WhisperBuds: Whispered Speech Input With Earbuds

ANONYMOUS AUTHOR(S)*

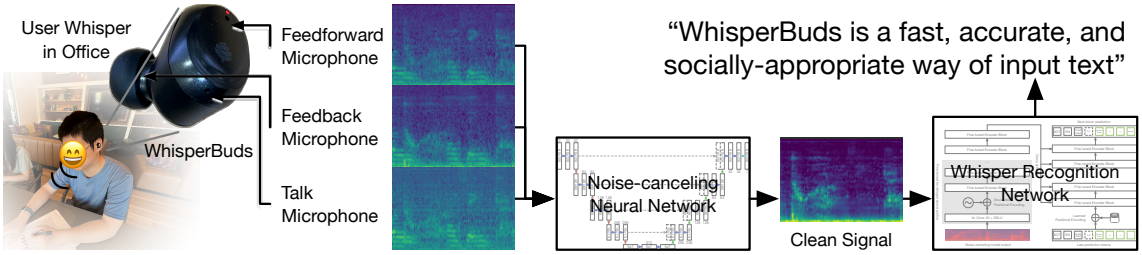

Fig. 1. WhisperBuds allow users to efficiently input text without disturbing others in the same office, even quieter than most computer keyboards. It captures signals from the three microphone channels commonly present on commercial earbuds. Then, it uses a noise-canceling neural network to separate the user's whispering from background sound and improve the signal-to-noise ratio. The fine-tuned whisper recognition network takes the clean signal as input and predicts live transcription with minimal latency (1.06s on average) and high accuracy (11.60% word error rate).

With the increased adoption of conversational interfaces such as ChatGPT, improving the bandwidth of text-based communication with computers is essential. We can speak almost three times as fast as we can type. However, typical speech interfaces require users to either speak loudly, disturbing nearby people, or use near-mouth devices, which are socially inappropriate. To overcome these problems, we propose WhisperBuds, a real-time, low-error whisper input system with commercial off-the-shelf earbuds. Existing earbuds are equipped with multiple microphone signals to reduce hearing noise. In contrast, we repurposed the same input for a "noise-canceling" model to recover users' clean whisper signals and a fine-tuned OpenAI Whisper mode (trained with 21.2 hours of whisper data) to recover user's transcript. A user study (n=12) demonstrated that WhisperBuds achieved 88.5% of the input speed and 97% of the accuracy of traditional normal voice speech input while maintaining a noise level lower than most computer keyboards.

CCS Concepts: • **Human-centered computing** → **Mobile devices**; **Sound-based input / output**; **Text input**; • **Computing methodologies** → **Speech recognition**; • **Hardware** → *Noise reduction.*

Additional Key Words and Phrases: wearable devices, voice recognition, noise reduction, earbuds, mobile devices, silent speech input, whisper recognition

ACM Reference Format:

Anonymous Author(s). 2024. WhisperBuds: Whispered Speech Input With Earbuds. In . ACM, New York, NY, USA, 19 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Conversational interfaces like ChatGPT have recently gained popularity due to their natural interaction style and versatility. However, the conventional text-based input format makes it hard to use on the go without access to a physical keyboard. While voice input's faster speed [30] can mitigate that issue, typical voice inputs are loud and

Manuscript submitted to ACM

disturbing to the people around the user, which renders them not usable in quiet or public environments, such as libraries, classrooms, office spaces, and public transport.

One potential solution to this problem is to recognize people's whispered speech. Being soft and quiet, whispers are naturally less disturbing to cohabitants than normal speech. However, prior work on whispered speech detection often requires extra contraptions to be worn around the mouth [8]. These additional devices might look awkward in social settings, thus limiting their adoption. Therefore, we seek to recognize whisper with commonly worn devices that support usual wearable form factors.

We present WhisperBuds, a socially appropriate way of whispered speech interaction using earbuds. WhisperBuds used a commercial off-the-shelf hardware design with customized firmware, driver, and machine learning models to achieve a discreet look and high whisper recognition accuracy.

There are two challenges to the design of WhisperBuds: 1) The low signal-to-noise ratio (SNR) of whisper speech, and 2) The uncommon sound profiles due to the unique location of the sensor and the sound-producing method. We resolved the first challenge by employing a unique neural network for *noise-canceling*. The noise-canceling neural network utilized the microphones commonly found in earbuds to enhance the whisper signal and cancel out environment signals, resulting in a more precise *whisper* capture. For the second challenge, we collected a data set with 65 people and fine-tuned a large speech-to-text model, OpenAI Whisper[1] [26], to achieve a highly accurate and versatile speech recognition model.

We evaluated WhisperBuds by comparing it with mobile keyboard input and normal speech implemented with the same baseline OpenAI Whisper model. We observed that WhisperBuds reach 88.5% of text input speed (word-per-minute) compared to normal speech. We also measured the user's produced noise while using WhisperBuds and found it to be lower than 111 out of 120 keyboards measured by rtings.com.

The contributions of this paper include:

(1) The WhisperBuds introduces a real-time whisper input system with a pair of commercial off-the-shelf earbuds, achieving high accuracy and speed while remaining quiet and non-intrusive.
(2) The system tackles whispered speech's low signal-to-noise ratio and unique sound profiles using a specialized neural network for noise cancellation to enhance speech capture with multichannel audio.
(3) The evaluation of WhisperBuds showcased the feasibility of a whisper speech interface with earbuds, and shed light on multilingual generalizability and speech separatio.

2 RELATED WORK

2.1 Silent Speech Interfaces via Audio

The most relevant related work category is silent speech interfaces by detecting audio signals. One of the earliest works is done by Nakajima et al. [21]. They used a stethoscope microphone stuck on the user's head to record vibration and a hidden Markov model (HMM) to recognize the user's speech. Another intriguing direction is to use a microphone in front of the user's mouth to detect "ingressive speech", which is the sound produced while inhaling [10]. ProxiMic [25] explored using the artifacts produced by a close-to-mouth microphone to trigger voice assistants. SilentSpeech [13] used a similar setup for whispered speech recognition. DualVoice [27] further explored close-to-mouth microphones and used normal and whispered speech as different input modalities. WESPER [28] further increased accuracy by first

[1]Although the base model is called "Whisper", it is not trained with and does not support whispered speech. Our preliminary analysis showed that OpenAI whisper has a word error rate of 60% when used with WhisperBuds's whisper data.

converting the whispered speech captured near the user's mouth to normal speech and then using OpenAI whisper to recognize the transcript.

One problem with close-to-mouth microphones is that they are not socially appropriate in many situations. To solve this, SilentMask [12] proposed a use case where the user wears the microphone within a mask. However, this solution is not generalizable to other situations where the user is not wearing a mask. Like WhisperBuds, EarCommand [15] used an earbuds form factor to capture the user's speech. They made custom hardware containing a speaker mic pair inside the ear canal. They used ultrasonic frequency sweeping and the sound reflected from the ear canal to capture the user's ear canal's deformation and to recognize the user's speech. They have achieved a 10.05% word error rate (WER) on a 32-word vocabulary, which still needs improvement for general-purpose speech recognition with an open vocabulary. Compared with EarCommand, WhisperBuds uses a more common earbuds form factor and uses a more generalizable approach to capture the user's speech.

2.2 Silent Speech Interface via Other Modalities

Previous silent speech interfaces have explored various modalities, many constrained by limited vocabulary and the use of unconventional hardware. The earliest work on silent speech interfaces is done by Schönle et al. [31], which uses a magnet attached to the user's tongue and magnetic field sensors attached to the user's head to detect the tongue's movement. Later work by Fagan et al. [9] further refined the idea and used a system with similar sensing techniques to help patients with laryngectomy. Derma [29] instead used an inertial measurement unit (IMU) attached to the user's neck to detect the user's silent speech. Another approach is to use Ng et al. [22] uses an electromagnetic radar and microphone to detect the user's muscle movement and airflow to detect the user's silent speech. Jorgensen et al. [16] uses electromyogram (EMG) signals to recognize the word from a six-word vocabulary. Similarly, AlterEgo [17] used a similar method to achieve 92% accuracy on about 50-word vocabulary. Even acoustic sensing methods [39], using ultrasonic transmitters for mouth movement detection, faced similar vocabulary and hardware limitations. These systems require unconventional hardware and wearable placement, increasing the barrier to adoption. WhisperBuds uses commercial-off-the-shelf hardware and a common wearable location to ease the adoption of silent speech interfaces.

Computer vision techniques, notably lip-reading, represent another prevalent approach in silent speech interface development. One of the earliest approaches is LipNet [2], which uses a convolutional neural network (CNN) to classify lip image sequences into words. LipType [2] further improved the accuracy of LipNet by using a repair model. However, capturing these lip images requires a camera in front of the user's mouth, which is not socially appropriate in many situations. One solution is to use a camera located on a necklace [38] to capture the user's neck and chin movement. However, lip reading has intrinsic ambiguity since the number of distinguishable speech units from lips (visemes) is much less than from sound (phenomes). LipLearner [33] tries to address this by allowing the user to create one-shot customization for specific, frequently used actions. In general, image-based silent speech interfaces still have limitations to being a generic silent speech interface.

2.3 Earbuds Interfaces

Like WhisperBuds, many projects have used earbuds as a user interaction interface for purposes other than speech input.

One common direction is health sensing. Poh et al. [24] have proposed using earphones to monitor the user's heart rate using PPG. Earbit [3] have used a proximity sensor and accelerometer while Morshed et al. [20] used IMU to track the user's eating moments. Chan et al. [4] used ultrasound transducers and microphones to detect hearing problems.

EarHealth [14] modded an earphone to equip a microphone and a speaker inside the ear canal to help diagnose ear diseases.

Another direction is user identification. Earmonitor [34] used a setup similar to EarHealth for user identification and heart rate monitoring. EarPPG [6] achieved user identification using PPG instead of a microphone and speaker pair.

These projects explore different capabilities of the earbud form factors other than speech recognition, which is what WhisperBuds's focus is on.

3 SYSTEM DESIGN

To provide a socially appropriate and accurate way of whisper speech input, we want WhisperBuds to achieve the following design goals:

- **D1** Users can wear WhisperBuds at a location similar to common wearable devices, so wearing the device in daily life is not awkward.
- **D2** Users can whisper at a low volume, similar to common office tools, so it's not disturbing for people around the primary user.
- **D3** Users can use their natural whisper voice for an easy learning curve and fast input speed while achieving high accuracy.

3.1 Socially Appropriate Location

For **D1**, we must find a socially appropriate location for WhisperBuds. Many other whisper speech recognition projects use either a near mouth microphone like SilentWhisper [13] or SilentMask [12]. Although these devices effectively capture whispered speech, the form factors of the wearable are not commonly worn in daily life and may raise social concerns. WhisperBuds tries to solve this problem by exploring standard wearable devices and identifying a socially acceptable location capable of capturing whisper speech. From Zeagler [37], we know the most common wearable device locations closest to the mouth are the ears (like earbuds), the neck (like necklaces), and the upper arm (like smartwatches). We chose the ear location because it is closer to the mouth, and its distance is more stable than the other two locations. This closeness and stability are essential for receiving high-quality whisper speech signals and enhancing the signal quality further.

3.2 Work with Low Volume Signals

Even with the ear location, the volume of whisper speech is still too low for the microphone to capture (**D2**). According to Engineering Toolbox [35], a quiet whisper is around 30 dB at 1 meter away from the speaker, which is around 50 dB when measured from the ear's location (assuming the distance between the mouth and the ear is 0.1 meters). Compared to another person speaking at a normal volume, around 60 dB at 1 meter from the speaker, the user's whisper speech is a thousand times quieter. This low volume means a low signal-to-noise ratio (SNR). With poor signal quality, a transcription model will have difficulty transcribing only from the whispered speech.

This low SNR issue can't be solved simply by applying better speech-enhancement models such as Chatterjee et al. [5]. The model/algorithm considers the user's whispered speech and the environment's speech-like noise as speech signals. A good speech recognition model will accurately recognize both without distinguishing between them. The inability to distinguish signals of different sources but the same type is intrinsic to single microphone data collection, which is insufficient to distinguish the sound sources.

Luckily, common in-ear earbuds have multiple microphones to cancel out background noise for a better listening experience. A pair of typical earbuds have three microphones:

- **Talk microphone** is closest to the mouth and captures the user's voice.
- **Feedforward microphone** (i.e., reference microphone) is the microphone closest to the outside world and captures the outside world's sound for noise cancellation.
- **Feedback microphone** (i.e., error microphone) is inside the earbuds's sound channel. It forms a sealed space with the user's ear canal and captures the sound reflected for noise cancellation.

In regular noise-cancelling earbuds, feedforward and feedback microphones collectively provide noise cancellation for less background noise in the hearing experience. The feedforward microphone collects external noise and passes it through a filter to produce an "anti-noise" signal. This inversely proportional signal effectively neutralizes the external noise when played. Due to the earcup itself already isolating the noise from the user's ear, this "anit-noise" signal usually needs to be transformed before playing back in the earbuds' speaker. However, this transformation is hard to be exact in practice. The fit of the headphones, the type of noise source, or the position of the sound source may change the ideal transform. Therefore, many earbuds have another feedback microphone that can be used to observe the sound in the sealed ear canal space to approximate what the user's ear is hearing. With this, the earbuds can dynamically adjust the transform function to reach the lowest noise picked up from the feedback mic to reduce noise at the user's ear. This hybrid approach [32] ensures high-level noise cancellation and a clean, unimpeded listening experience in high-noise environments.

For WhisperBuds, we use these microphones differently to perform a "noise-canceling" for a cleaner signal of whispered speech. We found that the feedback microphone inside the ear canal contains the best quality whisper speech signal. However, even that signal is severely affected by the environmental noise. To solve this problem, WhisperBuds employs a *noise-canceling* neural network to enhance the signal quality by leveraging the multiple microphones on the earbuds. The noise-canceling neural network takes in the three microphone signals and outputs a single signal that resembles the feedback microphone's signal in a quiet environment. Conceptually, the neural network identifies the noise signal by looking at stronger signals in the feedforward and talk mic (which contains more noise and less whisper) compared to the feedback mic (which contains more whisper and less noise) and removes these noise signals from the feedback mic. As a result, we get a clean whispered speech signal that is much easier to transcribe.

We trained the noise-canceling neural network to reconstruct a clean feedback microphone signal from the noisy signal captured by all three microphone channels. We collected a dataset of 65 people's whispered speech and 10 hours of background noise, both recorded with the same earbuds. We then used dataset augmentation techniques to randomly overlay the background noise on top of the whispered speech to simulate a noisy environment. The ground truth for each item is the feedback microphone signal before mixing with background noise.

3.3 Adapt to the Sound Profile of Whispered Speech

Regular speech recognition models wouldn't work with the sound captured by WhisperBuds because the sound profile differs significantly from those in a typical speech dataset (**D3**). Two reasons cause the difference:

- Different sound source: Whispered speech intrinsic has a different sound profile than normal speech due to people's vocal cords not vibrating during whispered speech.

- Different transmission route: WhisperBuds records sound transmitted through the user's body tissue in their head and captured in an enclosed ear canal cavity compared with a typical speech dataset where the sound is transmitted in open air.

We compared the performance of the state-of-the-art speech recognition model, OpenAI Whisper, on the captured whispered speech, and the word error rate (WER) is around 60%. This is completely unusable for a speech recognition system. Therefore, we must produce a new model *general-purpose whisper recognition model* that can recognize whisper speech in general.

A typical approach [12] is to train a new model for the new sound profile from the ground up. However, this requires substantial additional training data to achieve a versatile model capable of handling various accent styles and texts. For reference, OpenAI Whisper is trained on 680k data hours, which is infeasible in a research setting.

Instead, we use a transfer-learning approach to adapt the model to the new sound profile. The OpenAI Whisper model consists of a transformer-based encoder-decoder model. After being transformed into a Mel spectrogram representation, the audio signal is first encoded as a vector of hidden states and then decoded into a sequence of words. To ensure the model adapts to the new sound profile without losing the ability to generate different kinds of text, we only fine-tune the first few encoder blocks of the model. Conceptually, the first few layers/encoder blocks are responsible for extracting the sound profile information from the audio signal, and the last few layers/decoder blocks are for understanding semantics and generating text.

4 SYSTEM IMPLEMENTATION

4.1 EarBuds

The crucial pieces of the WhisperBuds system are the earbuds. Because the whispered speech magnitude is very low and may fade into the background noise, the typical compression codec used in Bluetooth earbuds can easily remove the whispered speech signal altogether. Also, we need to access the signal from all three microphones to perform the noise-canceling. As of Bluetooth 5.3, there is neither any way to carry a three-channel audio input signal nor any lossless way to transmit a microphone signal.

WhisperBuds uses an open-hardware open-source earbuds design, the PineBuds Pro[2], as the base hardware. We customized the firmware to access the raw audio from the three microphones at 16kHz sampling frequency and 16-bit depth. As for transmission, three channels of raw 16-bit PCM audio at 16kHz sampling frequency is 768kbps in data. This is too high for typical Bluetooth Low Energy (BLE) transmission with a maximum data rate of 1Mbps. We tried Bluetooth 5.3, which has a maximum data rate of 2Mbps, but the transmission is unstable. Therefore, we choose to use Bluetooth Classic + Extended Data Rate (EDR), which has a maximum data rate of 3Mbps. We used the Serial Port Profile (SPP) protocol, usually used for text communication, to transmit the audio signal.

4.2 Dataset Collection

We need to collect data from diverse people for a large dataset to train the noise-canceling and whisper-recognition models. Luckily, because WhisperBuds uses a commercially available earbuds design, and it's wireless, we can easily collect data from many people. We build a data collector device based on Raspberry Pi 4 and a custom-designed app that can record the three-channel audio from the earbuds and a separate near-mouth microphone as a backup to debug the data collection process.

[2] https://wiki.pine64.org/wiki/PineBuds_Pro

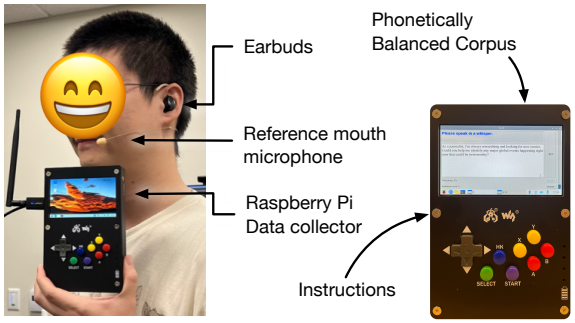

Fig. 2. We collected 21.2 hours of recordings from 65 individuals using a custom-built device to collect speech and ambient noise data. Left: A participant preparing to collect data with the handheld device, a Raspberry Pi 4 housed in a game console-like casing, with a Bluetooth dongle attached for enhanced Bluetooth connectivity stability. The user is equipped with headphones and a front-mouth microphone.
Right: The interface of the data collection device, features an app that provides instructions and text content for the user to read. The buttons below facilitate interaction with the data collector.

We collected two datasets:

- **Background noise dataset**: Two of our paper authors collected 12.4 hours of background noise in different environments, including office, meeting, driving, street, coffee shop, cafeteria with the earbuds.
- **Whisper speech dataset**: We recruited 65 people and recorded 21.2 hours of whisper speech in a typical office room (~35dBA) with the earbuds.

We build empirical algorithms for the background noise dataset to detect when the participant is speaking or if the earbuds are being adjusted and remove those segments from the dataset.

For the whisper speech dataset, we asked participants to read a set of 150 sentences each. These sentences were randomly drawn with a 50% probability from phonetically-balanced datasets (TIMIT [11] and Arctic Speech [19]) and a 50% probability from a regular speech dataset (Common Voice [1]). We then cut the audio by sentences and removed the sentence where the user marked that they made a mistake.

The 65 participants are aged from 18 to 65 (μ = 27.3, σ = 11.0). We asked for 29 of the participants their gender and we got 13 females, 16 males. Every participant for the whisper speech dataset did a data collection session for 30 minutes and was compensated with a $15 gift card. This is approved by our institutional review board.

4.3 Model Training

We trained the model in two stages, first the noise-canceling model and then an end-to-end whisper recognition model.

4.3.1 Noise-Canceling Model. The noise-canceling model uses the three-channel (feedback, feedforward, and talk mic) audio signal with noise and outputs a single-channel (like feedback) audio signal without noise.

To train the model, we first segmented the background noise and whisper speech dataset into 1.28-second segments. We then randomly overlayed the background noise on top of the whisper speech dataset to simulate a noisy environment. Then, we can use the feedback microphone signal in the whisper speech dataset as the ground truth for the noise-canceling model. We used 80% of the data for training, 10% for validation, and 10% for testing.

The model's architecture is constructed as a five-layer U-Net, whose input and output are both 80x128 Mel spectrograms. The input has three channels corresponding to three microphones, and the output is a single channel (See

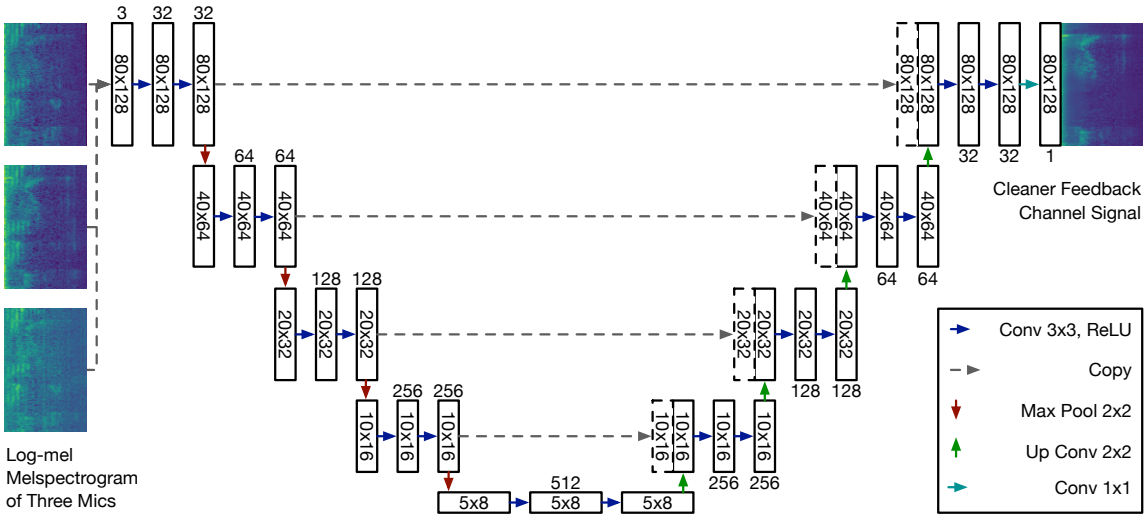

Fig. 3. WhisperBuds' noise canceling model can remove the background noise and preserve users' whispered speech only. This five-layer U-Net model takes in 80x128 Mel spectrograms from three microphones (three channels) during training and outputs a single-channel Mel spectrogram. For U-Net's encoder, The number of features in each layer is twice that of the previous layer, while the width and height of the data are reduced by half. Between each layer is the repeated application of two 3x3 convolutions, a rectified linear unit (ReLU), and a 2x2 max pooling operation with stride 2 for downsampling. Every step in the decoder consists of an upsampling by a 2x2 up-convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the encoder layer, and two 3x3 convolutions, each followed by a ReLU.

Figure 3). The multi-channel design makes it easy to expand to more complex microphone array scenarios. Thanks to the structure of UNet, the input length of the model during inference is dynamic. Any audio segment that is an integer multiple of 0.16s in length can be denoised. It is worth mentioning that the model is relatively small, with only 7.8 million parameters and occupying 30 MB of space.

The model is trained with a batch size of 128 for 20000 epochs with an initial learning rate of 0.001 and a ReduceOnPlateau scheduler.

4.3.2 Whisper Recognition Model. With a clean whisper speech signal, we can finally output a transcript with a speech recognition model. We used the OpenAI Whisper model as the base model. We trained the model end-to-end with three channels of audio input and output of text. During training, we piped the output of the noise-canceling model into the OpenAI Whisper model.

We fine-tuned the noise-canceling model and the first 12 encoder blocks of OpenAI's Whisper-medium model (24 encoder blocks total). We only touch the first half of the encoder blocks in order to make the base OpenAI Whisper model adapt to WhisperBuds' sound profile while keeping its capability to generate diverse human languages. The initial layers/encoder blocks extract sound profile information from the audio signal, while the final layers/decoder blocks comprehend semantics and generate text.

For fine-tuning the model, we used a batch size of 32 for 4000 epochs with a learning rate of 1e-4, larger than the typical fine-tune learning rate of 1e-5, to nudge the model to learn the new sound profile. We used a warm-up of 500 steps and linear decay for the learning rate. To make the convergence faster and more stable, the parameters of the noise canceling model are initialized from the trained noise canceling model stated above. However, we initialized the

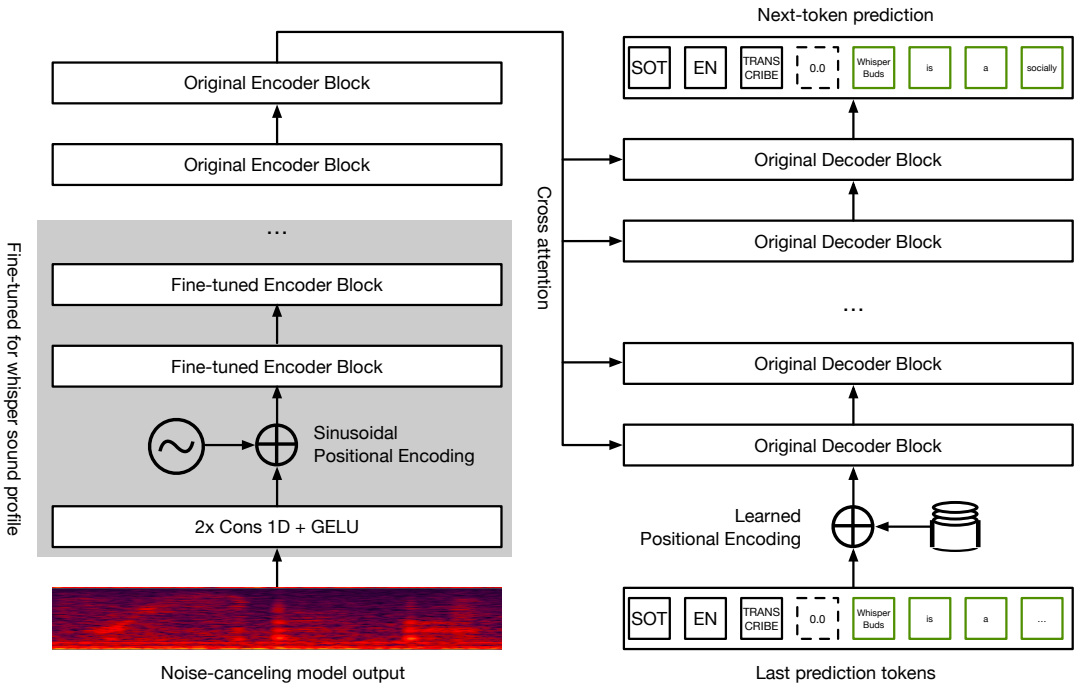

Fig. 4. The structure of the Whisper speech recognition model is based on an encoder-decoder transformer. The Whisper model's input is a Mel-frequency cepstrum, which is of the same format as our noise-canceling model's output. The cepstrum is passed to an encoder. A decoder is trained to predict later text captions. Special tokens are used to perform several tasks, such as specifying some language. We only fine-tuned the convolutional layers and the first half (12/24) of the encoder blocks to keep its understanding of lingual knowledge.

Whisper model from OpenAI's checkpoint instead of the one fine-tuned on our dataset without the noise-canceling model, because we wanted to avoid overfitting to our limited text corpus.

4.3.3 *Dataset.*

Data Augmentation. To prepare the dataset, we cut the noise data into segments of 30 seconds. We then used the whisper speech dataset, where the user sentences are always shorter than 30 seconds, and randomly overlayed a random part of a random segment of background noise on top of the whisper speech dataset to simulate a noisy environment. We match one clean speech with five random noise segments, resulting in a five times larger dataset. According to our observation, the data augmentation greatly helps with the generalizability of the speech-to-text model.

Out-of-distribution Removal. An additional insight that can be drawn from this study is that despite the large size of the dataset, a small amount of bad data can have a significant negative impact. By eliminating 3% of the outlier data in the training set, we have reduced the error rate by 2-3%. There is an overall improvement in the accuracy of each user. We identified these outlier data testing inferencing accuracy on a previously trained model.

4.4 Live Inferencing System

WhisperBuds support live inferencing with the entire reference noise-canceling, whisper recognition, and speech self-correction pipeline. To achieve this, the system is divided into three parts:

(1) A *user interface (UI) Library* built in React and Typescript that allows application developers to integrate WhisperBuds input into their applications easily.
(2) A *driver* built with Python that communicates with the earbuds, library, and server to transfer information and commands.
(3) A *server* built with Python that runs the noise-canceling and whisper recognition model.

4.4.1 UI Library. The UI library renders a text box and a button to start the speech recognition process. When the user clicks the button, the UI library will send a command to the driver to start the speech recognition process.

4.4.2 Driver. The driver is responsible for communicating with the earbuds, UI library, and server. It uses the PyBluez library to communicate with the earbuds via SPP and the `websockets` library to communicate with the UI library and server. The driver maintains an ongoing background connection with the earbuds. Upon receiving a request from the library to begin speech recognition, it instructs the earbuds to start recording and forwards the audio signal to the server. The driver then relays the server-generated transcript back to the UI library.

4.4.3 Server. The server in WhisperBuds system, equipped with an Nvidia A100 GPU, is responsible for executing the noise-canceling and whisper-recognition models. It first employs a Voice Activity Detection (VAD) algorithm to detect user speech, then uses the noise-canceling model to enhance signal quality, and finally, the whisper recognition model generates the transcript.

To better support streaming inference, we want to have a low latency and maintain high accuracy. The OpenAI Whisper decoder may look at previous and future tokens to determine the best current. So, maintaining a data window before and after the current timestamp is important for the model to generate the best transcript.

Inspired by fast-whipser[3], we maintain a trailing time window as the model's input, and then move the time window forward based on the model's output. Specifically, if the results of two consecutive inferences start with the same set of words, these words are "committed" and will not change thereafter. In another scenario, if the time window exceeds 30 seconds, words are committed until the window is less than 30 seconds. Whenever the committed words form a complete sentence, the time window is moved to the end of the sentence. The committed sentences serve as prompts for Whisper, ensuring more consistent generation in subsequent outputs.

We further adjusted the repetition penalty (1.1), lowered the max temperature (0.8), and raised the no-speech threshold (0.55) to improve the accuracy of the model for live inference. It turned out that improving accuracy also helped to decrease the inference time, as the model is more assured of giving no-speech prediction, stopping unnecessary decoding earlier.

5 EVALUATION

To evaluate WhisperBuds, we conducted a model performance evaluation to evaluate the model's performance and validate the design choices. We also conducted a user study to evaluate the user experience of WhisperBuds. In the following sections, we will measure word error rate (WER) similarly to the OpenAI Whisper paper [26].

[3] https://github.com/guillaumekln/faster-whisper

Table 1. WhisperBuds outperformed the other two models to a large extent regardless of whether the user's data was included in the training set. The within-session WER is obtained by randomly separating 10% of the dataset for testing, while the cross-user WER is determined by randomly selecting 20% of the users from the dataset as the test set. By comparing the two metrics, we found that WhisperBuds can generalize well to new users.

Model	Train Loss	Test Loss	Within-session WER(%)	Cross-user WER(%)
Original Whisper	N/A	3.751	N/A	60.02
Fine-tuned Whisper	0.563	0.978	21.05	37.97
Full Pipeline	**0.159**	**0.202**	**2.55**	**13.91**

5.1 Model Performance

We first evaluated the performance of the entire noise-canceling and whisper-recognition pipeline. Then, we tried ablation studies to evaluate the effectiveness of the noise-canceling model and the fine-tuning of the OpenAI Whisper model. Finally, we evaluated the performance of the noise-canceling model alone.

5.1.1 Performance comparison with different models. We trained the model as described in Section 4.3. In order to evaluate the performance of our model, we have designed two methods. The first method, producing within-session WER, involves randomly dividing the entire dataset into 80% for training, 10% for validation, and 10% for testing, and reporting the performance of the model that performs best on the validation set on the test set. The second method involves randomly selecting 20% of the users from the dataset as the test set, referred to as cross-user accuracy. This allows us to assess the model's ability to generalize. For the second method, we repeat it seven times and take the average. Results are shown in Table 1.

The original model's error rate is unacceptably high. After fine-tuning with the augmented dataset, the Whisper model learned to adapt to the new sound profile of the recorded whispered speech. We have seen a significant error rate drop from 60% to 21%, but it is far from perfect. In terms of zero-shot cross-user WER, basic finetuning resulted in a 37.97% WER, rendering it ineffective.

However, by incorporating a small noise-canceling model before the Whisper's input, we observed a substantial increase in both accuracy and robustness. It achieved 2.55% of within-session WER and 13.91% of cross-user WER. We can infer from the narrowed gap that the noise-canceling model improves the SNR of the audio signal, thus guiding the model to adapt to different noises and user profiles.

It's worth noting that even though there has been extensive research on speech recognition models, their robustness in practical applications, especially in tests of lower quality, remains a problem. Even the state-of-the-art normal speech recognition model, OpenAI's Whisper, reports an average WER of 12.8% across various datasets in its paper [26].

5.1.2 Full pipeline performance with different amplitude of noise. As one might expect, typical office noise encompasses a variety of sounds, such as conversations, footsteps, ringing phones, and spinning fans. Among these, the sound of others speaking in the background often confuses traditional speech-to-text methods, as they struggle to distinguish between the user's voice and those of others. We utilize this evaluation to showcase the noise-canceling feature of WhisperBuds.

We collected an evaluation set with a male and a female speaker, each reading the same randomly selected 150 sentences from the Common Voice dataset's test set under four different noise levels (35dBA, 45dBA, 55dBA) of office

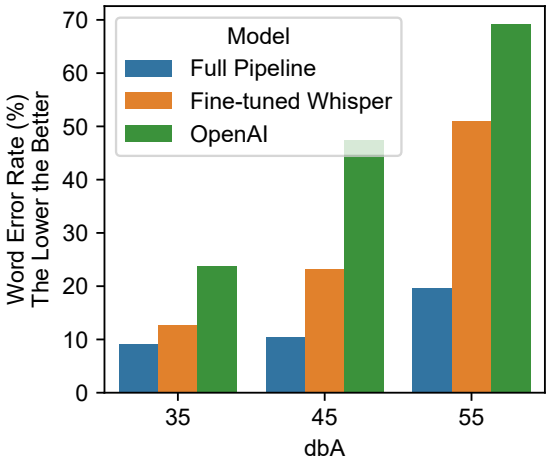

Fig. 5. The WER of models under different amplitude of noise. WhisperBuds managed to keep almost the same performance when background noise rose to 45 dbA, while other methods doubled their error rate and became unusable at 45 dbA. The results demonstrated that background chatter significantly impairs the effectiveness of contemporary speech-to-text models. However, WhisperBuds can isolate the user's voice from surrounding noise, maintaining optimal performance.

noise[4]. It is worth noting that our model never meets the speakers' recordings nor the office noise we used during training. The evaluation results are shown in Figure 5.

Our full-pipeline model can maintain high accuracy with background noise up to 55 dbA and stay at a usable accuracy. The average WER of original Whisper, fine-tuned Whisper, and full pipeline model are 10.40%, 23.10%, and 47.33%, respectively, at 45 dbA, while those at 55 dbA are 19.68%, 50.93%, and 69.13%. The performance of our system at 35 dbA and 45 dbA are roughly the same, both substantially better than the other two approaches. Although 45 dbA or 55 dbA may not seem excessively loud, these noise levels can already compromise the performance of existing state-of-the-art models. That means our noise-canceling model can extract useful information from considerably noisy environments.

5.1.3 Performance of the noise-canceling model. We independently tested the performance of the noise-canceling model. Our training method is described in Section 4.3.1. The results were very encouraging; we found that the output of the noise reduction model not only removes background noise (such as the sound of people talking) from the input, but in many cases, the overall quality of the output even surpasses our labels. Although our training labels did not include overlaid noise, they still contained noise due to inherent room noise and poor microphone quality. Figure 6 demonstrates the performance of WhisperBuds' noise-canceling model on the test set. The output of the noise-canceling model after end-to-end training is also shown in the figure, which seems to bring even more details of the whisper signal.

5.2 Text Input Comparisions

To verify the feasibility of WhisperBuds as a text input method, we conducted a user study comparing WhisperBuds with normal speech and mobile keyboard input.

[4]https://www.youtube.com/watch?v=D7ZZp8XuUTE

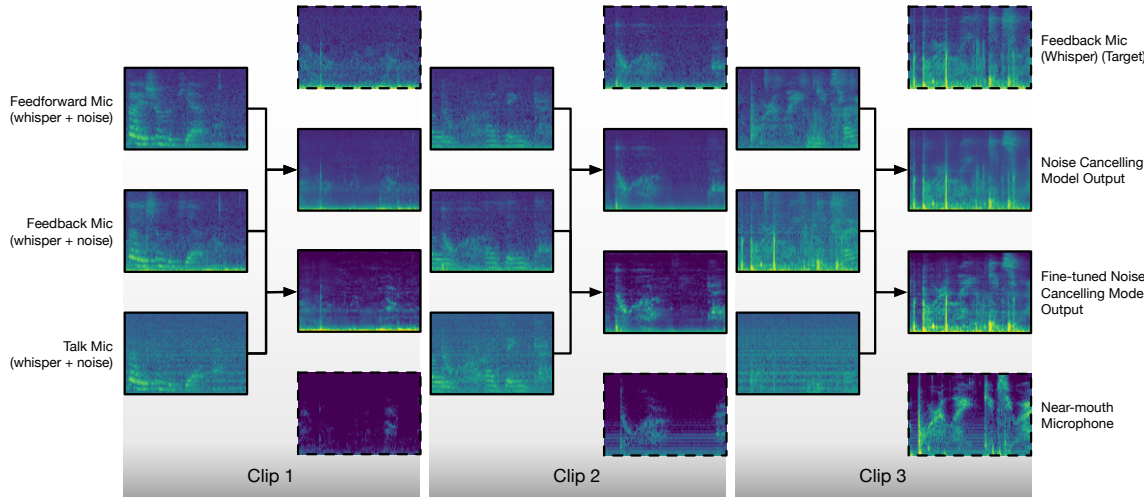

Fig. 6. The end-to-end trained noise-canceling model can recover the most signals from noisy inputs. The figure shows three recorded audio clips from the test set and the noise-canceling model's inference results. For each clip, the left side shows the input from three channels, while the right side, from top to bottom, displays the reference output, the output from the model trained on the noise-canceling task only, the output from the noise-canceling model after end-to-end training, and the signal from the front mouth microphone.
We are interested in the subtle speech features in the spectrogram, which the front-mouth microphone can capture. Therefore, the closer the image is to the front-mouth microphone, the more information it contains. Comparing the two models, we found that the model without end-to-end training removes background noise but also weakens useful information. In contrast, the model with end-to-end training has lower background noise and further enhances speech signals.

5.2.1 Procedure. We selected 12 paragraphs of text from https://www.reddit.com/, each trimmed to around 120 words. All the posts are created after June 2023 to ensure the test set is not in the base OpenAI Whisper's training set. We asked participants to try three different input methods (WhisperBuds, normal speech, and mobile keyboard) in a counter-balanced order to input seven paragraphs each. The first paragraph of each input method is a practice paragraph to familiarize the participants with the input method and the interface. We built an interface allowing participants to see the target text while inputting it with the designated technique. We asked the participants to submit the text once they thought the input text conveyed the same meaning as the target text. We measured the text input time and the final text's word error rate (WER) for each input method. After each condition, we asked the participant to fill out a NASA-TLX questionnaire for perceived workload and an SUS questionnaire for usability. At the end of the study, we asked the participant to fill out a questionnaire about their experience with different conditions and which input method they would prefer at different locations.

5.2.2 Participants. For this study, we recruited 12 proficient English-speaking participants (six females, six males) aged 20-30 years ($\mu = 24.6, \sigma = 3.6$). All participants used the mobile keyboard a few times a day. Eight participants used normal voice input a few times a day, and the other four used voice input less than once a week (the lowest possible option). One of our participants tried whisper speech input[5] before, two of our participants used it once per week, and the other nine used it less than once per week (the lowest possible option). The study takes around 30 minutes, and participants are compensated with a $15 gift card. Our institutional review board approved the study.

[5]Smartphone's voice input can recognize whispered speech when the user's mouth is close to the microphone.

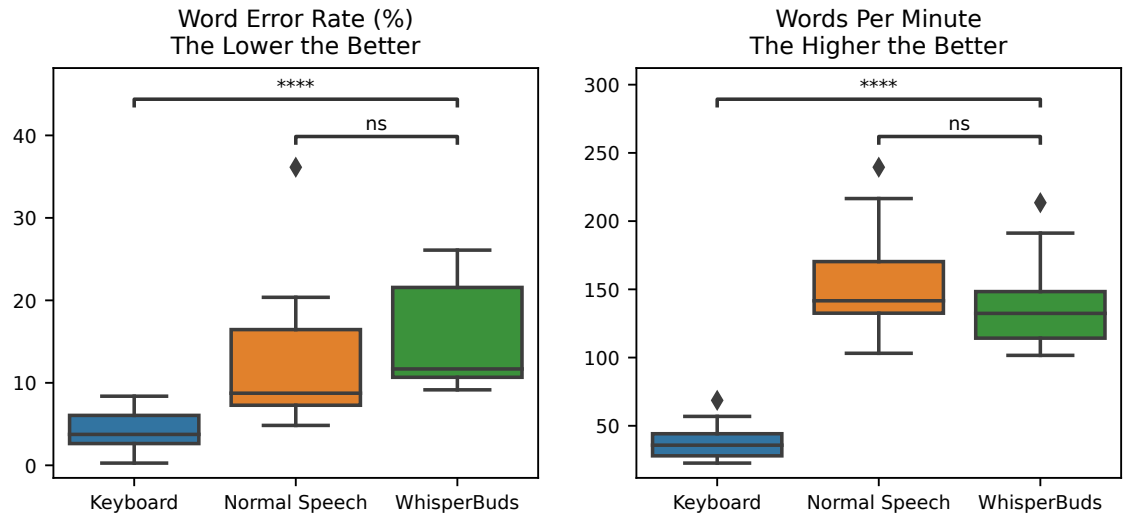

Fig. 7. Type speed and accuracy of WhisperBuds, normal speech, and mobile keyboard. WhisperBuds achieved comparable accuracy to normal speech recognition, and both are significantly faster than typing on a keyboard.

5.2.3 *Results.*

Type speed and accuracy. We measured the type speed and accuracy of WhisperBuds, normal speech, and mobile keyboard. The result is shown in Figure 7. The average WPM of keyboard input is 38.5 WPM (σ = 2.5), similar to the 36.2 WPM reported in Palin et al. [23]. The average WPM of normal speech input is 156.4 WPM (σ = 38.8), similar to the 153 WPM reported in Ruan et al. [30]. The average WPM of WhisperBuds input is 138.4 WPM (σ = 33.2), which is 88.5% of the normal speech input speed.

The average WER of keyboard input, normal speech input, and WhisperBuds input are 4.2% (σ=2.5%), 12.7% (σ=8.5%), and 15.5% (σ=6.2%), respectively. That means WhisperBuds can reach 97%(= $(100 - 15.5)/(100 - 12.7)$) of the accuracy of normal speech input.

This showed that WhisperBuds can achieve a similar type of speed and accuracy as normal speech input. It's much faster than mobile keyboard input but has a slightly lower accuracy.

Perceived workload and usability. We measured the perceived workload and usability of WhisperBuds, normal speech, and mobile keyboard. The result is shown in Figure 8. The median (per-user average) NASA-TLX score of keyboard input, normal speech input, and WhisperBuds input is 4.2 (σ=1.3), 2.1 (σ=1.3), and 2.6 (σ=1.2), respectively. We conducted a Wilcoxon signed-rank test and found that WhisperBuds has a significantly lower perceived workload than keyboard input (p=0.03), while there is no significant difference between WhisperBuds and normal speech input (p=0.23). The average SUS score of keyboard input, normal speech input, and WhisperBuds input is 70.6 (σ=12.2), 72.1 (σ=10.8), and 71.7 (σ=11.7), respectively. We conducted a paired t-test and found no significant difference between WhisperBuds and the other two input methods.

This showed that WhisperBuds has a similar perceived workload and usability as normal speech input and a lower perceived workload than mobile keyboard input.

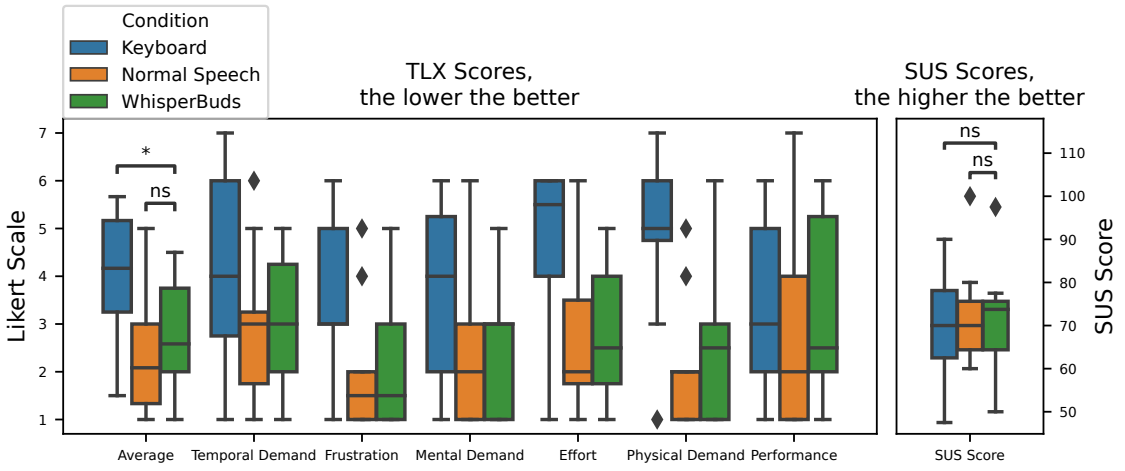

Fig. 8. Perceived workload and usability of WhisperBuds, normal speech, and mobile keyboard. WhisperBuds has a significantly lower perceived workload than keyboard input (p=0.03), while there is no significant difference between WhisperBuds and normal speech input.

User preference. In our study, we explored participant preferences for different input methods across common locations in daily American life [18]. The results indicated a clear preference for mobile keyboards in offices, bars, other indoors, and overall, favored for the unobtrusiveness in public spaces where loud speech might be intrusive. Participants appreciated the keyboard for its accuracy (P1, P12) and privacy (P5, P8), though some (P1, P2, P10) noted its limited speed and expressed a desire for improved prediction and auto-correction.

In contrast, loud speech was the preferred method in private settings such as residences, vehicles, and outdoors, where speaking freely is more socially acceptable. Participants appreciated loud speech for its speed (P10, P11) and naturalness (P3, P6). However, opinions were divided regarding the effort involved in loud versus whispered speech. While P3 found loud speech more effortless, P5, P7, P9, and P10 leaned towards whispered speech.

Although WhisperBuds did not emerge as the top choice in any location, they were the second most preferred method in offices, bars, other indoors, outdoors, and overall where the mobile keyboard was predominant. This underscores WhisperBuds' potential in contexts where loud speech is inappropriate.

Whisper speech was highlighted for its discreet nature (P1, P3, P4, P7, P9, P12). P1 pointed out that while reading long texts like the 120-word Reddit paragraphs was unusual, WhisperBuds would be highly beneficial for shorter messages. A common suggestion for both normal speech and WhisperBuds, particularly from P4, P7, and P11, was the addition of a live error correction system for on-the-fly adjustments.

Overall, the study shows WhisperBuds as a strong complementary option to traditional input methods, particularly valued for its naturalness and suitability in various social environments.

Noise level. According to our measurements, the background noise in the test site was 35.08 dB. We adopted a similar testing method as how rting.com measures keyboard noise[6]. If a user uses normal voice input, the sound measured from 40cm away is 44.55 dB, while a whisper is only 37.42 dB. In fact, WhisperBuds is quieter than most (111 out of 120)

[6]https://www.rtings.com/keyboard/tests/typing-noise

keyboards measured by rting.com. This makes WhisperBuds as quiet as a keyboard, expanding the use cases for voice input.

Inference Latency. Inference latency is crucial for a real-time speech recognition system. To this end, we measured the average time for a single inference on a live inference system with Voice Activity Detection (VAD) enabled for two models. The system used one Nvidia A100 GPU to accelerate the inference. The original Whisper model had a latency of 0.418 ± 0.298s, while our full pipeline had a latency of 1.056 ± 0.602s. We believe the increased latency in the fine-tuned model is due to the higher uncertainty associated with whispered speech, which results in a more time-consuming search process when decoding.

Additionally, the noise-canceling model took 0.02s to process a 0.64s long audio clip, indicating that it is not a notable latency overhead during live inference.

6 DISCUSSION

6.1 Applications

As a general-purpose, socially appropriate speech interface, WhisperBuds can be used in many applications.

One obvious application is to replace the keyboard on mobile devices. Emerging devices such as smartwatches, smart TVs, and smart glasses typically have much slower input speeds than mobile phones. For example, a typical VR input speed is around 5.9 WPM [36], while a typical mobile input speed is around 36.2 WPM [23]. This significantly limits the productivity of these devices. With the help of WhisperBuds, these devices can achieve much higher input speed while maintaining a socially appropriate form factor.

Another application is to be used to communicate with an always-available AI agent that can understand your context and help you with your daily life. WhisperBuds can listen to the user's daily conversations and get familiar with the user's life and preferences. When the user needs help, the user can converse with the AI agent with whispered speech. The user's whispered speech is very low volume and the agent's response can be played back with the earbuds, so the conversation is completely private.

6.2 Multilingual Generalizablitiy

One exciting possibility we discovered while building WhipserBuds is its generalizability across languages. We tried to fine-tune a multilingual Whisper-large model with the same English-only dataset and observed strong performance across languages.

We have created a Chinese whisper dataset, the text corpus of which is excerpted from Zhihu (similar to Quora, but in Chinese) after 2021. Each entry is about 70 characters long, read by two people, with a total duration of approximately half an hour.

The evaluated character error rate (CER) of our model is 11.49%. For comparison, OpenAI reports the original model's performance on Mandarin Chinese to be 7.7% (on FLEURS [7]) and 8.2% (on Common Voice [1]). We also tested the stock model's performance on our Chinese whisper dataset, yielding a result of 25.45%. We observed that the model can recognize Spanish and Portuguese with a very usable accuracy, although we haven't formally evaluated it. This is a pleasant surprise even to us and this highlights that our fine-tuning approach is very effective at adopting sound profiles and yet making sure the model does not lose the ability to generate diverse text across languages.

6.3 Limitations and Future Work

Although the capabilities of WhisperBuds are promising, there are still limitations and future work to be done. One major limitation is the resource requirement of the system. While the current implementation of our model relies on independent graphics cards for processing, it is important to note that this is still a viable approach. The underlying Whisper model, as utilized by OpenAI in commercial applications like ChatGPT. This demonstrates that while WhisperBuds at its current stage is relatively resource-intensive, there is still a wide range of applications that can leverage an online API architecture to serve the users.

Looking towards the future, one promising direction is to explore models like the Fast Conformer-Transducer[40], which are known for their streaming capabilities. This could reduce the resource requirements and even inference on edge devices, making WhisperBuds more accessible and practical for a wider range of use.

Another limitation is the inability to work with earbuds audio playback while performing whispered speech recognition. Our noise-canceling model currently doesn't accommodate the impact of a speaker in the user's ear canal. Thanks to the multi-channel input capability of convolutional neural networks, future efforts could enhance the model to function effectively with an ear canal speaker.

7 CONCLUSION

In conclusion, WhisperBuds opens new possibilities for how we interact with our devices, offering a more natural, private, and non-intrusive mode of communication. We are optimistic about the future applications and development of this technology, hoping it will pave the way for more user-friendly silent speech interfaces.

REFERENCES

[1] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2019. Common Voice: A Massively-Multilingual Speech Corpus. https://doi.org/10.48550/ARXIV.1912.06670

[2] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. 2016. LipNet: End-to-End Sentence-level Lipreading. https://doi.org/10.48550/ARXIV.1611.01599

[3] Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. 2017. EarBit. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (sep 2017), 1–20. https://doi.org/10.1145/3130902

[4] Justin Chan, Antonio Glenn, Malek Itani, Lisa R. Mancl, Emily Gallagher, Randall Bly, Shwetak Patel, and Shyamnath Gollakota. 2023. Wireless earbuds for low-cost hearing screening. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. ACM. https://doi.org/10.1145/3581791.3596856

[5] Ishan Chatterjee, Maruchi Kim, Vivek Jayaram, Shyamnath Gollakota, Ira Kemelmacher, Shwetak Patel, and Steven M. Seitz. 2022. ClearBuds. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. ACM. https://doi.org/10.1145/3498361.3538933

[6] Seokmin Choi, Junghwan Yim, Yincheng Jin, Yang Gao, Jiyang Li, and Zhanpeng Jin. 2023. EarPPG: Securing Your Identity with Your Ears. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. ACM. https://doi.org/10.1145/3581641.3584070

[7] Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech. arXiv:2205.12446 [cs.CL]

[8] B. Denby, T. Schultz, K. Honda, T. Hueber, J.M. Gilbert, and J.S. Brumberg. 2010. Silent speech interfaces. *Speech Communication* 52, 4 (apr 2010), 270–287. https://doi.org/10.1016/j.specom.2009.08.002

[9] M.J. Fagan, S.R. Ell, J.M. Gilbert, E. Sarrazin, and P.M. Chapman. 2008. Development of a (silent) speech recognition system for patients following laryngectomy. *Medical Engineering & Physics* 30, 4 (may 2008), 419–425. https://doi.org/10.1016/j.medengphy.2007.05.003

[10] Masaaki Fukumoto. 2018. SilentVoice. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. ACM. https://doi.org/10.1145/3242587.3242603

[11] Garofolo, John S., Lamel, Lori F., Fisher, William M., Pallett, David S., Dahlgren, Nancy L., Zue, Victor, and Fiscus, Jonathan G. 1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus. https://doi.org/10.35111/17GK-BN40

[12] Hirotaka Hiraki and Jun Rekimoto. 2021. SilentMask: Mask-type Silent Speech Interface with Measurement of Mouth Movement. In *Augmented Humans Conference 2021*. ACM. https://doi.org/10.1145/3458709.3458985

[13] Hirotaka Hiraki and Jun Rekimoto. 2022. SilentWhisper: faint whisper speech using wearable microphone. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM. https://doi.org/10.1145/3526114.3558715

[14] Yincheng Jin, Yang Gao, Xiaotao Guo, Jun Wen, Zhengxiong Li, and Zhanpeng Jin. 2022. EarHealth. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. ACM. https://doi.org/10.1145/3498361.3538935

[15] Yincheng Jin, Yang Gao, Xuhai Xu, Seokmin Choi, Jiyang Li, Feng Liu, Zhengxiong Li, and Zhanpeng Jin. 2022. EarCommand. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (jul 2022), 1–28. https://doi.org/10.1145/3534613

[16] C. Jorgensen, D.D. Lee, and S. Agabon. [n. d.]. Sub auditory speech recognition based on EMG signals. In *Proceedings of the International Joint Conference on Neural Networks, 2003.* IEEE. https://doi.org/10.1109/ijcnn.2003.1224072

[17] Arnav Kapur, Shreyas Kapur, and Pattie Maes. 2018. AlterEgo. In *23rd International Conference on Intelligent User Interfaces*. ACM. https://doi.org/10.1145/3172944.3172977

[18] NEIL E KLEPEIS, WILLIAM C NELSON, WAYNE R OTT, JOHN P ROBINSON, ANDY M TSANG, PAUL SWITZER, JOSEPH V BEHAR, STEPHEN C HERN, and WILLIAM H ENGELMANN. 2001. The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology* 11, 3 (July 2001), 231–252. https://doi.org/10.1038/sj.jea.7500165

[19] John Kominek and Alan W. Black. 2004. The CMU Arctic speech databases. In *Proc. 5th ISCA Workshop on Speech Synthesis (SSW 5)*. 223–224.

[20] Mehrab Bin Morshed, Harish Kashyap Haresamudram, Dheeraj Bandaru, Gregory D. Abowd, and Thomas Ploetz. 2022. A Personalized Approach for Developing a Snacking Detection System using Earbuds in a Semi-Naturalistic Setting. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. ACM. https://doi.org/10.1145/3544794.3558469

[21] Y. Nakajima, H. Kashioka, K. Shikano, and N. Campbell. [n. d.]. Non-audible murmur recognition input interface using stethoscopic microphone attached to the skin. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).* IEEE. https://doi.org/10.1109/icassp.2003.1200069

[22] L.C. Ng, G.C. Burnett, J.F. Holzrichter, and T.J. Gable. [n. d.]. Denoising of human speech using combined acoustic and EM sensor signal processing. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100).* IEEE. https://doi.org/10.1109/icassp.2000.861925

[23] Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. 2019. How do People Type on Mobile Devices?. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. https://doi.org/10.1145/3338286.3340120

[24] Ming-Zher Poh, Kyunghee Kim, Andrew Goessling, Nicholas Swenson, and Rosalind Picard. 2012. Cardiovascular Monitoring Using Earphones and a Mobile Device. *IEEE Pervasive Computing* 11, 4 (oct 2012), 18–26. https://doi.org/10.1109/mprv.2010.91

[25] 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 *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3411764.3445687

[26] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. https://doi.org/10.48550/ARXIV.2212.04356

[27] Jun Rekimoto. 2022. DualVoice: Speech Interaction that Discriminates between Normal and Whispered Voice Input. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM. https://doi.org/10.1145/3526113.3545685

[28] Jun Rekimoto. 2023. WESPER: Zero-shot and Realtime Whisper to Normal Voice Conversion for Whisper-based Speech Interactions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3544548.3580706

[29] Jun Rekimoto and Yu Nishimura. 2021. Derma: Silent Speech Interaction Using Transcutaneous Motion Sensing. In *Augmented Humans Conference 2021*. ACM. https://doi.org/10.1145/3458709.3458941

[30] Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James A. Landay. 2018. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (jan 2018), 1–23. https://doi.org/10.1145/3161187

[31] Paul W. Schönle, Klaus Gräbe, Peter Wenig, Jörg Höhne, Jörg Schrader, and Bastian Conrad. 1987. Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract. *Brain and Language* 31, 1 (may 1987), 26–35. https://doi.org/10.1016/0093-934x(87)90058-7

[32] A.D. Streeter, L.R. Ray, and R.D. Collier. 2004. Hybrid feedforward-feedback active noise control. In *Proceedings of the 2004 American Control Conference*. IEEE. https://doi.org/10.23919/acc.2004.1383903

[33] Zixiong Su, Shitao Fang, and Jun Rekimoto. 2023. LipLearner: Customizable Silent Speech Interactions on Mobile Devices. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3544548.3581465

[34] Xue Sun, Jie Xiong, Chao Feng, Wenwen Deng, Xudong Wei, Dingyi Fang, and Xiaojiang Chen. 2022. Earmonitor. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (dec 2022), 1–22. https://doi.org/10.1145/3569472

[35] The Engineering ToolBox. 2004. Sound Pressure. https://www.engineeringtoolbox.com/sound-pressure-d_711.html. (Accessed on 09/10/2023).

[36] Yueyang Wang, Yahui Wang, Xiaoqiong Li, Chengyi Zhao, Ning Ma, and Zixuan Guo. 2023. A Comparative Study of the Typing Performance of Two Mid-Air Text Input Methods in Virtual Environments. *Sensors* 23, 15 (Aug. 2023), 6988. https://doi.org/10.3390/s23156988

[37] Clint Zeagler. 2017. Where to wear it. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM. https://doi.org/10.1145/3123021.3123042

[38] Ruidong Zhang, Mingyang Chen, Benjamin Steeper, Yaxuan Li, Zihan Yan, Yizhuo Chen, Songyun Tao, Tuochao Chen, Hyunchul Lim, and Cheng Zhang. 2021. SpeeChin. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (dec 2021), 1–23. https://doi.org/10.1145/3494987

[39] Ruidong Zhang, Ke Li, Yihong Hao, Yufan Wang, Zhengnan Lai, François Guimbretière, and Cheng Zhang. 2023. EchoSpeech: Continuous Silent Speech Recognition on Minimally-obtrusive Eyewear Powered by Acoustic Sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3544548.3580801

[40] Wei Zhou, Wilfried Michel, Ralf Schlüter, and Hermann Ney. 2022. Efficient Training of Neural Transducer for Speech Recognition. In *Interspeech 2022*. ISCA. https://doi.org/10.21437/interspeech.2022-829