Speech Interface. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. ACM, New York, NY, USA, 43–53. https://doi.org/10.1145/3172944.3172977
[19] Naoki Kimura, Michinari Kono, and Jun Rekimoto. 2019. SottoVoce: An Ultrasound Imaging-Based Silent Speech Interaction Using Deep Neural Networks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300376
[20] BRET KINSELLA. 2018. New Report: Over 1 Billion Devices Provide Voice Assistant Access Today and Highest Usage is on Smartphones. Website.
[21] Charles Knapp and Glifford Carter. 1976. The generalized correlation method for estimation of time delay. *IEEE transactions on acoustics, speech, and signal processing* 24, 4 (1976), 320–327.
[22] Kenichi Kumatani, Sankaran Panchapagesan, Minhua Wu, Minjae Kim, Nikko Strom, Gautam Tiwari, and Arindam Mandai. 2017. Direct modeling of raw audio with DNNs for wake word detection. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, Piscataway, NJ, USA, 252–257.
[23] Gustavo López, Luis Quesada, and Luis A Guerrero. 2017. Alexa vs. Siri vs. Cortana vs. Google Assistant: a comparison of speech-based natural user interfaces. In *International Conference on Applied Human Factors and Ergonomics*. Springer-Verlag, Cham, Switzerland, 241–250.
[24] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI EA '03)*. ACM, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971
[25] Fabrice Matulic, Riku Arakawa, Brian Vogel, and Daniel Vogel. 2020. PenSight: Enhanced Interaction with a Pen-Top Camera. In *Proceedings of the 2020 CHI*

*Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376147

[26] Donald McMillan, Barry Brown, Ikkaku Kawaguchi, Razan Jaber, Jordi Solsona Belenguer, and Hideaki Kuzuoka. 2019. Designing with Gaze: Tama–a Gaze Activated Smart-Speaker. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–26.

[27] Raveesh Meena, José Lopes, Gabriel Skantze, and Joakim Gustafson. 2015. Automatic Detection of Miscommunication in Spoken Dialogue Systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Prague, Czech Republic, 354–363. https://doi.org/10.18653/v1/W15-4647

[28] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. 2010. Acoustic event detection in real life recordings. In *2010 18th European Signal Processing Conference*. IEEE, Piscataway, NJ, USA, 1267–1271.

[29] Beomjun Min, Jongin Kim, Hyeong-Jun Park, and Boreom Lee. 2016. Vowel Imagery Decoding toward Silent Speech BCI Using Extreme Learning Machine with Electroencephalogram. *BioMed Research International* 2016 (01 2016), 1–11. https://doi.org/10.1155/2016/2618265

[30] Martin Porcheron, Joel E. Fischer, Stuart Reeves, and Sarah Sharples. 2018. Voice Interfaces in Everyday Life. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174214

[31] François Portet, Michel Vacher, Caroline Golanski, Camille Roux, and Brigitte Meillon. 2013. Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects. *Personal and Ubiquitous Computing* 17, 1 (2013), 127–144. https://doi.org/10.1007/s00779-011-0470-5

[32] Javier Ramırez, José C Segura, Carmen Benıtez, Angel De La Torre, and Antonio Rubio. 2004. Efficient voice activity detection algorithms using long-term speech information. *Speech communication* 42, 3-4 (2004), 271–287.

[33] Florian Roider, Lars Reisig, and Tom Gross. 2018. Just Look: The Benefits of Gaze-Activated Voice Input in the Car. In *Adjunct Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Toronto, ON, Canada) *(AutomotiveUI '18)*. ACM, New York, NY, USA, 210–214. https://doi.org/10.1145/3239092.3265968

[34] Richard Roy and Thomas Kailath. 1989. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing* 37, 7 (1989), 984–995.

[35] R. Schmidt. 1986. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation* 34, 3 (1986), 276–280.

[36] Sayaka Shiota, Fernando Villavicencio, Junichi Yamagishi, Nobutaka Ono, Isao Echizen, and Tomoko Matsui. 2015. Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, Baixas, France, 239–243. http://www.isca-speech.org/archive/interspeech_2015/i15_0239.html

[37] Sayaka Shiota, Fernando Villavicencio, Junichi Yamagishi, Nobutaka Ono, Isao Echizen, and Tomoko Matsui. 2016. Voice Liveness Detection for Speaker Verification based on a Tandem Single/Double-channel Pop Noise Detector. In *Odyssey 2016: The Speaker and Language Recognition Workshop, Bilbao, Spain, June 21-24, 2016*, Luis Javier Rodríguez-Fuentes and Eduardo Lleida (Eds.). ISCA, Baixas, France, 259–263. https://doi.org/10.21437/Odyssey.2016-37

[38] ShotSpotter. 2020. ShotSpotter. https://www.shotspotter.com/.

[39] Siddharth Sigtia, Rob Haynes, Hywel Richards, Erik Marchi, and John Bridle. 2018. Efficient Voice Trigger Detection for Low Resource Hardware. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana (Ed.). ISCA, Baixas, France, 2092–2096. https://doi.org/10.21437/Interspeech.2018-2204

[40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *2nd International Conference on Learning Representations, Workshop Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). ICLR 2014, Banff, Canada, 1–8. http://arxiv.org/abs/1312.6034

[41] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. 1999. A statistical model-based voice activity detection. *IEEE signal processing letters* 6, 1 (1999), 1–3.

[42] Ke Sun, Chun Yu, Weinan Shi, Lan Liu, and Yuanchun Shi. 2018. Lip-interact: Improving mobile device interaction with silent speech commands. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 581–593.

[43] S Gökhun Tanyer and Hamza Ozer. 2000. Voice activity detection in nonstationary noise. *IEEE Transactions on speech and audio processing* 8, 4 (2000), 478–482.

[44] Jean-Marc Valin, François Michaud, and Jean Rouat. 2007. Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Robotics and Autonomous Systems* 55, 3 (2007), 216–228.

[45] Barry D Van Veen and Kevin M Buckley. 1988. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine* 5, 2 (1988), 4–24.

[46] Yueting Weng, Chun Yu, Yingtian Shi, Yuhang Zhao, Yukang Yang, and Yuanchun Shi. 2021. FaceSight: Enabling Hand-to-Face Gesture Interaction on AR Glasses with a Downward-Facing Camera Vision. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Tokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3411764.3445484

[47] Yukang Yan, Chun Yu, Yingtian Shi, and Minxing Xie. 2019. PrivateTalk: Activating Voice Input with Hand-On-Mouth Gesture Detected by Bluetooth Earphones. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. ACM, New York, NY, USA, 1013–1020. https://doi.org/10.1145/3332165.3347950

[48] Yukang Yan, Chun Yu, Wengrui Zheng, Ruining Tang, Xuhai Xu, and Yuanchun Shi. 2020. FrownOnError: Interrupting Responses from Smart Speakers by Facial Expressions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376810

[49] Zhican Yang, Chun Yu, Fengshi Zheng, and Yuanchun Shi. 2019. ProxiTalk: Activate Speech Input by Bringing Smartphone to the Mouth. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–25.

[50] Andreas Zehetner, Martin Hagmüller, and Franz Pernkopf. 2014. Wake-up-word spotting for mobile systems. In *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, Piscataway, NJ, USA, 1472–1476.

[51] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 8689)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer-Verlag, Cham, Switzerland, 818–833. https://doi.org/10.1007/978-3-319-10590-1_53

[52] Shiwen Zhao, Brandt Westing, Shawn Scully, Heri Nieto, Roman Holenstein, Minwoo Jeong, Krishna Sridhar, Brandon Newendorp, Mike Bastian, Sethu Raman, Tim Paek, Kevin Lynch, and Carlos Guestrin. 2019. Raise to Speak: An Accurate, Low-Power Detector for Activating Voice Assistants on Smartwatches. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. ACM, New York, NY, USA, 2736–2744. https://doi.org/10.1145/3292500.3330761