LipLearner: Customizable Silent Speech Interactions on Mobile Devices

Zixiong Su
The University of Tokyo
Tokyo, Japan
zxsu@g.ecc.u-tokyo.ac.jp

Shitao Fang
The University of Tokyo
Tokyo, Japan
fst@iis-lab.org

Jun Rekimoto
The University of Tokyo
Sony CSL Kyoto
Kyoto, Japan
rekimoto@acm.org

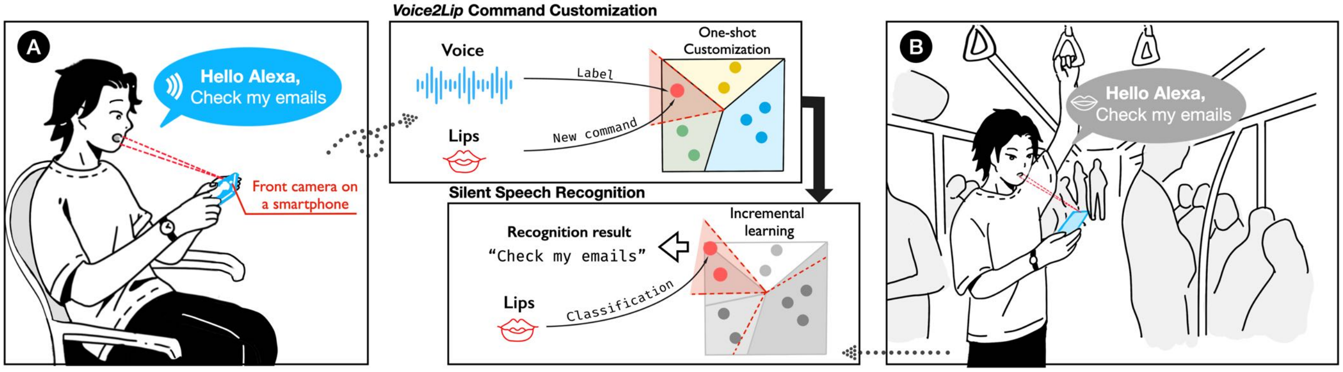
A
Hello Alexa, Check my emails
Front camera on a smartphone
Voice2Lip Command Customization
Voice
Label
Lips
New command
One-shot Customization
Silent Speech Recognition
Recognition result
"Check my emails"
Lips
Classification
Incremental learning
B
Hello Alexa, Check my emails

Figure 1: Example interaction of LipLearner. A) *Voice2Lip* in-situ command registration. The user records a silent speech command by vocalizing it once, then LipLearner automatically learns to lip-read it with the text recognized from the voice signal as the label. B) The command then can be used *without* vocalization, triggered by a silent keyword. LipLearner enables silent speech recognition which can be used in public settings (e.g., on the subway). Furthermore, it leverages incremental learning to proactively extend the model's knowledge when new samples become available.

ABSTRACT

Silent speech interface is a promising technology that enables private communications in natural language. However, previous approaches only support a small and inflexible vocabulary, which leads to limited expressiveness. We leverage contrastive learning to learn efficient lipreading representations, enabling few-shot command customization with minimal user effort. Our model exhibits high robustness to different lighting, posture, and gesture conditions on an in-the-wild dataset. For 25-command classification, an F1-score of 0.8947 is achievable only using one shot, and its performance can be further boosted by adaptively learning from more data. This generalizability allowed us to develop a mobile silent speech interface empowered with on-device fine-tuning and visual keyword spotting. A user study demonstrated that with LipLearner, users could define their own commands with high reliability guaranteed by an online incremental learning scheme. Subjective feedback indicated that our system provides essential functionalities for customizable silent speech interactions with high usability and learnability.

CHI '23, April 23–28, 2023, Hamburg, Germany
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3581465

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**; **Sound-based input / output**.

KEYWORDS

Silent Speech Interface, Lipreading, Few-shot Learning, Customization

ACM Reference Format:
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 (CHI '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3544548.3581465

1 INTRODUCTION

Conversational agents are becoming increasingly integrated into our daily lives, serving as a fundamental element of ubiquitous computing and Internet-of-Things (IoT). They facilitate our approaches to edge devices by providing intuitive and efficient interactions, allowing people to communicate directly with devices in natural language. Thanks to the tremendous prevalence of smartphones, voice assistants [10] have been unprecedentedly popular, giving users handy access to smartphone functionalities, smart home control, real-time information, and so forth. Despite the great convenience offered by voice input, there are three major limitations hampering its usability in practice. Voice User Interfaces (VUIs) 1) is not a preferred option in public settings due to the risk of privacy and security problems, and people may feel awkward talking to a smartphone in front of others [44], 2) relies on accurate speech recognition, which can be difficult when the environment is noisy, and 3) is not available for people with speech disorders.

To tackle the privacy and social acceptance issues in VUIs, Silent Speech Interface (SSI) has emerged as a promising alternative that exploits non-acoustic signals to enable speech recognition without voice. SSI provides seamless and confidential interactions in various situations, especially in those where voice interaction is inappropriate or unavailable. Recent research on SSI has proposed to use various sensing methods such as Electromyography (EMG) [31, 43], ultrasound imaging [35], capacitive sensing [40] and video camera (lipreading) [34, 45, 58] to track the movement of speech articulators and decode silent speech.

We focus on the last method, which is also known as lipreading, and augment it with few-shot learning to enable customizable silent speech commands on smartphones. Compared to other approaches, lipreading has minimal device requirements but provides rich information with high temporal and spatial resolution. Nowadays, smartphones have become the most popular devices and most of them are equipped with high-quality digital cameras. Therefore, implementing lipreading systems on smartphones further pushes forward the convenience and lowers the bar of silent speech input. On the other hand, as the primary input method on smartphones, touch gestures can be cumbersome when only single-handed input is available. For such situations, silent speech has been proven stable and efficient as a supplementary input modality [58].

However, there are three major challenges to building expressive lipreading systems in practice. First, the data collection process should minimize the effort for new users to get started with. However, previous approaches to SSIs, not limited to lipreading-based approaches, adopt a train-from-scratch model that requires collecting hundreds of samples from real users [34, 57, 58, 69], leading to excessive mental and physical user burden. Second, such data collected intensively in controlled laboratory environments causes a biased model, which can be sensitive to even minor changes in factors such as lighting, face orientations, and postures, yet there is little discussion on the model's ability to generalize to unseen environments. Third, the model training process is time-consuming and requires high-end GPUs, but they are not always accessible to users for many reasons (requiring internet connection, high computing cost, privacy concerns of uploading face data to the cloud, etc.). Adding new commands is even more difficult, because it requires collecting new data as well as re-training the model from scratch. As a result, only a limited number of pre-defined commands are available, and the rich interaction space in silent speech is still waiting to be mined.

In this research, our goal is to liberate the expensiveness of lipreading as well as reduce the user burden in the data collection process. We propose a few-shot lipreading framework that enables in-situ silent speech command customization. We set off by pre-training a lipreading encoder model using a contrastive learning objective, which learns efficient and robust visual speech representations from public datasets in a semi-supervised manner. We then employ a simple linear classifier, which can be trained in a few seconds, to transfer the model to unseen users and words using a few-shot learning strategy. Hence, the user can freely define any phrase in any language, or even non-verbal lip gestures, as a silent speech command by providing at least one sample. We further minimize the user effort of enrolling new commands by introducing *Voice2Lip*, a multimodal command registration technique that automatically learns lipreading from voice input in a one-shot manner. To register a new command, the user just says it aloud, and then our system will learn the lip movements using the text recognized from voice signals as labels.

To ensure the applicability of our method in the real world, we performed a model test under diverse conditions that covered a broad range of daily scenarios, including different lighting conditions, body postures, and holding gestures. The results show that our few-shot customization framework could achieve an F1-score of 0.8947 in unseen conditions with only one shot, significantly outperforming the conventional user-dependent approach though the latter used four times more training data. We then built a mobile application called LipLearner on a commodity smartphone.

To empower LipLearner with reliable hands-free activation, we propose a visual keyword spotting method that detects the user-defined keyword from lip movements. Most previous lipreading interfaces use the mouth opening degree (MOD) [57, 58] as the only cue to trigger silent speech input, which is prone to misactivation. For example, the system can easily respond to the user's unintentional behavior, such as smiling or talking to others. Our lipreading model encodes lip movements into embedding vectors, which can be used to identify the keyword from continuous input by computing the cosine similarity. This function can also be customized and initialized with only a few positive samples (i.e., no negative samples required). Moreover, we introduced an online incremental learning scheme to allow users to continually improve the performance of the model by providing new samples during interaction. With the efficient lip embeddings, it is trivial to fine-tune the model by only updating the liner classifier, which significantly reduces the computing resource demand and thus allows all real-time customizations and recognitions to be performed on-device for privacy preservation. Our quantitative user study results show that LipLearner could recognize 30 commands (20 of which were user-dependent) with a one-shot accuracy of 81.7%. The performance improved gradually while more samples were provided by the user; finally, 98.8% accuracy was achieved with five samples per command. Subjective feedback indicates that our system was easy to use and learn, and the human-AI interaction experience was enjoyed by many participants. We have made our machine learning scripts, models, and the source code of LipLearner (iOS App) publicly available at https://github.com/rkmtlab/LipLearner to facilitate further work.

In summary, this paper makes four key contributions:

1. A semi-supervised lipreading encoder that exploits public datasets to learn efficient visual speech representations and a few-shot silent speech customization framework to support novel commands with a small number of samples.

2. A model test demonstrating our method works robustly in a variety of environment and interaction factors, namely lighting conditions, body postures, and holding gestures.

3. A mobile application that provides real-time and customizable silent speech interactions, empowered by a visual keyword spotting method for hands-free activation and an online incremental learning scheme for extendable vocabulary and performance.

4. A comprehensive user study that evaluated the system's real-world performance and usability with customizable commands.

2 RELATED WORK

In this section, we overview related literature in the domains of silent speech interfaces and relevant machine learning techniques.

2.1 Silent Speech Interface

Silent speech interfaces have been a research topic of vast research interest for decades, aiming to provide confidential and seamless communications. Similar to VUIs, SSIs allow users to converse with computers in natural language, which provides expressive commands without requiring them to remember complicated actions or gestures. Existing SSIs are characterized by what kind of sensing methods and biosignals are used, such as tracking the movement of speech articulators using electromagnetic articulography (EMA) [13, 17, 53], vocal tract imaging using ultrasound imaging [22, 35], capturing subtle sounds produced by non-audible murmur (NAM) [59–61] and ingressive speech [15], placing capacitive sensors inside the mouth [33, 40], and capturing facial electrical activity using electromyography (sEMG) [31, 65]. In the field of Brain-Computer Interfaces (BCI), researchers seek to decode human speech directly from the electrical activity of the brain, where the approaches can be categorized into invasive systems implanted in the cerebral cortex using electrocorticography (ECoG) [1, 49] and non-invasive systems attached to the scalp using Electroencephalogram (EEG) [18, 20, 47].

The most related literature to this work is lipreading-based SSIs. Lipreading is a technology that utilizes a camera to visually capture movement around the mouth and interpret speech from the image sequence. HCI researchers have proposed to use devices such as smartphones [45, 58] and wearable cameras [6, 34, 69] to provide mobile silent speech interaction, as well as multimodal approaches such as using silent speech to facilitate eye-gaze-based selection [57].

The challenges in lipreading stem from the inherent ambiguity of lip movements. The number of distinguishable visemes (i.e., minimum visual speech units) is usually considered to be 10-16 [11, 12, 64], which is much less than the number of phonemes. Researchers have proposed using ultrasound imaging to track movements of the oral cavity and tongue as a complementary method for lipreading [30, 36, 37]. While this multimodal approach could significantly improve the performance of silent speech recognition, ultrasound imaging devices are cumbersome and impractical for mobile interactions. In contrast, our few-shot lipreading framework enables customizable and reliable silent speech interactions using only a commodity mobile phone.

2.2 Machine Learning Approaches to Lipreading Interfaces

Recent work in the deep learning field has shown the effectiveness of using deep neural networks (DNN) for lipreading [14, 42, 46], while the machine learning paradigms used to build such a model can have a significant impact on its performance.

As shown in Table 1, we broadly divided previous lipreading interfaces into two categories: 1) user-dependent models, which collect data from each user and train individual models from scratch, and 2) off-the-shelf-models, which leverage either public datasets or pre-collected data to enable user-independent recognition. User-dependent models offer better performance because they have obtained knowledge from actual users. However, this method imposes a huge burden on new users, making it a difficult trade-off between the vocabulary and the amount of training data. Off-the-shelf models are available immediately without requiring new data. Nonetheless, building a model that can generalize to unseen users remains a huge challenge, as conventional methods either have a small vocabulary [58] or limited accuracy [45, 52]. One workaround is to use a context-dependent vocabulary to improve accuracy, but it also limits the number of available commands at a time [57, 58]. Furthermore, a common issue in both user-dependent models and off-the-shelf models is that the commands are pre-defined by the researchers. Making changes to the command set requires tremendous training data and computing resources, which are not accessible to users. Additionally, there is a lack of a practical activating method to initiate silent speech input. Previous methods such as offline segmentation [6, 34, 69] or trigger buttons [45, 52] are not feasible for hands-free real-time interactions, and MOD-based methods can be vulnerable to misactivations [57, 58]. We propose a novel few-shot transfer learning paradigm to enable customizable silent speech commands. LipLearner can achieve promising accuracy with a small amount of training samples, thus making it possible for the user to add arbitrary new commands. The few-shot paradigm also opens the door for *visual* keyword spotting, which enables using silent speech to wake up devices.

The generalizability of our model benefits from the contrastive pre-training pipeline. The weak supervision thereof makes the model more flexible when transferring to a new data domain, outperforming supervised approaches [8]. Recent research on using contrastive learning for lipreading has been focusing on learning from unlabeled audio-visual data [21, 56]. Although the abundant information of audio signals provoked an array of inspiring work, such as synthesizing speech from lips [48, 66]), localizing sounds in video frames [2, 55], and separating speech signals [16], it could bring unnecessary complexity to silent speech recognition. Our work differs from the previous ones in that we leverage a more straightforward approach that only utilizes the visual modality to obtain efficient representations with rich semantic information.

2.3 Few-shot Transfer Learning in Human-Computer Interaction

Few-shot learning (FSL) is a deep learning paradigm, where a model is first pre-trained on large datasets and then fine-tuned using a few new samples to generalize to unseen data distributions. Instead of training the entire model from scratch each time, FSL can incrementally obtain new knowledge by only partially updating the model. HCI researchers have been applying FSL to tasks, such as sound recognition [29, 67] and human activity recognition [19], enabling in-situ model fine-tuning in the real world. One of the

Paradigm	Research	Device	Vocabulary	Samples	Accuracy	Activation	Language
User-dependent model	Kimura et al. 2020 [34]	Wearable camera	15	40	94%	Offline	English
	Chen 2020 et al. [6]	Wearable camera	8	10	84.70%	Offline	English
	Su et al. 2021 [57]	Fixed camera	27 (6†)	18	91.63%	MOD	English
	Zhang et al. 2021 [69]	Wearable IR camera	54/44	24	90.5%/91.6%	Offline	English/Chinese
Off-the-shelf model	Sun et al. 2018 [58]	Smartphone	44 (6-10†)	-	95.40%	MOD	Chinese
	Saitoh and Kubowaka 2019 [52]	Smartphone	25	-	73.40%	Manual	Japanese
	Laxmi and Sabbir 2021 [45]	Smartphone	51††	-	WER 40.9%	Manual	English
	Zhang et al. 2021 [69]	Wearable IR camera	54/44	-	54.4%/61.2%	Offline	English/Chinese
Few-shot transfer learning	**LipLearner (1-shot)**	**Smartphone**	**30†††**	**1**	**81.7%**	**Keyword**	**Arbitrary**
	LipLearner (3-shot)			**3**	**96.5%**		
	LipLearner (5-shot)			**5**	**98.8%**		

Table 1: Machine learning (ML) paradigms and their specifications in recent lipreading interfaces. The sample column shows the number of training samples the user needs to record for each command. † The actual vocabulary size depends on the context. †† The vocabulary is word-level. ††† The vocabulary is custom (defined by each user). While some research only conducted offline experiments or asked the user to trigger the recognizer manually, LipLearner offers online keyword activation and recognition and is evaluated via a live user study.

most relevant literature is few-shot gesture recognition [68], as gestures and lip movements are both time series human motion signals. This work utilizes the IMU signals from a smartwatch to enable users to add custom gestures with a few samples. However, the model was pre-trained in a supervised manner, which could limit the model's generalizability: although the system applied data augmentation (which was performed on a laptop) to obtain more data for fine-tuning, the 1-shot accuracy was only 55.3% in 12-gesture classification. Our approach leverages semi-supervised learning to learn more efficient representations, achieving high accuracy with a more lightweight architecture that can be deployed on a smartphone.

3 CONTRASTIVE PRE-TRAINING

To overcome the limitation of vocabulary as well as minimize the user burden in the data collection process, we leverage contrastive learning to exploit knowledge from public datasets. In this section, we elaborate on the methods and techniques we used in this pre-training process, including the large-scale lipreading dataset, the neural network architecture, and the training details. The pre-trained lipreading encoder is the cornerstone of our few-shot customization framework.

3.1 Pre-training Dataset and Preprocessing

We use a public large-scale lipreading dataset, LRW [9], which comprises video segments extracted from the BBC news, to pre-train a robust feature extractor for few-shot lipreading. The dataset consists of up to 1000 utterances of 500 different words, spoken by hundreds of different speaker, thus providing rich utterances and face patterns. The speaker's face is cropped with the mouth centered using a facial landmark detection algorithm [32] provided by the Dlib Library [38]. The dataset also covers diverse recording conditions, such as lighting, background, and camera perspective, which is expected to enhance the performance of model in real-world settings.

Nonetheless, there are still discrepancies between the data distributions of LRW and mobile silent speech scenarios. For instance, most videos in LRW were captured with fixed or stabilized cameras from a third person point of view. While in our scenarios, handheld devices, such as smartphones, inevitably lead to shaking videos, and their wide-angle lens can cause barrel distortion. Additionally, all LRW videos are sampled to 29 frames at 25fps (1.16 seconds), which can make the model sensitive to variations in video duration. To fill this gap, we apply several data augmentations to generate more data simulating smartphone videos, namely random crop, random frame drop, random shaking, and random barrel distortion. Finally, the frames were converted to grayscale and resized to 88 (H) × 88 (W) pixels.

3.2 Model Architecture

We adopt an encoder model based on the architecture proposed in [14], which has achieved a state-of-the-art level performance in lipreading classification tasks. As shown in Figure 2, the neural network first extracts both spatial and temporal information using ResNet-18 with a 3D convolutional architecture. After a global pooling layer, the output is reshaped into $T \times 512$ (T denotes time). We then apply the same word boundary technique described in [14], which appends a binary vector to indicate the duration of the keyword. Finally, the feature is processed sequentially using a bidirectional Gated Recurrent Unit (GRU) followed by an average pooling and a fully connected layer, outputting a 500-dimensional feature vector.

3.3 Contrastive Learning Pipeline

Conventional supervised learning uses labeled data to learn to classify the inputs into known classes. The vocabulary of LRW consists of 500 individual words, which is, however, biased and far from allowing natural communications with smart assistants (e.g., "Question" and "Questions" take up 2 classes, but there are no words such as "What" for interrogative expression which is essential for a conversational interface). To overcome this limitation, we leverage contrastive learning, in which the objective is to learn an embedding space where similar samples are close to each other while dissimilar ones are far apart. Thus, we can use the model to find the most similar command when given samples, even if the samples belong to previously unseen classes.

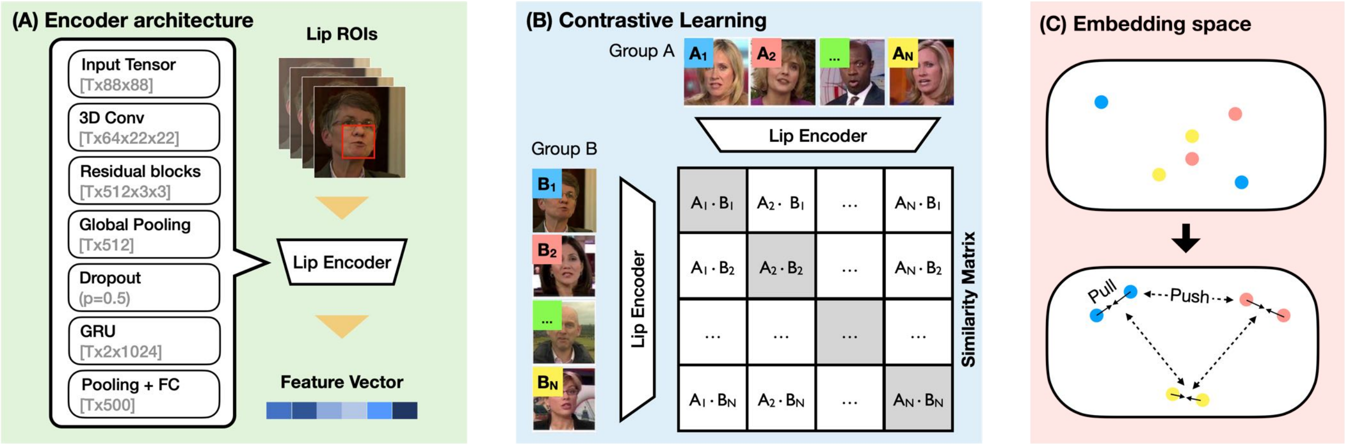

Figure 2: The pre-learning pipeline. (A) We use a 3D CNN encoder to extract a low-dimensional feature vector from lip images. (B) The contrastive objective maximizes the similarities between utterances of the same words (diagonal elements in the similarity matrix) while minimizing similarities between utterances of different words (non-diagonal elements in the similarity matrix). The subscript numbers indicate the class indexes. (C) The learned embedding space.

In our implementation, we use the CLIP objective [50] to let the model only learn the similarity between samples without remembering the exact label. As shown in Figure 2 (B), we randomly select one sample from each of N (N = batch size) classes as group A, and then select another N samples from the same classes as B. Next, the samples are encoded into embeddings, and a cosine similarity matrix is calculated among the embeddings, scaled by a temperature parameter τ:

$$S_{i,j} = sim(A_i, B_j)/\tau$$

Here, we use the same τ of 0.07 as CLIP. The cosine similarity sim(,) is measured by the dot product of two L2-normalized embedding vectors A_i and B_j, where $i, j \in [0, N]$ denote the class indexes. Note that unlike CLIP used different encoders for text and image data, our data only has the visual channel. Therefore, the encoders for the two data groups share the same weights. Diagonal values in the matrix are similarities between embeddings from the same class, while non-diagonal values are those between different classes. The model contrastively learns from the positive N pairs and the negative $N^2 - N$ pairs using the InfoNCE loss [63], which averages the cross entropy loss of group A and group B.

$$\mathcal{L} = -\frac{1}{2N}\left(\sum_{i=1}^{N} \log \frac{e^{S_{i,i}}}{\sum_j e^{S_{i,j}}} + \sum_{j=1}^{N} \log \frac{e^{S_{j,j}}}{\sum_i e^{S_{i,j}}}\right)$$

3.4 Training details

The training starts from pre-trained weights provided by Feng et al. [14]. We use a ReduceLROnPlateau scheduler with an initial learning rate of 3×10^{-4}, which is reduced by a factor of 0.5 once the validation loss stagnates for 40 epochs. The training loss converged after 500 epochs, taking around 34 hours across 2 NVIDIA GeForce RTX 2080 Ti GPUs. We save the model with the least loss on the validation set for our system.

4 DATA COLLECTION FOR MODEL TEST

There are many variables that could affect the performance of the lip encoder model. Particularly, we seek to analyze the model's robustness against challenges such as different environment configurations and user behaviors. To this end, we set off by collecting an in-the-wild dataset that covers various practical settings that simulate mobile interaction scenarios.

4.1 Command Set

First of all, we designed a 25-sentence corpus for silent speech interaction (see Figure 3). This command set is intended to contextualize a scenario where people interact with a conversational assistant to operate the smartphone, control smart home devices, or find information. The phrases are partially selected from the most popular Alexa commands according to a recent research [54], and the rest are from Apple's official guide to Siri [28]. We include both concise commands and casual expressions, covering all kinds of visemes and various lengths (3-22 visemes, average length 10.08 ± 4.47; we first translate the words to phonemes by referring to the CMU Pronouncing Dictionary [62] and then map the phonemes into visemes using Lee and Yook's approach [39]). Therefore, this corpus is also phonetically well-balanced and suitable for testing the model's capability.

4.2 Recording Conditions

A mobile interface should provide stable performance across different conditions. Especially, we consider that there are three key factors, namely lighting, posture, and grasp gesture, that pose challenges to silent speech recognition. In this section, we elaborate on the various recording conditions contained in the dataset.

4.2.1 Lighting. We change the recording locations and time of day to achieve different luminance levels. Further investigations show that those daily scenarios can have a wide light intensity range.

[Music and Podcasts]		[Calls and Texts]		[Smart Home]	
1. Play	3	6. Call mom	6	11. I'm home	5
2. Stop	4	7. Call Rick	6	12. Turn on lights	9
3. Next	5	8. Text dad	8	13. Close the shades	9
4. Volume up	8	9. Emergency	8	14. Watch Netflix	11
5. Volume down	9	10. Send an e-mail to John	15	15. Warm it up in here	13

[Information and Navigation]		[System Control]	
16. What time is it	10	21. Take a picture	9
17. What's the weather	10	22. Open Twitter	9
18. What's the news today	13	23. Turn on flashlight	12
19. Get directions home	14	24. Increase brightness	13
20. Where's the closest gas station	22	25. Set a timer for 5 minutes	20

🟪 Command intent
🟧 Number of visemes

Figure 3: Command set used for the model test.

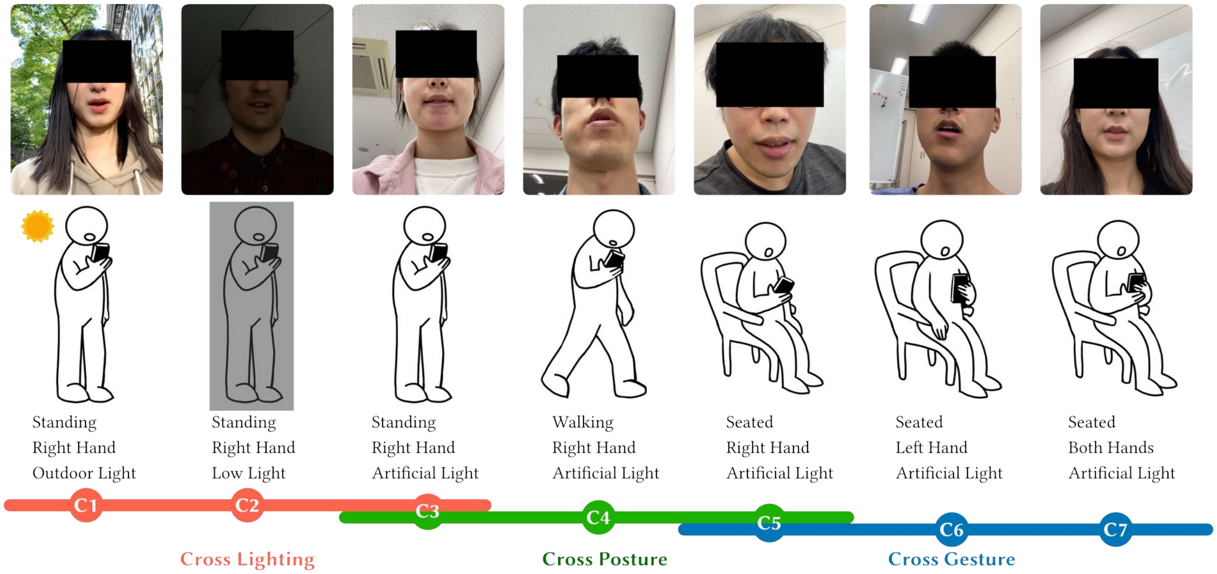

Figure 4: Illustration of seven conditions during data collection and their corresponding captured views. Selected frames are processed for privacy protection. The recording conditions are intended for cross-lighting, cross-posture, and cross-gesture tests.

- **Outdoor Daylight**: outdoor environment on sunny afternoons (1:00 PM - 3:00 PM).
- **Low Light**: laboratory environment on later afternoons (3:00 PM - 5:00 PM), simulated by partially blocking the natural light.
- **Artificial Light**: laboratory environment with good lighting provided, natural light is blocked.

4.2.2 Posture. Participants were asked to record while seated, standing, or walking. Different postures could cause different levels of shaking, leading to blurry videos and varying face positions.

- **Standing**: participants are asked to stand in place.
- **Walking**: participants are asked to record while walking along a straight line.
- **Seated**: participants are seated on a chair with their hands placed on the armrest.

4.2.3 Grasp Gesture. Participants were asked to hold the smartphone with their right hand, left hand, or both hands. Different grasp gestures result in significant differences in the face orientation relative to the camera.

- **Right Hand**: the smartphone is held with the user's right hand.
- **Left Hand**: the smartphone is held with the user's left hand.
- **Both Hands**: the smartphone is held with the user's both hands.

4.3 Procedure

We recruited 11 participants (4 females and 7 males) from the local university, all right-handed. Note that to distinguish from the user study section, participants in this section are identified as *speakers* (S1-S11). We used iPhone 11 and iPhone 13 Pro for video recording. All videos are saved in a MOV format with 1080 (H) × 1920 (W) pixels at 30 fps. In the collection process, the user is asked to press the record button at the bottom of the screen and then subvocalize the command prompt shown on the top. During the speech, the user needs to keep pressing the recording button and release as soon as they finish speaking to indicate the beginning and the end of the recording. Next, the subsequent command will be prompted. To avoid errors caused by unfamiliarity, we ask the user to read each of the commands at least once before collecting. If the user did not read the command correctly or fluently, they can use the rollback button at the bottom-right corner to record the last command again.

The data collection was approved by the university's Institutional Review Board (IRB), and all participants have filled out an IRB-approved consent form. All participants completed seven collection sessions, each of which is under a condition that is a combination of the three key factors (see Figure 4). For each session, participants were tasked to repeat each of the 25 commands five times. Between the sessions, participants were allowed to take a one-minute break. This procedure took around 40 minutes, and we compensated the participant 1050JPY for their time. In total, 11 participants × 7 sessions × 25 commands × 5 repetitions = 9625 data points were collected.

5 CUSTOMIZATION PIPELINE AND MODEL PERFORMANCE

This section presents the few-shot tuning pipeline used to recognize novel silent speech commands with very few samples. Furthermore, we performed a comprehensive test to show that our approach is robust to a wide range of environment configurations.

5.1 Pre-processing and Data Visualization

We extracted the mouth region from our study data using the MediaPipe face detector [41] to identify the face landmarks. For each frame, we cropped a square region of interest (ROI) with the mouth centered according to the landmarks, which describes the location of the mouth. The ROI was converted to a grayscale image and then resized to 88 (H) × 88 (W) pixels, which follows the same pre-processing procedure as the LRW dataset. With the pre-trained lip encoder model, we embedded the ROI into a 500-dimension feature vector as a semantic representation of the silent speech command.

To better understand how the feature vectors are distributed, we use the uniform manifold approximation and projection (UMAP) to visualize a subset of data obtained from a single speaker (S10) in a 2D space. UMAP is an unsupervised dimensionality reduction technique that clusters the data points without accounting for the labels in the transformation. As shown in Figure 5, there are 25 distinct clusters corresponding to the 25 commands in the command set, which are linearly separable. In addition, our model exhibits a good generalization ability. For example, when zooming into two of the clusters (*"Call mom"*, and *"Volume up"*), it was unlikely to separate the data by the recording condition. Moreover, the distance between different conditions was much larger than that between different commands. Similar observations were also found in other users' collected data, which supports our assumption that the encoder model has learned efficient semantic representation that can be generalized to unseen speakers and phrases.

5.2 Few-shot Fine-tuning Architecture

Instead of directly computing the similarity, we used a simple linear logistic regression classifier, which is shown sufficient to achieve high accuracy with a very small amount of training samples [7, 8], to learn novel commands. Logistic regression is adept at fitting linearly separable data, which is suitable for the highly abstracted features extracted by the encoder model. In the fine-tuning stage, we freeze the weights of the encoder model and only train the linear classifier, thus making it trivial to perform in-situ command customization on mobile devices. Note that the linear classifier is user-dependent and trained on each user's data to maximize accuracy.

To better understand the capability and limitations of the silent speech representations, we conducted comprehensive experiments to test the model's performance in different dimensions.

5.3 Experiment 1: Effect of number of commands and number of shots

Our in-situ customization framework allows the user to enroll new commands or provide new samples for existing commands anytime and anywhere. We used our dataset to simulate this dynamic process and investigated how the number of commands and shots would affect recognition accuracy. In this session, we first randomly selected M commands ($M \in \{5, 10, 15, 20, 25\}$). The last two shots from all conditions are selected as test data. We then trained the model with N ($N \in [1..10]$) shot(s) randomly selected from the remaining data, which can belong to different conditions. Since there are too many possible combinations of data selection, we repeated the test 1000 times to simulate that training data is collected over various conditions in daily use. As illustrated in Figure 6, The model's performance improved rapidly as the number of shots increased. In 5-command classification, the F1-score was 0.9574 ± 0.0286 with only one shot and became 0.9924 ± 0.0058 with three shots of each command. Compared to other input modalities (e.g., gesture, eye gaze), one of the most important advantages of speech is its expressiveness. Therefore, supporting more commands is crucial to providing better silent speech interactions. The result showed that although more commands led to slight performance degradation, the model still obtains a one-shot F1-score of 0.8947 ± 0.0530 when classifying 25 commands and an F1-score of 0.9819 ± 0.0120 was achieved with four shots. The standard deviation was also reduced when the number of shots was increased, indicating that more training samples can improve the model's robustness. Thus, the proposed method is promising for recognizing a large number of silent speech commands, and the model's knowledge can be extended by collecting more data in real use.

5.4 Experiment 2: Generalization ability

A common scenario is that the recording setting is significantly different from where the user actually uses it. The model can learn

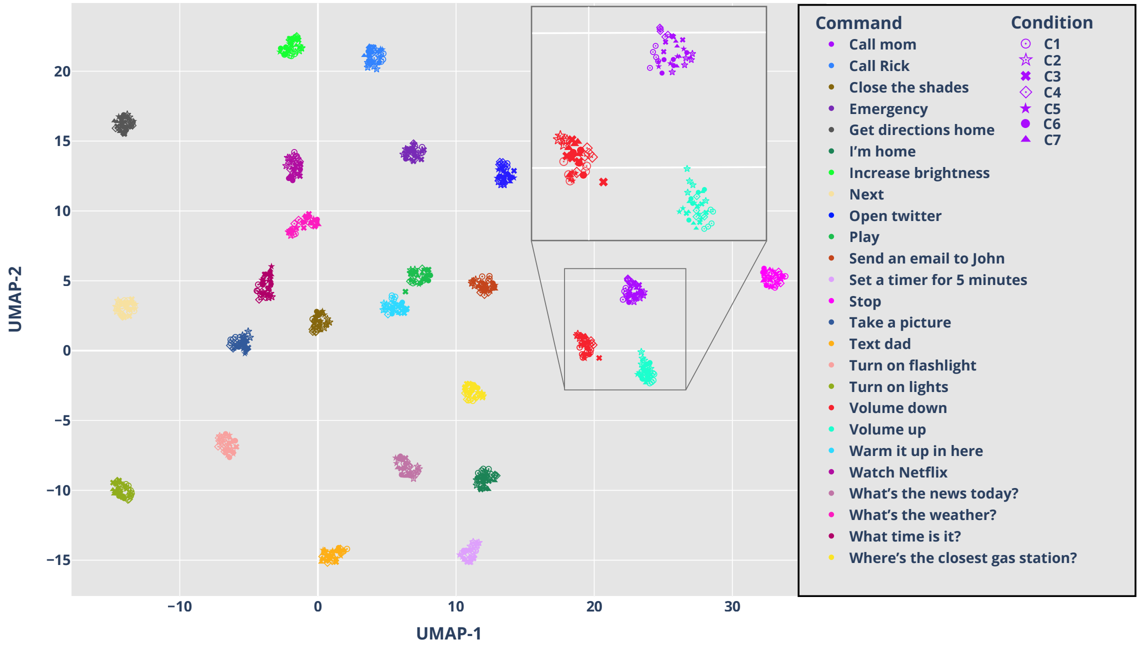

Figure 5: 2D UMAP Visualization of the feature embedding space with data from S10 as an example. Commands and conditions are depicted in colors and symbols, respectively. The zoom-in area shows that the data distributions of the same command from different conditions are mostly overlapped, suggesting that our visual speech representation is robust to environment factors.

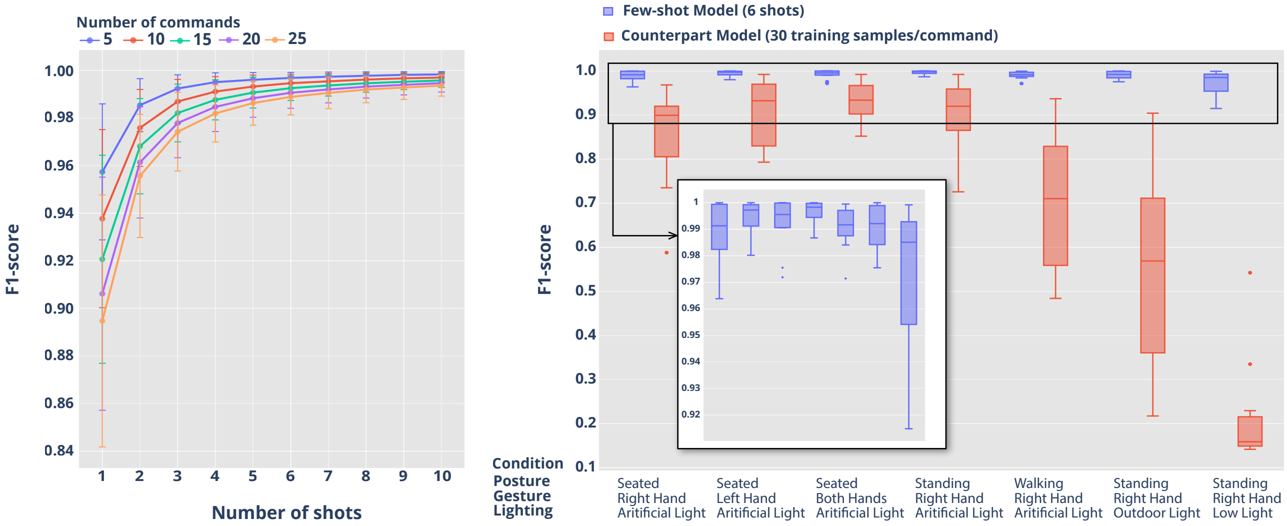

Figure 6: Model test results in F1 measure. Left: Effect of the number of commands and the number of shots. Right: Generalization ability test.

these differences by asking the user to provide samples in every possible condition, which however leads to user burden implications. We believe that our approach can be generalized to completely unseen conditions without having such training data. First, we ran a leave-one-condition-out test by training the classifier on data from six conditions and testing on data from the one remaining condition. For each training condition, we randomly selected only one sample from each class, forming a 6-shot training dataset. This test was repeated 100 times with random seeds. The box plot in Figure 6 illustrates the distribution of the F1-scores for the 11 participants. To

		Number of shots				
	Left-out condition	1	2	3	4	5
Cross Lighting	**Average**	0.8954	0.9227	0.9391	0.9463	0.9504
	Artificial Light	0.9079	0.9344	0.9509	0.9587	0.9629
	Outdoor Daylight	0.8876	0.9125	0.9283	0.9347	0.9391
	Low Light	0.8906	0.9212	0.9382	0.9454	0.9493
Cross Posture	**Average**	0.9189	0.9436	0.9595	0.9665	0.9702
	Standing	0.9291	0.9504	0.9637	0.9697	0.9717
	Walking	0.9183	0.9431	0.9601	0.9669	0.9717
	Seated	0.9093	0.9374	0.9546	0.9629	0.9674
Cross Gesture	**Average**	0.9332	0.9555	0.9680	0.9746	0.9780
	Right Hand	0.9162	0.9425	0.9568	0.9646	0.9689
	Left Hand	0.9456	0.9632	0.9739	0.9797	0.9823
	Both Hand	0.9377	0.9609	0.9733	0.9796	0.9828

Table 2: Cross-condition model performance in F1 measure.

compare with the predominant approach, which trains the model from scratch with considerable data collected from real users, we built a counterpart model that had the same architecture as our encoder but was trained in a supervised fashion. The counterpart model was trained on all data obtained from the training conditions (i.e., 6 conditions × 5 repetitions = 30 training samples per command), and it corresponds to the user-dependent train-from-scratch model in previous literature. Overall, our method achieved an F1-score of 0.9895 ± 0.0078 (averaged over conditions), surpassing the counterpart model's score of 0.7147 ± 0.2576. This result shows that our method provides significantly higher recognition accuracy and is more robust to unseen environments. In addition, the counterpart model exhibited worse performance especially in the last three conditions: walking posture (F1-score = 0.6930), outdoor light (F1-score = 0.5510), and low light (F1-score = 0.2134). This indicates that the accuracy of the conventional train-from-scratch method can be most severely affected by shaking videos and varying illuminations. To investigate our method's capability to cope with this problem, we further conducted cross-condition experiments with control variables in the following sections.

5.5 Cross-condition Performance

People use smartphones in different places and at different times, leading to varying lighting conditions that can significantly affect the video's quality. For example, insufficient lighting requires longer exposure time and higher sensor sensitivity, which can result in blurry images with noise. In contrast, bright sunlight can cause overexposed images that lacked highlight details. We select the data recorded under conditions C1, C2, and C3, corresponding to outdoor daylight, low light, and artificial light, respectively, while the keeping posture and grasp gesture are fixed to standing and right-hand holding. A cross-lighting test was conducted by training the classifier under two conditions and testing under the other condition.

The gesture of holding a smartphone depends on personal habits and the usage scenario. As a result, the camera angle relative to the face can vary in a wide range, causing different distortion effects in the image. We ran a cross-gesture test across conditions C3, C4, and C5, corresponding to standing, walking, and seated postures, respectively. While the gesture and lighting were set to right hand and artificial light.

Similarly, posture is also a vital factor in mobile lipreading, taking a video while walking leads to frequent camera angle changes and shaking videos with blurry frames. The cross-posture test was performed across conditions C5, C6, and C7, where the user was seated under artificial lights but with different gestures, namely right hand, left hand, and both hands.

All cross-condition tests were repeated 1000 times to mitigate the randomness of data selection. The results are shown in Table 2 with all conditions showing a similar trend: the more shots, the better performance. We also find that the cross-lighting condition was more challenging, as its 3-shot average F1-score was 0.9391, which was notably lower than the cross-posture and cross-gesture conditions (F1-score 0.9595 and 0.9680). Overall, we conclude our framework still shows high and robust performance even in unseen conditions, which is promising for real-world applications.

6 LIPLEARNER: CUSTOMIZABLE AND LEARNABLE SILENT SPEECH ASSISTANT

To investigate the usability of our silent speech customization method, we implemented LipLearner, a mobile application for in-situ customizable silent speech interaction with online few-shot learning. In this section, we elaborate on the implementation details of the application, including visual keyword spotting (KWS), online learning scheme, and interface design.

6.1 Visual Keyword Spotting

Detecting and segmenting the user's silent speech has been challenging in real-time lipreading. Previous researchers have proposed to activate the recognition algorithm by using the opening degree of the mouth to identify silent speech [57, 58, 69]. However, this approach is prone to misactivation because it can be easily confused when the user is talking to others or unintentionally opens their mouth.

We propose a few-shot visual keyword spotting method by leveraging the efficient representations extracted by our lip encoder

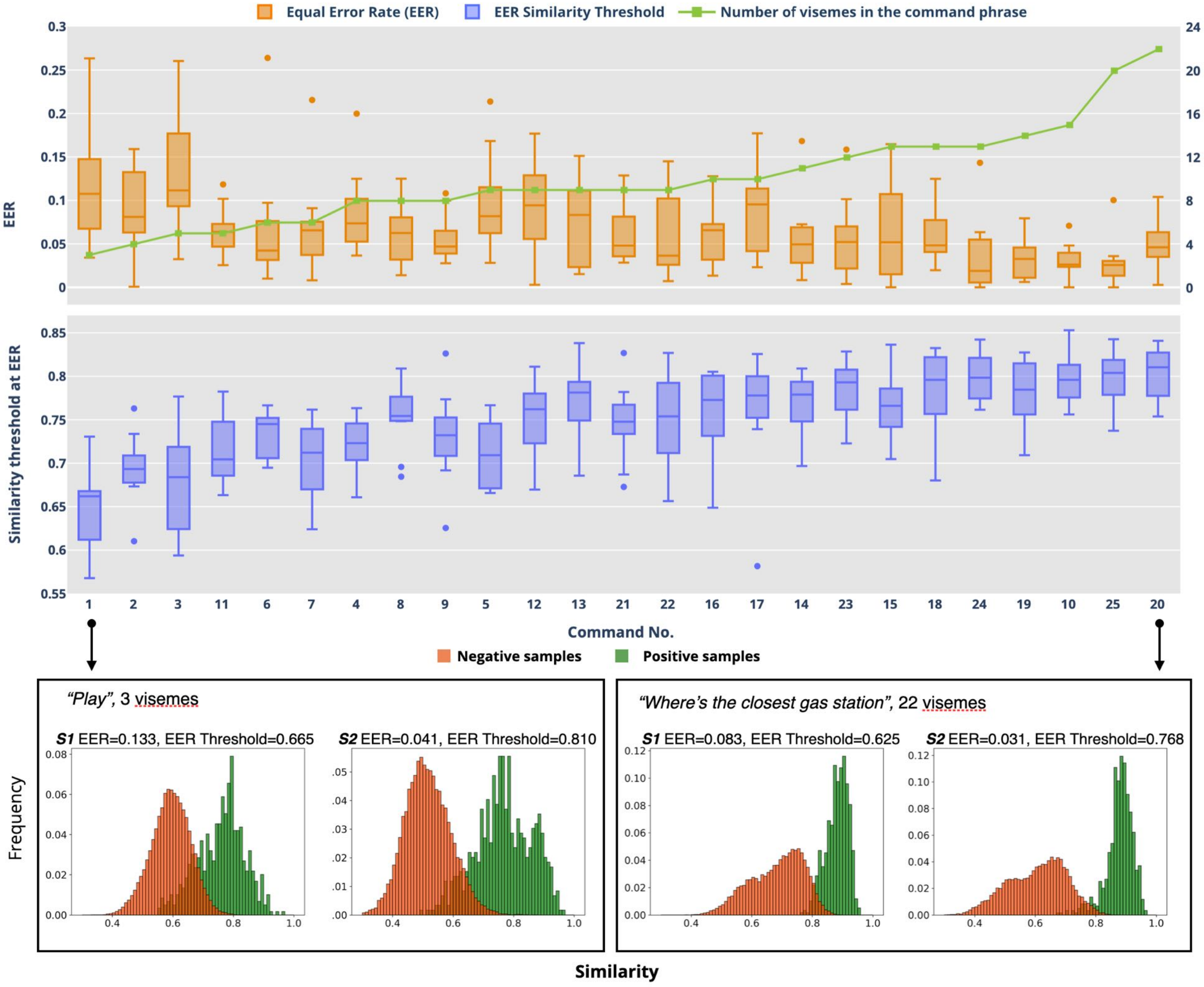

Figure 7: Top: Per-command EER and EER similarity threshold. We find that commands consisting of more visemes have higher EER and EER thresholds. Note that for better visualization, the commands are sorted by the length in viseme. Bottom: an example illustration of the positive and negative distributions of command "Play" and "Where's the closest gas station," subvocalized by S1 and S2.

model. Although KWS as an activation method has been predominant in voice interactions, to our best knowledge, this technique has not yet been applied in lipreading-based interfaces. Building a KWS model usually requires a huge number of positive and negative training samples, and it is difficult to provide user-defined wake-up keywords. We exploited the generalization ability of the encoder model, which is obtained during the contrastive pre-training, to enable silent keyword detection with customization and rapid calibration. To initialize the KWS module, the user registers a customized phrase as the keyword. Our system then calculates the similarity between the user's real-time lip movements with the keyword utterance sample, (i.e., the cosine similarity between the normalized vectors), thereby spotting when the user is issuing the keyword by comparing the similarity value with a specified threshold. Thus, our technique is available with very few keyword samples and no negative samples.

To determine the optimal threshold for keyword spotting, we leveraged our dataset to estimate the equal error rate (EER) threshold by discriminating one command (deemed as positive samples) over the other commands (deemed as negative samples). The EER results and the corresponding similarity threshold of each command, averaged over all participants, are shown in Figure 7. Overall, our method achieved an average EER of 6.75% (standard deviation 2.53%). In addition, the number of visemes in the command had a negative correlation with the EER (r=0.688), and an even stronger positive correlation with the EER threshold (r=0.852). This result suggests that using commands with more visemes (i.e., having more complicated lip movements) as the wake-up keyword can yield a lower error rate, but also requires a higher similarity threshold.

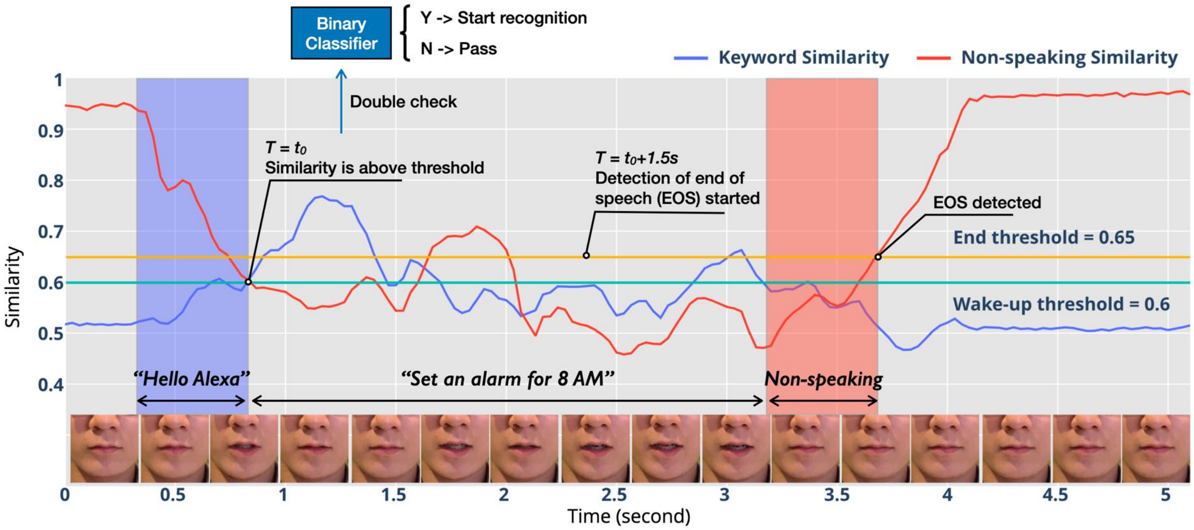

Figure 8: An example that illustrates the visual keyword spotting technique when the user says "Hello Alexa, set an alarm for 8 AM." When the similarity between the window input and the keyword is above a threshold of 0.6, an additional binary classifier is used to re-examine whether the keyword is issued. If so, the system starts to recognize the following speech as a command. After 1.5s (3 window step size), the system starts to detect the end of the speech (EOS) by calculating the similarity between the window input and the non-speaking sample with a threshold of 0.65.

On the other hand, the optimal threshold can vary widely among individuals. For example, for command No.16 *"What time is it"*, the optimal thresholds for P9 and P10 were 0.649 and 0.805. To better understand the data distribution, we visualized the similarity frequency of *Play*, the command with the lowest EER threshold, and *Where's the closest gas station* the command with the highest EER threshold, by using the data from S1 and S2.

Based on these observations, we concluded that although a high keyword spotting accuracy can be achieved using similarity alone, the practical performance optimal threshold can vary considerably depending on the length of the phrase in viseme and the pattern of the user's speech. Therefore, we adopted a relatively low threshold of 0.6, which can accept almost all positive samples over all users and commands in the dataset while still rejecting most negative samples. We employed another logistic regression binary classifier to perform a rapid calibration to reduce false positives to discriminate in actual use. As shown in Figure 10 C, the user can report when a false positive occurs, and the utterance that has misactivated the system will then be learned as negative. Fortunately, as demonstrated in Figure 8, the similarities between non-speaking lip movements were significantly higher, making it much easier to spot the end of the silent speech input. Therefore, we only used a similarity threshold of 0.65 without additional classifiers. Furthermore, we set a maximum utterance length of 4s, which means the system will automatically stop recording and perform recognition when the input is longer than 4s.

In real-time use, we used a sliding window of 30 frames (assuming 1s) to extract feature vectors over time. Suspected keyword utterances were detected using the similarity threshold and re-examined using the additional binary classifier. If the utterance is classified as positive, the system is activated and will recognize the subsequent input as a command. Since there is usually a pause between the keyword and the command, the system will start to detect the end of the utterance after a delay of 1.5 times of window length (approximately 1.5s).

6.2 System Implementation and Online Incremental Learning Scheme

We developed an iOS application on an iPhone 13 Pro as a proof-of-concept prototype of LipLearner. The video stream from the front camera was first cropped into the ROIs by using the Vision [27] framework to detect the face and lips. The PyTorch-format lip encoder model was converted into the Core ML [23] format, which extracts feature vectors from the ROIs. Finally, we employed the MLLogisticRegressionClassifier of the Create ML [24] framework to learn the vectors for keyword spotting and silent speech command recognition. The system latency was approximately 250ms feature extraction for 30 frames + 172ms classification ≈ 422ms, which is sufficient for real-time interactions. Note that all recognition and fine-tuning processing is done on a commodity mobile phone. Thus, LipLearner can be used without network connections and has all data stored locally, addressing the privacy concerns in lipreading.

Model tests in section 5.3 have shown our method can exploit multiple shots for more accurate and robust recognition. To apply this ability in practice, we designed an incremental learning scheme that continuously learns from new data to maximize accuracy (Figure 10). The interaction design of LipLearner can be divided into the following four stages.

6.2.1 Initialization phase. To start with, LipLearner will require the user to set up the KWS system for activation and speech segmentation. The user can record several keyword and non-speaking samples by holding the record button at the bottom of the screen. Feature vectors will be extracted from these samples, and the average vectors of each will be used to calculate the similarity for detecting keywords and EOS. As described in Section 6, we also initialized the additional binary classifier with those samples to re-examine suspected keywords. In the subsequent stages, users can report misactivations to improve the KWS classifier.

6.2.2 Command registration mode. The user can create novel commands at any time by switching to this mode. To offer a more accessible command registration, we incorporate speech recognition to automatically learn new commands from the voice input using the built-in speech recognizer on iOS 16 [25]. Figure 10 B illustrates the registration mode. When the user speaks the new command aloud, LipLearner will record the lip movements and prompt the text recognized from the voice signal as the label. The user can make corrections to the text if incorrect, or just manually input the label if vocalizing is not preferred. Note that to maximize the accuracy, the registration phase also requires the user to first wake up the system using the keyword.

6.2.3 Active learning mode. When the quantity of training data is small (e.g., less than 3 shots), the user can use the system in the active learning mode to improve the model. The system will proactively solicit new data by asking the user to confirm whether the prediction is correct, if not, the user needs to select the correct label from existing commands. Since we only need to re-train the logistic regression classifier part of the model, after new samples are collected, the user can perform on-device fine-tuning at any time. We report that this process can be finished in 2217ms (10-test average) with 30 commands × 5 shots = 150 samples as training data, suggesting that it is possible to update the model in an in-situ manner.

6.2.4 On-demand Learning Mode. If the user thinks that the model has already achieved high performance, they can use the on-demand Learning mode, where the system does not actively collect any data. Instead, the user can choose to correct and add only the misrecognized samples. This mode requires the least effort and prevents the classifier model from overfitting.

7 USER STUDY

We conducted a user study to evaluate LipLearner's usability. This study is distinct from the model test because the silent speech command is issued in real-time and segmented by the KWS algorithm. Furthermore, we wanted to investigate whether our method is able to recognize user-created commands, which can be meant for different intentions with diverse expressions, even in different languages. Finally, it was also important to observe the user's behavior in our human-involved online learning process.

7.1 Participants and Apparatus

We recruited 16 participants experienced in using voice assistants to use LipLearner. The participants' native languages are ranging from English, Chinese(including Mandarin, Cantonese, and Hakka), Spanish, Japanese, Malay, and French. This user study also got approved by the university's IRB and all participants were paid 2100 JPY for compensation.

An iPhone 13 Pro running the LipLearner application was used as the apparatus for the user study. The participants were seated in an armchair and encouraged to hold the phone in the usual way.

7.2 Design and Procedure

The user experience design of LipLearner is shown in Figure 10 and our user study is consistent with it.

Participants were first given a brief introduction to the system and the interface, after that they were asked to define their wake-up keyword in the format of *"Hello, X"*, where "X" is their preferred name for a smart assistant. Since we have found that phrases with more visemes can provide better KWS performance, "X" was limited to those have more than 3 visemes. Next, participants initialized the system by recording keyword samples and non-speaking samples three times each. Then participants were given five minutes to get themselves familiar with LipLearner by using the activation, command registration, and recognition functions. After participants had sufficiently practiced, they were asked to define their own command set in advance. The command set for user study was divided into three categories based on the level of creative freedom they permit, listed in ascending order as follows:

1. **Pre-defined**. We pre-defined 10 English commands (Table 3 in appendix). Participants were asked to register each command exactly as it is.
2. **User-described**. We illustrated 10 scenarios where smart assistant could be used (see Figure 9 and Table 4). Participants were asked to use their own words to describe the command they prefer to say in the scenario. There were no restrictions on the language.
3. **User-created**. Participants were asked to freely create 10 commands with no restrictions or guidance. (Table 4).

Participants registered the 30 commands in one shot using the *Voice2Lip* technique by speaking aloud *"Hello [Name], [Command]"*. Alternatively, they could also choose to input the label manually in

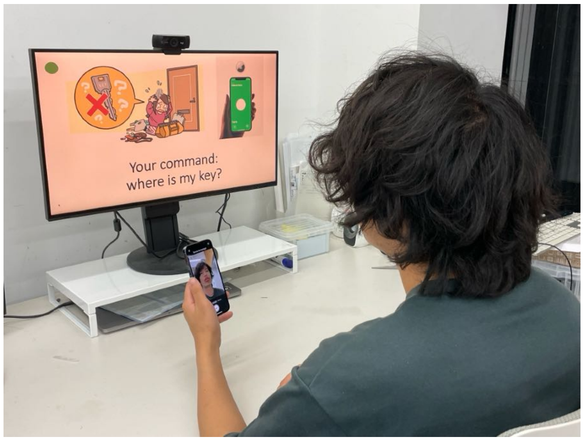

Figure 9: The user is registering a user-described command that is defined with the guidance of an illustration.

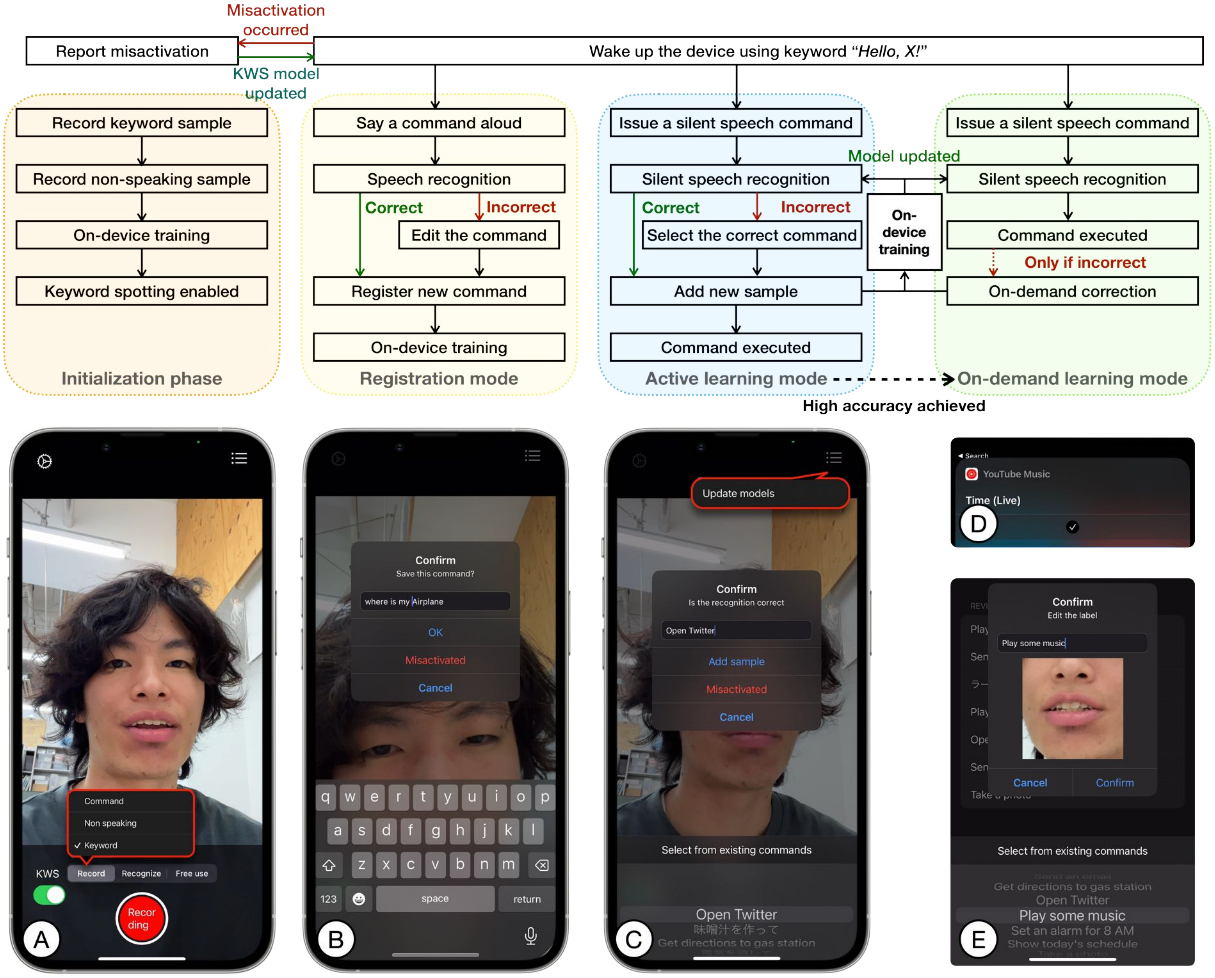

Figure 10: User experience and interface design. (A) The interface of the initialization phase. The user first needs to record keyword and non-speaking samples to enable KWS activation. (B) The user says a command aloud for command registration. The voice signal will be leveraged to label the silent speech, allowing fast command registration (*Voice2Lip*). (C) The interface for querying the right label in the active learning mode. Users can slide through the existing commands sorted by similarity to select and add a new sample to the model. Users can update the model at any time by using the button at the upper-right corner, which usually takes around 2 seconds on iPhone. (D) An example showing the command "play some music" is recognized correctly and executed successfully by the pre-set shortcut. (E) The interface for correcting the predictions in on-demand learning mode. The user can review recent utterances displayed as a GIF animation.

cases where they preferred to do so or the speech recognition was not functioning correctly.

After finishing command registration, participants had a live test session to test LipLearner's performance over six trials. During the test, the experimenter could be directly consulted for clarifications when desired. In each trial, the participant issued each of the 30 commands once. The command to be issued was prompted on a 27-inch monitor in random order. To evaluate the effectiveness of the online incremental learning scheme, the application was set to active learning mode to collect new data from each recognition. If the recognition result was correct, the participant was asked to tap the "add sample" button. Otherwise, they were asked to first select the correct label for the command and then tap the "add sample" button. Upon completion of each trial, LipLearner would obtain a new sample for each command. The participant then could update the model with the new samples by tapping the update button at the top-right corner. In this test, the recognition results were shown on the top of the screen without command execution. We also wanted to verify whether the patterns of lip movements in voiced (normal) speech and silent speech are different, and whether this potential difference would lead to inconsistent recognition performance. To do so, in the first two trials, participants were asked to say the command either in voiced speech or silent speech. The order of the voice trial and the silent trial was counterbalanced

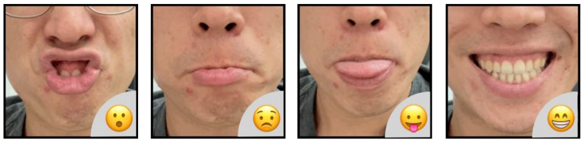

Figure 11: Facial expression registered as emojis by P10.

among participants. To avoid effects on subsequent trials, only the samples from the silent trial were used for incremental learning as the second shot.

After the six trials, participants were given 5 to 10 minutes to use LipLearner freely in on-demand learning mode, where they can optionally correct misrecognized commands for better performance. As a proof-of-concept system, we pre-configured the 10 pre-defined commands with the Shortcuts [26] function on iOS, while the other 20 custom commands would still only show the recognition result. We encouraged the participants to experience all pre-configured shortcuts at least once. Finally, they filled out a System Usability Scale (SUS) [4] questionnaire before attending a semi-structured interview about the experience of using LipLearner.

8 RESULTS

8.1 Observations

In order to better understand the effect of LipLearner and seek new insights, we noted down the observations during the user study.

Overall, all participants used the LipLearner smoothly to register and issue silent speech commands. They have personalized LipLearner's names and defined a wide variety of commands (see Table 4 in appendix). All non-native English speakers customized the commands in their native language, and 4 participants used more than 2 languages. P12 even used 5 languages to customize commands. This linguistic diversity and promising performance suggest that LipLearner holds the promise of enabling arbitrary language for silent speech.

In the case of user-defined commands with given scenarios, although some were relatively similar or even the same (e.g. P2, P5, P9 used exactly the same command "关灯" in Chinese), the participants used the expressions that fit their language and speaking habits most. As for the user-created commands, the great richness indicates that LipLearner can exploit much expressiveness of lipreading.

Some of the participants noticed that LipLearner recognized correctly even if they did not say the commands exactly the same as the commands they have registered. For example, the registered *"what's the weather today"* can be used as *"What's the weather like today"*. The model shows some certain tolerance in all language tested, particularly for minor changes in mid-sentence and end-of-sentence. This nature demonstrates the affinity to real scenarios in which people will register more than 30 commands and may not precisely remember every command.

In the free-use session, P10 tried recording four facial expressions as commands (Figure 11 A) and labeled them with emojis. Since this interesting behavior was never observed before, the experimenter noted down the following recognition results of those expressions. Note that those expressions were recorded in a one-shot manner and classified along with the existing 30 commands. Our system correctly recognized 9 out of 11 tries, and the participant commented, *"It knows what expression I'm trying to make! It's so fun!"* This revealed LipLeaner's potential in recognizing non-verbal commands, which will be discussed later in Section 9.1.

8.2 Quantitative Results

8.2.1 *Keyword Spotting performance.*

We logged the number of misactivations and false negatives in each trial and depicted it in Figure 12 (A). The FPR began from 0.26% in the first trial and decreased rapidly as the user reported more misactivations, finally achieving 0.07% with approximately 7 samples. This result indicates that although the KWS function was initialized with only positive samples, it could provide good performance in an early stage and learns efficiently from negative samples over time.

The average false negative rate (FNR) across 7 trials was 1.43% without notable changes (standard deviation is 0.45%), because we did not collect positive samples for keywords except in the initialization phase. Note that a lower similarity threshold can reduce false negatives. Although it may also lead to a higher false positive rate (FPR), we think it is admissible given LipLearner's remarkable ability to cope with misactivation. However, since determining the best threshold for all users is impossible, future work should open this setting to the user's choice.

8.2.2 *Overall Recognition Performance.*

As shown in Figure 12 (B), first, we find that the one-shot model whose training data all comes from voice input had better accuracy in recognizing vocalized utterances (87.29% ± 10.42%) than recognizing unvocalized utterances (81.67%±12.80%). This suggests that voiced speech and silent speech can have different patterns in lip movements, and learning silent speech from normal speech led to a slight drop in classification accuracy. However, in the post-experiment interview, all participants still expressed a preference for *Voice2Lip* when registering new commands, while using the keyboard to input the command label was considered only when speech recognition fails. Therefore, we believe that sacrificing approximately 5.6% accuracy in 30-command classification to expedite the command registration process is acceptable.

Furthermore, LipLearner could efficiently expand its knowledge with new samples, which is consistent with the result of the model test. The accuracy rose from 96.04% ± 4.12% with 3 shots to 98.75% ± 2.60% with 5 shots. Notably, 14 out of 16 participants achieved 100% accuracy within 5 shots. Most participants favored the on-demand Learning mode because the accuracy was sufficient after finishing the active learning phase and they felt confident using the system ([P7, P9, P15]). To highlight the effect of the online incremental scheme, we simulated a situation where the model did not learn new data during the experiment (Figure 12 (B)). We evaluated the system with the same data collected from the user study, while the model was maintained to be the first one-shot model. The result shows that the performance does not improve as the number of trials increases, suggesting that the performance improvement was accomplished solely by incremental learning, instead of the user's familiarization of saying the commands.

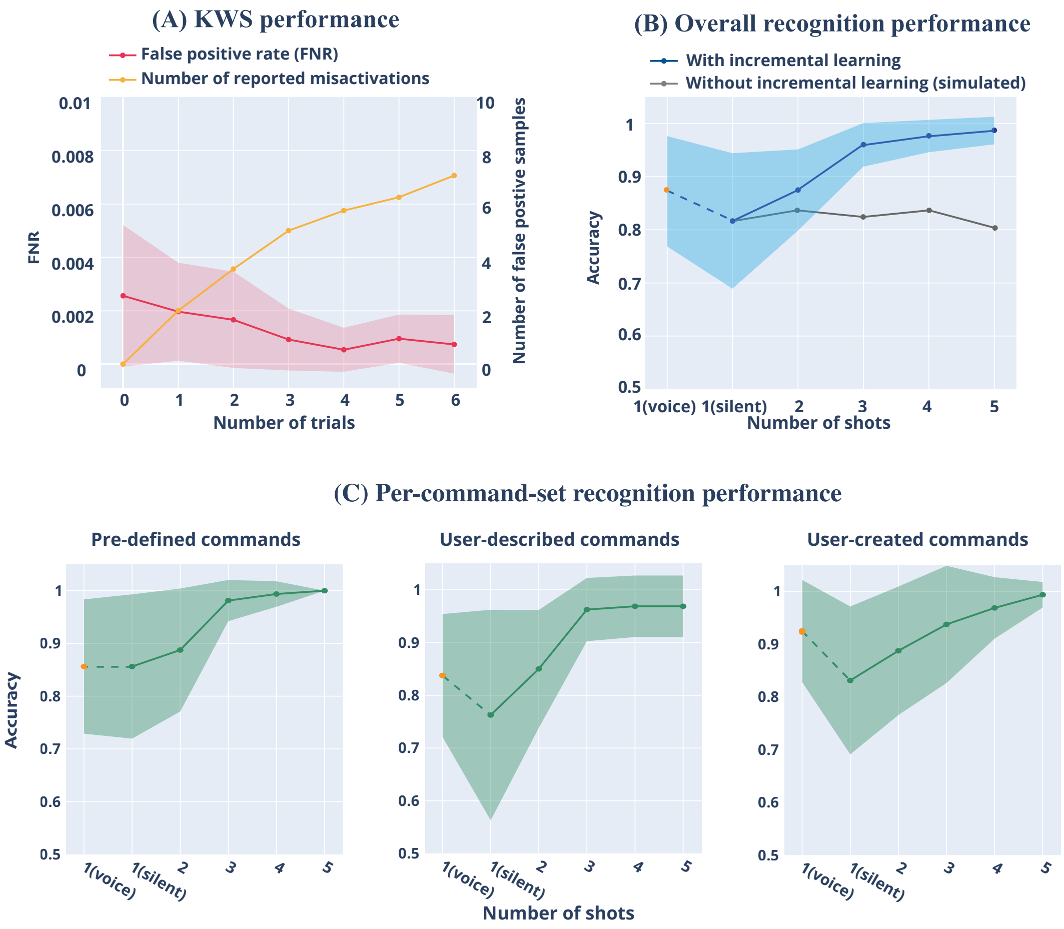

Figure 12: The false positive rate and recognition performance of LipLearner.

8.2.3 Per Command Set Recognition Performance. We examined whether LipLearner could provide consistent performance regardless of how the commands were defined by calculating the recognition accuracy in a per-command-set manner 12. In the first silent trial, where the model only used one shot for training, LipLearner achieved better performance on the pre-defined and user-created commands (average accuracy 0.8646 and 0.8500), while the accuracy on user-described commands was lower (average accuracy 0.7646). Considering the findings in Section 6, we speculate this difference is caused by the command length. We observed that in the user-described part, participants tended to use short but concise commands to follow the guidance in the illustrations, such as *"Call mom"* and *"Find my car"*. In contrast, user-created commands were longer, more casual, yet full of creativity, e.g., "What are you doing in my swamp!" and "さっきの写真をインスタグラムにあげて (Post my recent photos to Instagram)". The gap among different command sets was closed substantially as more samples were provided. Eventually, all accuracies became above 99% with 5 shots, demonstrating LipLearner's ability to learn complicated commands in different languages efficiently.

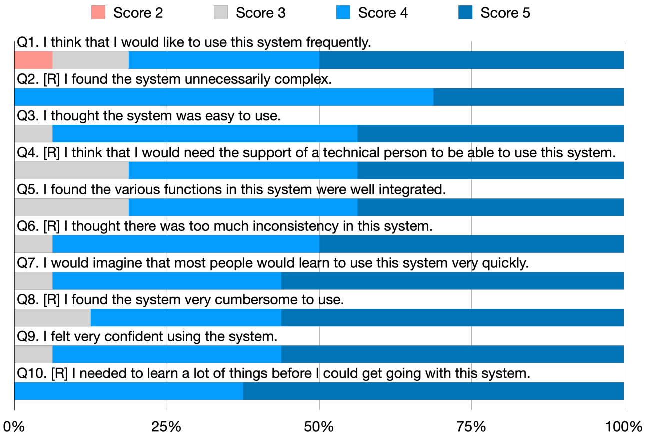

Figure 13: Usability test results using a 5-scale SUS questionnaire. The horizontal axis is the percentage of responses in each category. Note that the scores of negatively worded statements (Q2,4,6,8,10) are reversed for better visualization.

8.3 Qualitative Results

8.3.1 Questionnaire Results. The SUS results suggest generally positive feedback on usability from participants with an overall score of 84.8±6.6, which means it is highly usable and acceptable by users according to Bangor et al.'s empirical evaluation [3]. In general, participants expressed confidence in their ability to effectively use the system and rated it as highly easy to use and easy to learn. The details of each SUS question can be found in Figure 13.

8.3.2 Interview feedback. We further transcribed the interviews and extracted quotes that were related to user experience and opinions about LipLearner. All participants were the first time using a silent speech interface. For the overall usability, 13 out of 16 participants explicitly mentioned that they would like to use LipLearner in the future: *"Now I can use my smart assistant anywhere* "[P2].

Participants were also impressed by the accuracy of the model and the rapid learning process. *"It learns so efficiently, [LipLearner] almost can read all my commands by only listening to me once"*[P9], *"It's amazing that the model can be trained in the blink of an eye."*[P15]

All participants have noticed the improvement in recognition performance, 11 of them found it enjoyable to see the model performs better and better. *"I enjoyed teaching the AI model, it brings me closer to my smart assistant, making it no longer feel like a cold algorithm."*[P7] When asked further how many times they were willing to teach the model, most answers were around 3-5 times. P14 even expressed that *"I am willing to provide more samples for each command since I will gradually enrich my command set instead of immediately registering 30 commands as we did in the user study."*

Some participants further provided suggestions on how we could improve the prototype. Regarding the user interface and interaction, P8 believed that *"The camera view was distracting. I don't think it should necessarily be displayed to users."* and P13 mentioned *"I would be happy if the confirmation process could also be done using silent speech."*

While most of the participants were satisfied with using LipLearner in the on-demand learning mode, P6, P7, and P16 all mentioned about consequences of command execution with misrecognition. *"The commands have different importance and priority. It is better to confirm before the important commands, otherwise, something misrecognized as 'call the police' may lead to a bad consequence."* [P16]

To conclude, subjective feedback indicated that our system was easy to use and easy to learn, and has provided essential functionalities that allow users to customize their silent speech input experience in real-time.

9 DISCUSSION

9.1 Lipreading Beyond Speech

LipLearner benefits from the efficient visual speech representations learned via a contrastive learning strategy. Through our usability studies, we have demonstrated that our method enables to recognize silent speech with a small amount of training data, and its excellent performance can generalize to different phrasing, languages, and even non-verbal lip gestures such as making facial expressions. This ability push forward lipreading beyond speech. One potential application is using lipreading for user authentication in complement to face recognition, preventing spoofing attacks and password leakage. The user can define a secret "lip password" by combining several lip gestures, and our few-shot learning technique allows the user to change the password with little effort. Such nonverbal password is difficult to be inferred or remembered by others, therefore being suitable for high-security authentications, e.g., unlocking the device or making a payment. Furthermore, although our model is purposed to learn semantic information, we expect the semi-supervised visual speech representations also have the potential to inform user-dependent patterns stemming from subtle lip movements, making it more unlikely to be reproduced by others. Investigating the difference among individuals can help further understand the feasibility of lipreading-based speaker verification.

9.2 Towards Wearable Lipreading

This research is based on mobile interactions because of the prevalence of smartphones. However, we believe lipreading technologies can facilitate communication between humans and computers in a diversity of scenarios. The recent boom in head-mounted displays (HMD) based VR/AR applications calls for natural input methods with high mobility. Lipreading is a promising approach for its expressiveness and low learning cost, and it can be easily implemented by embedding a lip-observing camera in the headset. However, lipreading at such a close distance is not trivial because capturing the mouth usually requires a fish eye camera, whose distortion effects can pose challenges for recognition. Yet, placing the camera in the front of face is obtrusive. Our method in contrast has shown a consistently good performance recognizing from different points of view. To explore the feasibility of applying LipLearner in wearable scenarios, we did a preliminary study by mounting a USB camera on a 3D-printed headset (Figure 14) that captures the user's profile face. We collect a dataset from one of the authors with the same command set used in Section 4, making up a dataset of 25 commands × 4 repetitions = 100 samples. We evaluated the system's performance by running an offline test on a PC, and the 1-shot, 2-shot, and 3-shot accuracies are 0.7941, 0.9387, 1.0 (averaged over 100 random seeds). These early results indicate that our model can achieve good performance even recognizing profile faces. Furthermore, the visual KWS technique can free users' hands and better make them immersed in the virtual worlds. This preliminary

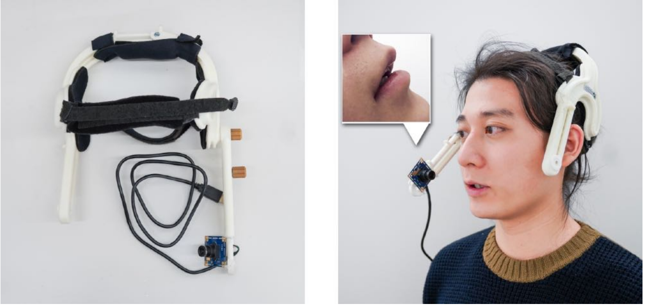

Figure 14: The device used for the preliminary test on wearable lipreading using our few-shot customization framework.

study demonstrates that our few-shot lipreading framework holds the promise of extending the dimensions of VR/AR interactions.

9.3 Human-in-the-loop Incremental Learning

LipLearner sheds new light on human-in-the-loop interactions by focusing on offering a natural and easy way to involve users. Instead of immediately requiring enormous data to pursue high accuracy, we introduce a one-shot command registration technique *Voice2Lip* to allow rapid initialization. LipLearner proactively solicits new samples from the user when the data is insufficient, and learns in an on-demand mode when high accuracy is achieved. Feedback from the user study suggested that participants enjoyed this human-AI interaction, and they were willing to help with improving the AI system during use. We envision that in the future, the design space of how to engage users to provide knowledge for learnable AI systems, such as minimizing the disruptions, will be an important topic in HCI.

10 LIMITATIONS AND FUTURE WORK

While LipLearner demonstrates favorable usability, there are several key limitations that will need to be overcome in the future.

First, there is still room to lessen the physical and cognitive labor of active learning. Several participants mentioned despite the fact that they enjoyed helping improve the model in the active learning mode, it would be better to be able to validate or correct the predictions also using silent speech (e.g., saying *"Yes"* or *"Cancel"*) instead of tapping buttons. Although this feedback also indicates that silent speech is preferred for its low effort in mobile interactions, the interaction design should be optimized to better involve the user in the human-in-the-loop flow.

Second, although our user study observations revealed LipLearner's tolerance for minor changes of expressions, this may make it more difficult to distinguish very similar commands. For example, we find that one of the common misrecognition is between *"Turn on the light"* and *"Turn on the flashlight"*. The problem can be alleviated by proactively soliciting more samples for low-accuracy commands or asking the user to rephrase.

Undoubtedly, few-shot learning has enhanced silent speech by extending the vocabulary capacity and minimizing the user burden in command registration. However, due to the lack of context, the level of abstraction of lip commands is still relatively low. For example, two separate commands need to be registered to set the alarm for 8 AM and 9 AM. We envision that the expressiveness and abstraction level of LipLearner can be further boosted by training zero-shot lipreading models jointly with language models such as GPT3 [5] or T5 [51]. In zero-shot lipreading, the user only has to prepare a bunch of command candidates they would like to use, and the model can recognize completely unseen commands by matching lipreading embeddings with text embeddings.

11 CONCLUSION

This paper presents LipLearner, a lipreading-based silent speech interface that enables in-situ command customization on mobile devices. We leverage contrastive learning to build a model to learn efficient visual speech representations from public datasets, providing in-situ fine-tuning for unseen users and words using few-shot learning. For a preliminary test, we collected a dataset covering various mobile interaction scenarios to evaluate the model's performance and robustness against lighting conditions, user posture, and hold gestures. The result showed that our method could provide consistent performance in different settings, outperforming conventional supervised methods. To investigate usability, we developed a prototype of LipLearner on iOS by integrating the few-shot customization framework with an online incremental learning scheme, involving the user in the learning process to improve the model on their demand. We further minimize the labor of command registration and incorporate speech recognition to automatically learn new commands from voice input. Through a user study, we demonstrated that LipLearner also has excellent performance with various commands defined by participants in different languages. The subjective feedback suggested that LipLearner is easy to use and easy to learn, and most participants enjoyed the human-AI integrated interaction. To conclude, our system democratizes silent speech by offering quick-start on-device lipreading, and it unleashes users' creativity with customizable commands. We hope our work can bring the vision of human-centered AI closer to reality, spotlighting the importance of intuitive and personalized interaction experiences.

ACKNOWLEDGMENTS

This work was supported by JST Moonshot R&D Grant Number JPMJMS2012, JST CREST Grant Number JPMJCR17A3, The University of Tokyo Human Augmentation Research Initiative, and a collaborative research fund between Mercari Inc. R4D and RIISE. We would further like to thank the anonymous reviewers for their constructive feedback and the participants of the user study.

REFERENCES

[1] Miguel Angrick, Christian Herff, Emily Mugler, Matthew C Tate, Marc W Slutzky, Dean J Krusienski, and Tanja Schultz. 2019. Speech synthesis from ECoG using densely connected 3D convolutional neural networks. *Journal of neural engineering* 16, 3 (2019), 036019.

[2] Relja Arandjelovic and Andrew Zisserman. 2018. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*. 435–451.

[3] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[4] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[6] Tuochao Chen, Benjamin Steeper, Kinan Alsheikh, Songyun Tao, François Guimbretière, and Cheng Zhang. 2020. C-Face: Continuously Reconstructing Facial Expressions by Deep Learning Contours of the Face with Ear-Mounted Miniature Cameras. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 112–125. https://doi.org/10.1145/3379337.3415879

[7] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A Closer Look at Few-shot Classification. *CoRR* abs/1904.04232 (2019). arXiv:1904.04232 http://arxiv.org/abs/1904.04232

[8] Xinlei Chen, Saining Xie, and Kaiming He. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9640–9649.

[9] Joon Son Chung and Andrew Zisserman. 2016. Lip reading in the wild. In *Asian conference on computer vision*. Springer, 87–103.

[10] Statista Research Department. 2022. *Main devices used with voice assistants in the U.S. 2021, by brand*. Retrieved April 28, 2022 from https://www.statista.com/statistics/1274398/voice-assistant-use-by-device-united-states/

[11] Tony Ezzat and Tomaso Poggio. 1998. Miketalk: A talking facial display based on morphing visemes. In *Proceedings Computer Animation'98 (Cat. No. 98EX169)*. IEEE, 96–102.

[12] Tony Ezzat and Tomaso Poggio. 2000. Visual speech synthesis by morphing visemes. *International Journal of Computer Vision* 38, 1 (2000), 45–57.

[13] Michael J Fagan, Stephen R Ell, James M Gilbert, E Sarrazin, and Peter M Chapman. 2008. Development of a (silent) speech recognition system for patients following laryngectomy. *Medical engineering & physics* 30, 4 (2008), 419–425.

[14] Dalu Feng, Shuang Yang, Shiguang Shan, and Xilin Chen. 2020. Learn an effective lip reading model without pains. *arXiv preprint arXiv:2011.07557* (2020).

[15] Masaaki Fukumoto. 2018. Silentvoice: Unnoticeable voice input by ingressive speech. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 237–246.

[16] Ruohan Gao and Kristen Grauman. 2021. Visualvoice: Audio-visual speech separation with cross-modal consistency. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 15490–15500.

[17] Jose A Gonzalez, Lam A Cheah, James M Gilbert, Jie Bai, Stephen R Ell, Phil D Green, and Roger K Moore. 2016. A silent speech system based on permanent magnet articulography and direct synthesis. *Computer Speech & Language* 39 (2016), 67–87.

[18] Frank H Guenther, Jonathan S Brumberg, E Joseph Wright, Alfonso Nieto-Castanon, Jason A Tourville, Mikhail Panko, Robert Law, Steven A Siebert, Jess L Bartels, Dinal S Andreasen, et al. 2009. A wireless brain-machine interface for real-time speech synthesis. *PloS one* 4, 12 (2009), e8218.

[19] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2021. Contrastive Predictive Coding for Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 65 (jun 2021), 26 pages. https://doi.org/10.1145/3463506

[20] Christian Herff, Dominic Heger, Adriana De Pesters, Dominic Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz. 2015. Brain-to-text: decoding spoken phrases from phone representations in the brain. *Frontiers in neuroscience* 9 (2015), 217.

[21] Yiyang Huang, Xuefeng Liang, and Chaowei Fang. 2021. CALLip: Lipreading Using Contrastive and Attribute Learning. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 2492–2500. https://doi.org/10.1145/3474085.3475420

[22] Thomas Hueber, Elie-Laurent Benaroya, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. 2010. Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips. *Speech Communication* 52, 4 (2010), 288–300.

[23] Apple Inc. 2022. *Core ML | Apple Developer Documentation*. Retrieved Feb. 9, 2023 from https://developer.apple.com/documentation/coreml

[24] Apple Inc. 2022. *Create ML | Apple Developer Documentation*. Retrieved Feb. 9, 2023 from https://developer.apple.com/documentation/createml

[25] Apple Inc. 2022. *SFSpeechRecognizer | Apple Developer Documentation*. Retrieved Feb. 9, 2023 from https://developer.apple.com/documentation/speech/sfspeechrecognizer

[26] Apple Inc. 2022. *Shortcuts User Guide - Apple Support*. Retrieved Feb. 9, 2023 from https://support.apple.com/guide/shortcuts/welcome/ios

[27] Apple Inc. 2022. *Vision | Apple Developer Documentation*. Retrieved Feb. 9, 2023 from https://developer.apple.com/documentation/vision

[28] Apple Inc. 2022. *What can I ask Siri? - Official Apple Support*. Retrieved Feb. 9, 2023 from https://support.apple.com/siri

[29] Dhruv Jain, Khoa Huynh Anh Nguyen, Steven M. Goodman, Rachel Grossman-Kahn, Hung Ngo, Aditya Kusupati, Ruofei Du, Alex Olwal, Leah Findlater, and Jon E. Froehlich. 2022. ProtoSound: A Personalized and Scalable Sound Recognition System for Deaf and Hard-of-Hearing Users. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 305, 16 pages. https://doi.org/10.1145/3491102.3502020

[30] Yan Ji, Licheng Liu, Hongcui Wang, Zhilei Liu, Zhibin Niu, and Bruce Denby. 2018. Updating the silent speech challenge benchmark with deep learning. *Speech Communication* 98 (2018), 42–50.

[31] Arnav Kapur, Shreyas Kapur, and Pattie Maes. 2018. Alterego: A personalized wearable silent speech interface. In *23rd International conference on intelligent user interfaces*. 43–53.

[32] Vahid Kazemi and Josephine Sullivan. 2014. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *CVPR*.

[33] Naoki Kimura, Tan Gemicioglu, Jonathan Womack, Richard Li, Yuhui Zhao, Abdelkareem Bedri, Zixiong Su, Alex Olwal, Jun Rekimoto, and Thad Starner. 2022. SilentSpeller: Towards mobile, hands-free, silent speech text entry using electropalatography. In *CHI Conference on Human Factors in Computing Systems*. 1–19.

[34] Naoki Kimura, Kentaro Hayashi, and Jun Rekimoto. 2020. TieLent: A Casual Neck-Mounted Mouth Capturing Device for Silent Speech Interaction. In *Proceedings of the International Conference on Advanced Visual Interfaces*. 1–8.

[35] 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*. 1–11.

[36] Naoki Kimura, Zixiong Su, and Takaaki Saeki. 2020. End-to-End Deep Learning Speech Recognition Model for Silent Speech Challenge.. In *INTERSPEECH*. 1025–1026.

[37] Naoki Kimura, Zixiong Su, Takaaki Saeki, and Jun Rekimoto. 2022. SSR7000: A Synchronized Corpus of Ultrasound Tongue Imaging for End-to-End Silent Speech Recognition. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 6866–6873.

[38] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.

[39] Soonkyu Lee and DongSuk Yook. 2002. Audio-to-visual conversion using hidden markov models. In *Pacific Rim International Conference on Artificial Intelligence*. Springer, 563–570.

[40] Richard Li, Jason Wu, and Thad Starner. 2019. Tongueboard: An oral interface for subtle input. In *Proceedings of the 10th Augmented Human International Conference 2019*. 1–9.

[41] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172* (2019).

[42] Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2020. Lipreading using temporal convolutional networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6319–6323.

[43] Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2018. Development of sEMG sensors and algorithms for silent speech recognition. *Journal of neural engineering* 15, 4 (2018), 046031.

[44] Carolina Milanesi. 2016. Voice Assistant Anyone? Yes please, but not in public. *Creative Strategies* (2016).

[45] Laxmi Pandey and Ahmed Sabbir Arif. 2021. LipType: A Silent Speech Recognizer Augmented with an Independent Repair Model. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–19.

[46] Stavros Petridis, Themos Stafylakis, Pingehuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. 2018. End-to-end audiovisual speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 6548–6552.

[47] Anne Porbadnigk, Marek Wester, Jan-P Calliess, and Tanja Schultz. 2009. EEG-based speech recognition.

[48] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. 2020. Learning individual speaking styles for accurate lip to speech synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13796–13805.

[49] Qinwan Rabbani, Griffin Milsap, and Nathan E Crone. 2019. The potential for a speech brain–computer interface using chronic electrocorticography. *Neurotherapeutics* 16, 1 (2019), 144–165.

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

[51] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[52] Takeshi Saitoh and Michiko Kubokawa. 2019. LiP25w: Word-level Lip Reading Web Application for Smart Device. *The 15th International Conference on Auditory-Visual Speech Processing* (2019).

[53] 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 (1987), 26–35.

[54] Alex Sciuto, Arnita Saini, Jodi Forlizzi, and Jason I Hong. 2018. " Hey Alexa, What's Up?" A Mixed-Methods Studies of In-Home Conversational Agent Usage. In *Proceedings of the 2018 designing interactive systems conference*. 857–868.

[55] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. 2018. Learning to localize sound source in visual scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4358–4366.

[56] Changchong Sheng, Matti Pietikäinen, Qi Tian, and Li Liu. 2021. Cross-Modal Self-Supervised Learning for Lip Reading: When Contrastive Learning Meets Adversarial Training. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 2456–2464. https://doi.org/10.1145/3474085.3475415

[57] Zixiong Su, Xinlei Zhang, Naoki Kimura, and Jun Rekimoto. 2021. Gaze+ Lip: Rapid, Precise and Expressive Interactions Combining Gaze Input and Silent Speech Commands for Hands-free Smart TV Control. In *ACM Symposium on Eye Tracking Research and Applications*. 1–6.

[58] 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* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 581–593. https://doi.org/10.1145/3242587.3242599

[59] Tomoki Toda, Mikihiro Nakagiri, and Kiyohiro Shikano. 2012. Statistical voice conversion techniques for body-conducted unvoiced speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 9 (2012), 2505–2517.

[60] Tomoki Toda, Keigo Nakamura, Hidehiko Sekimoto, and Kiyohiro Shikano. 2009. Voice conversion for various types of body transmitted speech. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 3601–3604.

[61] Tomoki Toda and Kiyohiro Shikano. 2005. NAM-to-speech conversion with Gaussian mixture models. (2005).

[62] Carnegie Mellon University. 2011. *The CMU Pronouncing Dictionary*. Retrieved Feb. 9, 2023 from http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[63] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* (2018), arXiv–1807.

[64] Michiel Visser, Mannes Poel, and Anton Nijholt. 1999. Classifying visemes for automatic lipreading. In *International Workshop on Text, Speech and Dialogue*. Springer, 349–352.

[65] Michael Wand and Tanja Schultz. 2011. Session-independent EMG-based Speech Recognition.. In *Biosignals*. 295–300.

[66] Disong Wang, Shan Yang, Dan Su, Xunying Liu, Dong Yu, and Helen Meng. 2022. VCVTS: Multi-Speaker Video-to-Speech Synthesis Via Cross-Modal Knowledge Transfer from Voice Conversion. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7252–7256. https://doi.org/10.1109/ICASSP43922.2022.9747427

[67] Jason Wu, Chris Harrison, Jeffrey P. Bigham, and Gierad Laput. 2020. Automated Class Discovery and One-Shot Interactions for Acoustic Activity Recognition. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376875

[68] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. 2022. Enabling Hand Gesture Customization on Wrist-Worn Devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 496, 19 pages. https://doi.org/10.1145/3491102.3501904

[69] Ruidong Zhang, Mingyang Chen, Benjamin Steeper, Yaxuan Li, Zihan Yan, Yizhuo Chen, Songyun Tao, Tuochao Chen, Hyunchul Lim, and Cheng Zhang. 2022. SpeeChin: A Smart Necklace for Silent Speech Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 192 (dec 2022), 23 pages. https://doi.org/10.1145/3494987

A USER STUDY COMMAND SET

Table 3: 10 Pre-defined Commands.

Get directions to gas station	Take a photo
Open Twitter	Turn on focus mode
Play some music	Turn on the flashlight
Send an email	What’s the weather today
Set an alarm for 8 am	Show today’s schedule

Table 4: 20 custom commands. B1 to B10 are the user-described commands with given scenarios. C1 to C10 are the user-created commands.

Participant	**P1**	**P2**	**P3**	**P4**
Languages	**French, Japanese, English**	**Chinese, English**	**Chinese, Japanese, English**	**Spanish, English**
Keyword	**Hello, Mirai**	**Hello, Baymax**	**Hello, Mugi**	**Hello, David**
B1	キーはどこ	我钥匙呢	Where is my key	Donde estan mis llaves
B2	Apelle maman	打给妈妈	Call my mom	Llama a mama
B3	Ouvre les rideaux	拉开窗帘	Open the curtain	Abre las cortinas
B4	Reserve un ticket d’avion pour le Japon	去东京的机票多少钱	Buy the ticket to tokyo	Compra pasaje de vuelo
B5	電気消して	关灯	Turn off the light	Prende las luces
B6	Order food	最近的汉堡店在哪	Reserve a restaurant	Llama a ubereats
B7	最近のニュースはどう	今天有什么新闻	Open Japan today	Dime las noticias de hoy
B8	6 minutes timer	倒计时6分钟	Count six minutes	Pon cronometro de seis minutos
B9	Where is my car	我车呢	Find my car	Encuentra mi auto
B10	Monte le chauffage	关空调	Turn off the air conditioner	Prende la calefaccion
C1	Read this	带我去最近的超市	分かりません	Reiniciate
C2	一番近いゲーセン	现在几点	また来週	Abre traductor de google
C3	Comment ca va?	明天八点叫我起床	だめだね	Call father
C4	I am having a lot of fun	东京大学怎么样	Do some research	Edit photo
C5	Quand ouvre le cinema	播放《武林外传》	关闭声音	Dim screen to minimun
C6	What are you doing in my swamp	打开亚马逊搜索	给我点钱	Delete photo
C7	Name all the pokemons	帮我打的去机场	帮我做饭	Lock my screen
C8	Chante une chanson	提醒我下周三上午九点有会议	打电话给爸爸	Open clash of clans
C9	Create a macro for my lilly heals	打开日语翻译器	打开电视机	Download file
C10	When is the next HatsuneMiku concert	豆沙馅怎么做	放点轻松音乐	Share video

Participant	**P5**	**P6**	**P7**	**P8**
Language Used	**Chinese, English**	**English**	**English**	**Japanese, English**
Keyword	**Hello, Mamun**	**Hello, Mr. Ha**	**Hello, friend**	**Hello, David**
B1	Where’s my key	Can you find my key	Where’s my key	鍵をさがして
B2	Call mom	Call Mom	Call my mom	お母さんをよんで
B3	Open curtains	Open the curtain	Open curtain	カーテンをあけて
B4	预定飞机票	I want to book a ticket to Tokyo	Find me a ticket to tokyo	航空券を予約して
B5	关灯	Turn to sleep mode	Turn off light	電気を消して
B6	预定餐厅	Find the nearest restaurant	Find me some restaurants	レストランを予約して
B7	浏览新闻	What’s the news today	Read me some news	ニュースを読んで
B8	设定6分钟的计时器	Set a timer for 6 minutes	Set a 6 minutes timer	６分間のタイマーをセットして
B9	寻找我的车	Find my car	Where did I park my car	車を探して
B10	打开暖风	Turn on the air conditioner	Turn off the aircon	暖房をつけて
C1	整理相册	Tell me how to say ”sorry” in Japanese	How to go to my university	一ドルは何円
C2	打开投影仪	Am I smart?	Text my mom	今日の予定を教えて
C3	设定闹钟	Delete Wechat	Clean my house	今日の天気を教えて
C4	今天的股票价格	Buy some pork in Amazon	Play happy eliminating	フェイスブックを開けて
C5	你的工作是什么	Update my calendar	Show my calendar	自宅までの距離は？
C6	播放音乐	Where is the bus stop	Where’s the nearest hospital	クラシック音楽をかけて
C7	讲个笑话吧	Call uber to my hometown	USD to Japanese yen	ドアを開けて
C8	最近的音乐会有哪些	Install Google	What’s gravity	大阪までの経路を教えて
C9	今天的天气如何	Buy 1 million stocks	Open camera	明日の７時に締め切りをリマインドして
C10	导航回家	Call my lover	Turn off volume	卵焼きの作り方を教えて

Participant	**P9**	**P10**	**P11**	**P12**
Language used	**Chinese, English**	**Chinese,English**	**Chinese, English**	**Chinese(Cantonese, Hakka), Malay, Japanese, English**
Keyword	**Hello, Tom**	**Hello, Alexa**	**Hello, Jessica**	**Hello, David**
B1	我的钥匙在哪	Find my key	Where is my key	我的钥匙呔
B2	打电话给妈妈	Make a phone call to Mom	Make a call	Call mami
B3	打开窗帘	Open the curtain	Open the curtain	Wake me up at 9
B4	帮我买去东京的机票	Book a flight ticket	Book a flight ticket	Tolong beli tiket ke jepun
B5	关灯	Turn on the light	Turn off light	Nak tidur ni
B6	查找附近的餐厅	Make a reservation of restaurants	Find a restaurant	いい感じのレストラン探して
B7	打开新闻	What’s today’s news	Show news	Pull up today’s news
B8	倒计时六分钟	Set a timer for six minutes	Set the alarm clock at 6	Set the timer to 6
B9	我的车在哪	where is my car	Find my car	車どこ
B10	提高空调的温度	Turn on the air conditioner	Turn up the temperature	太冷了
C1	提高耳机音量	放几首歌听	Tell a joke	I want to be rich
C2	帮我回复妈妈的微信，好的	今天天气好吗	Turn off camera and microphone	I will be back
C3	早上8点开始洗衣服	设定一个明早九点的闹钟	Clean the trash bin	Esok jangan lupa BBQ ye
C4	打扫房间	今天天气怎么样	How old are you?	きみ、かわいいね
C5	今天天气怎么样	怎样去学校	Navigate me to the conference center	あらあら
C6	明天9点叫我起床	明天会下雨吗	Calculate twenty three hundred divided by six	ラーメンは煮干しでしょ
C7	我的手机在哪	打电话给小李	推荐一些新书	CHI論文通して
C8	提醒我明天交作业	附近有什么餐厅	叫个外卖	可以点菜了吗
C9	每小时帮我倒水	放一首周杰伦	未来一周的天气如何	唔中意日本
C10	附近的商场有哪些	帮我发条短信	新建文件夹	这个たこ焼き不错

Participant	**P13**	**P14**	**P15**	**P16**
Language used	**Japanese, English**	**Japanese, English**	**Chinese, Japanese, English**	**English**
Keyword	**Hello, Alexa**	**Hello, Alexa**	**Hello, Oliver**	**Hello, Thomas**
B1	Where is my key?	I lost my key	钥匙去哪了	Looking for my key
B2	Hello mom	Call my mom	给妈妈打电话	Call mom
B3	Good morning	カーテンを開けて	打开窗帘	Draw the curtains
B4	一番安い飛行機はどれ？	I want to go to tokyo	查询一下去东京的机票	Check for tickets to tokyo
B5	電気を消して	Turn off the light	关灯	Turn off the lights
B6	一番近いレストランはどこ	お腹すいた	有没有什么推荐的餐厅	Find a restaurant for me
B7	ニュースを開いて	open news app	今天有什么新闻	What’s news today
B8	Set Timer	Set a timer	倒计时六分钟	Countdown 6 minutes
B9	where is my car	Find my car	我的车子在哪	Where did I park my car
B10	エアコンの温度を下げて	暖房つけて	调高空调温度	Warm up here
C1	Is Singularity already here?	お腹いっぱいです	家への経路を教えて	Check formula 1 schedule
C2	Make collage lunches better	家に帰りたい	明早7点叫我起床	USD to Japanese yen
C3	Write a book that sells well	喉が渇いた	さっきの写真をインスタグラムにあげて	Check youtube updates
C4	What is the raw material of these clothes	電源を消して	ラーメン食べたい	Monitoring my dog at home
C5	Grow houseplants	Find my laptop	麻婆豆腐怎么做	Next month’s bills
C6	味噌汁を作って	Misactivate	電気を消して	Play my daily mix
C7	ハンバーガーのハンバーグ抜きを頼んで	Tell me a joke	静かにして	Check out the nearby exhibitions
C8	遅刻の言い訳を考えて	Say hello	今日の終電は何時	Call the police
C9	日本メタバース協会ってなに	What time is it	最近のヒット曲を再生して	Todo list tomorrow
C10	私の博士論文を書いて	What’s your name	車借りて	Open Netflix