Watch Your Mouth: Silent Speech Recognition with Depth Sensing

Xue Wang
University of California, Los Angeles
Los Angeles, CA, USA
xw526@ucla.edu

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

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

Yang Zhang
University of California, Los Angeles
Los Angeles, CA, USA
yangzhang@ucla.edu

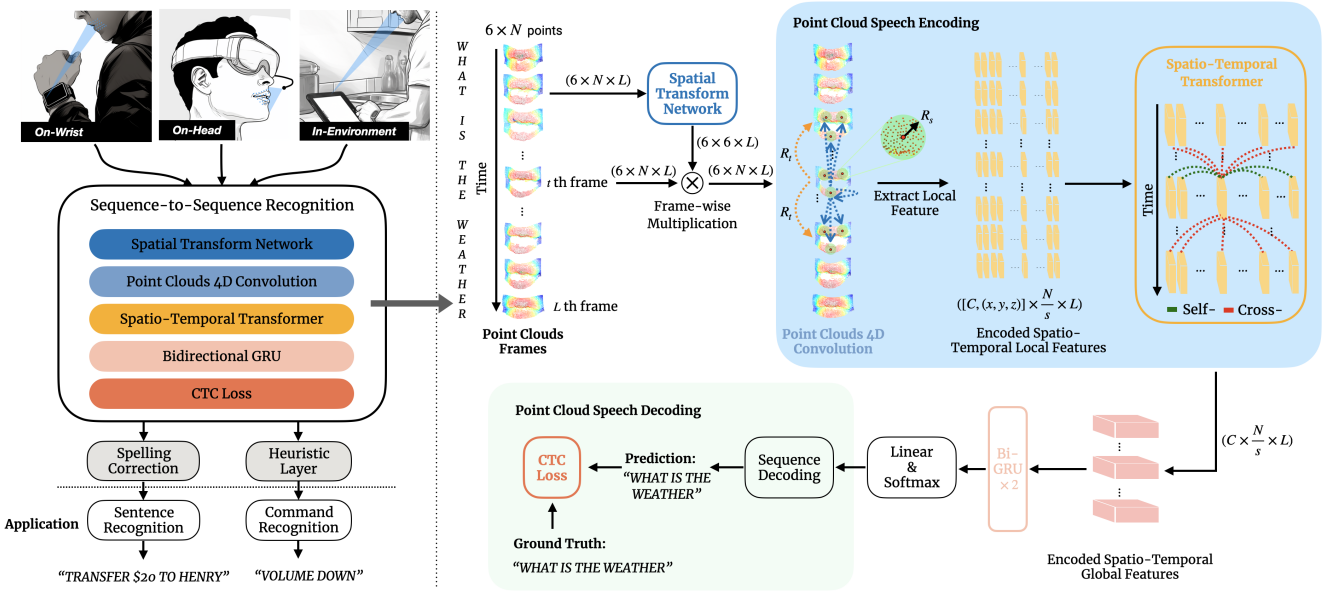

Figure 1: Left: *Watch Your Mouth* performs silent speech recognition through depth sensing at three different sensor locations: On-Wrist, On-Head, In-Environment; Right: The deep learning architecture of *Watch Your Mouth*, N represents the number of points in each point cloud frame, L signifies the length of the video, s denotes the sampling rate, R_s and R_t refer to the spatial radius and temporal kernel size, respectively. C is the dimension of the transformer layer in this pipeline.

ABSTRACT

Silent speech recognition is a promising technology that decodes human speech without requiring audio signals, enabling private human-computer interactions. In this paper, we propose *Watch Your Mouth*, a novel method that leverages depth sensing to enable accurate silent speech recognition. By leveraging depth information, our method provides unique resilience against environmental factors such as variations in lighting and device orientations, while further addressing privacy concerns by eliminating the need for sensitive RGB data. We started by building a deep-learning model that locates lips using depth data. We then designed a deep learning pipeline to efficiently learn from point clouds and translate lip movements into commands and sentences. We evaluated our technique and found it effective across diverse sensor locations: On-Head, On-Wrist, and In-Environment. *Watch Your Mouth* outperformed the state-of-the-art RGB-based method, demonstrating its potential as an accurate and reliable input technique.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05.
https://doi.org/10.1145/3613904.3642092

CCS CONCEPTS

• **Human-centered computing** → **Mobile devices**; **Interactive systems and tools**.

KEYWORDS

Silent Speech Recognition, Visual Speech Recognition, Lip Reading, Depth Sensing, Deep Learning, Input Techniques

ACM Reference Format:

Xue Wang, Zixiong Su, Jun Rekimoto, and Yang Zhang. 2024. Watch Your Mouth: Silent Speech Recognition with Depth Sensing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3613904.3642092

1 INTRODUCTION

Silent speech is a promising interaction modality for conveying user intents to a broad spectrum of computing devices, for its intuitiveness, efficacy, and privacy-preserving nature. Empowering these devices to decode silent speech effectively transforms them into optical microphones, enabling users to harness the intuitiveness and efficacy of speech without compromising privacy. Despite these merits, the integration of silent speech recognition into contemporary computing devices remains an open-ended research challenge. With silent speech recognition's unique advantage in privacy, we foresee a future where it could enhance smartwatches, VR/AR glasses, and environmentally deployed IoT devices with speech-based interactions, the capabilities of which have been significantly extended by the recent advances in Large Language Models (LLM). This elimination of tangible interfaces is inviting, and we believe the key lies in the improvements of silent speech recognition to have greater robustness against environmental variances.

Prior research on silent speech has employed a diverse set of sensors, most notably RGB cameras [4, 42, 51, 54], and ultrasound imaging [24, 25]. In this work, we first identify depth as a new unique sensory information of silent speech. Specifically, we utilized depth sensing to capture high-fidelity depth data in the form of point clouds to reconstruct user speech, at both world and sentence levels. Our sensing principle relies on the fact that human faces have distinctive shape changes resulting from movements of lips, tongue, teeth, and jaw during speech, which manifests as depth data that could be easily and cheaply acquired by depth sensing.

Depth sensing distinguishes itself from other vision sensors by its insensitivity to fluctuations in ambient lighting conditions, unlike RGB cameras which tend to be more susceptible to such variations. Moreover, depth sensing exhibits a consistent recognition accuracy across various skin tones [39], effectively broadening the technique's appeal to a diverse user demographic. Furthermore, computing devices with different instrumentation locations such as on-wrist (i.e., smartwatches) and on-head (i.e., VR/AR glasses) and in-environment (i.e., IoT devices) receive vastly different sensory information which poses generalizability challenges to conventional lipreading – models that are trained at certain device location might not generalize well to other locations unless new data is collected for calibration. The adaptability and robustness of depth sensing thanks to the perspective-invariant nature of depth make it more promising for addressing this generalizability challenge.

To generalize the utilization of depth sensing across various computing modalities for silent speech recognition, we transformed the depth image into point clouds. This conversion enhances adaptability to accommodate different angles and distances between users' lips and devices. Additionally, we calculated the point normals, which represent the local geometric property determined by its surrounding points [45]. These normals were then concatenated with the corresponding points to serve as input to our deep learning pipeline – PointVSR, which includes signal alignment, point cloud feature extraction, and sequence decoding using the connectionist temporal classification loss (CTC) with 26 English characters, 1 space character, and 1 blank as tokens. We proved the superiority of our PointVSR over existing silent speech recognition using RGB videos, with a Character Error Rate (CER) and a Word Error Rate (WER) decrease of 3.82% and 4.96% respectively in within-user validation and 5% and 4.57% in cross-user validation.

We have conducted in-lab user studies and included participants with diverse physiological features and native languages, imitating three real-world device locations: On-Wrist, On-Head, and In-Environment. We also performed within-user and cross-user evaluations, the results of which indicated that our system can recognize sets of 30 distinct commands, achieving an accuracy of 91.33% (within-user), 74.88% (cross-user), with a standard deviation of 1.44% (within-user), 13.47% (cross-user). Furthermore, we explored the feasibility of recognizing sentences, achieving WER of 8.06% (within-user), 29.14% (cross-user), along with CER of 4.13% (within-user), 18.28% (cross-user). We deep-dived into results and derived further insights into sources of errors and comparisons with a status-quo technique using RGB videos as inputs. These results signify that our approach surpasses previously attainable capabilities and holds significant promise for the future of silent speech recognition technology.

Below we list our key contributions:

- identified depth sensing as the new advantageous information source for silent speech recognition.
- realized a uniform recognition pipeline using depth information for word and sentence recognition.
- conducted validations and evaluations at three sensor locations to prove feasibility and superiority.

2 RELATED WORK

2.1 Depth Sensing in HCI

Depth sensors have improved drastically in precision and size, turning them into a popular sensing solution for a wide array of interactive systems. There has been a wide array of interactive systems using depth sensing in HCI applications. These prior works are relevant to our research regarding the fundamental sensing technique. Researchers have instrumented depth sensors on users as wearable devices. For instance, with 1D depth sensor arrays, ThumbTrack [55] and LumiWatch [61] track spatial relationships between a user's finger and the rest of the body to enhance wearable interactivity. 2D depth sensors (aka depth cameras) allow conventional interfaces to be rendered at non-conventional locations. For example, Skinput [18] and OmniTouch [17] use shoulder-worn depth cameras in concert with projectors to enable touchscreen-alike experience on a user's skin and everyday surfaces.

It is also possible to have depth sensors affixed to environments. For example, Worldkit [62] allows users to create ad-hoc interactors on everyday surfaces once they are recognized and tracked by a depth camera. LightSpace [59] and RoomAlive [19] combine multiple depth cameras and projectors to enable room-scale interactive experiences. Multiple users could be supported in the shared physical space (e.g., [5]). Systems that feature depth sensing have also been developed for special applications such as facilitating remote instructions of physical tasks [56], recognizing hand gestures for natural input [10, 47], and localizing optical tags [37]. Finally, the recent investments in Metaverse have given birth to extended reality devices that rely on depth sensing to correlate the physical environment with digital content. The extended reality could be used for entertainment [11], as well as improving accessibility of the physical environments for low-vision users [71, 72].

2.2 Silent Speech Recognition / Lip Reading

The concept of lip reading was first proposed by Sumby and Pollack [53] in the 1950s, who suggested that visual cues from the movements of speakers' mouths could be used to aid in speech recognition. This led to the development of the first automated speech recognition system, which relied on geometric features such as mouth height, width, area, and perimeter to identify speech content. Recent advancements in deep learning and the availability of large-scale datasets have enabled the development of more sophisticated and accurate techniques for facial visual feature extraction and speech recognition. These methods employed multiple modalities, the most common one of which is RGB videos [4, 38, 42, 44, 50, 52, 64]. Infrared videos could also be utilized in a wearable form factor [68]. Other sensing modalities include sEMG [21], ultrasound [24, 25], RFID [58], capacitive [23, 30], and mmWave sensors [65], each of which has demonstrated its unique strengths and shortcomings in speech recognition.

In HCI research, SilentSpeller [23] uses a capacitive sensor array placed inside the user's mouth to detect tongue movements and generate features for recognizing spoken words, using a PCA-compressed feature in conjunction with HMMs. EchoSpeech [69] leverages speaker-microphone pairs to sense skin deformation using active acoustic sensing. This signal is then used to recognize silent speech such as commands and numbers. Similarly, HPSpeech [67] monitors patterns from reflected acoustic signals emitted by existing speakers on headphones to recognize multiple commands. SottoVoce [24] is another promising approach that uses ultrasound images of the tongue and vocal tract to synthesize human speech. Closer to our research's techniques is LipNet [4] which is a sentence-level lipreading approach that maps a variable-length sequence of video frames to text using deep neural networks. Also, Lip Learner [51] proposes an approach that leverages contrastive learning to learn efficient lipreading representations that enable few-shot command customization with minimal user effort.

Closest to our system is Lip-Interact [54], a silent speech recognition technique to enrich interactions on mobile devices. With RGB videos as input, this prior work achieved high recognition accuracies for both within- and cross-user settings with a command set with a size of 20. A deep learning structure composed of CNN and Bi-GRU takes 20 feature points of a user's lip as input. In comparison, *Watch Your Mouth* leverages depth as a different input signal for silent speech. The different sensor input unveils drastically different discoveries and requires novel signal preprocessing and deep learning structure to achieve a performance that not only proves feasibility but also compares favorably than prior work. Finally, we evaluated three device locations – On-Wrist, On-Head, and In-Environment to simulate smartwatches, XR headsets, and IoT devices beyond the scope of considerations in all prior work.

3 SENSING PRINCIPLE

In this section, we conducted data collection and employed viseme detection accuracy as an evaluation metric to validate the principle of depth sensing for silent speech recognition. Additionally, we investigated the information degradation as the region of interest surrounding a user's lip increased, to select the optimal size for the bounding box in lip segmentation – the first step in our silent speech recognition pipeline.

3.1 Source of Information

Lip movements are an essential component of speech and result in distinctive shape changes in the speaker's face which can be captured at high fidelity by depth sensing. In visual speech, lip movements are the most apparent feature and are controlled by the same articulatory organs that control audible speech, including lips, tongue, teeth, jaw, velum, larynx, and lungs, among which only lips, teeth, jaw, and tongue are visible for lipreading [9, 29]. This limited information poses a significant challenge for understanding and modeling lip movements during speech. On the other hand, lip movements can vary significantly between individuals and even within individuals, depending on factors such as habits and emotions. Therefore, modeling lip movements for speech recognition requires a deep understanding of the nuances of individual speech patterns, which we aim to achieve using deep learning techniques and high-fidelity training data.

Lip movements are often decomposed into visemes which are visual tokens that can be mapped directly to the 3D shape of a speaker's face and are considered as visual representations of phonemes. Prior work has created a dictionary to infer speech from a stream of viseme [27]. In the following validation study, we used a straightforward viseme classifier to validate our sensing principle – the depth data of a speaker's face contains enough information for speech recognition. Additionally, we used the classification accuracy as an indicator of signal fidelity to optimize the scope of region of interest (i.e., how large of a user's mouth should be captured by a depth sensor for speech recognition).

3.2 Validation Study

We first selected a phonetically balanced sentence list from the Harward Sentence List (see Auxiliary file) to be recorded for viseme detection. To ensure the reliability of our study, We recruited 12 participants (4 Females), and asked each participant to read the sentence list once with a typical conversational style. This process allowed us to collect data on different visemes in a natural setting. The depth frames as well as the speaking audio were collected using an iPhone 12 mini *TrueDepth* Camera and microphone, respectively. The iPhone was placed on a tabletop in from of seated participants

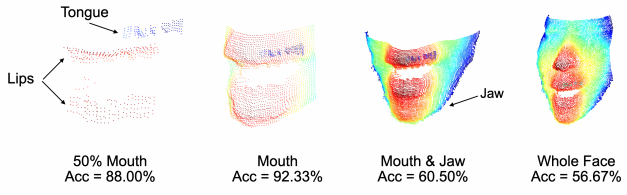

Figure 2: Illustration of point clouds on four facial ROIs and the viseme classification accuracies with them as inputs.

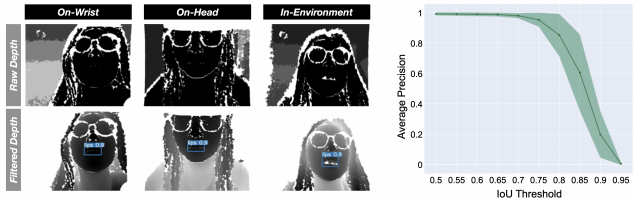

Figure 3: Left: The foreground (bottom) which contains a user's head is segmented from the raw depth image (top) and then normalized to optimize for lip detection. The detected bounding box is depicted in the images with a confidence value. We then use it to crop ROI from the raw depth image and convert it to a point cloud. Right: IoU Threshold versus average precision curve of lip detection.

in a quiet lab environment. After data from all participants was acquired, we first utilized the CMU dictionary [27] to translate a sentence into a sequence of phonemes. Subsequently, we employed a transformer-based phonetic aligner [73], to align the recognized phonemes with the recorded audio and locate the temporal boundaries of each phoneme in this sequence. This allowed us to label depth frames with recognized phonemes accordingly. Finally, we map phonemes into their corresponding visemes.

Previous studies on silent speech recognition using RGB images as input have typically focused on the mouth/lips region as the primary area of interest. However, as noted in [70], the extraoral regions of faces may also contribute to speech recognition. Therefore, in this section, we investigated the performance on viseme classification from four scopes of facial regions, cropped from raw depth images using MediaPipe: 1) the 50% of mouth area, 2) the mouth region alone, 3) the mouth with jaw area, and 4) the whole face (Fig. 2 bottom). By doing so, we aimed to find the optimal size of the bounding box around a user's lip to propagate depth data down to the next steps in the pipeline.

To convert depth data into a format that better supports flexible spatial manipulations, we transformed the cropped depth frames into point clouds using the intrinsic matrix gathered from iPhone. We employed the state-of-the-art point cloud classification model - PointNet [45, 46], as the classifier to perform the viseme classification. Fig. 2 visualizes the viseme classification results on the four facial regions. The high viseme detection accuracy indicates that depth sensing can accurately capture the visual cues of speech, which could validates our sensing principle. Our results also indicate that the *Mouth* region provides the most informative feature during speech with the highest accuracy of 92.33% in viseme classification. Consequently, we opted for a bounding box with an adaptive size to ensure the complete capture of a user's lips in the lip segmentation process. The dimensions of this bounding box, both width and height, are determined by the outermost points identified on the user's lips. Depth frames within this bounding box are cropped out and converted into point clouds for further processing which we document next.

4 IMPLEMENTATION

4.1 Depth Data Acquisition

We selected the *TrueDepth* camera featured on the iPhone 12 mini to acquire depth information related to the user's lip movements. The TrueDepth camera, primarily designed for Face ID and user identification on iPhones, provides real-time depth data streams and facilitates accurate computations of distances and orientations for individual pixels captured. To access the depth data, we developed a phone application built on the AVFoundation framework, which enabled us to efficiently capture and store the depth data required for our subsequent silent speech recognition steps. The depth sensing on the TrueDepth camera was configured to 480 by 640 pixels at 30 FPS, all of which are configurable maximums of stable depth sensing. Additionally, we collected RGB images with the same resolution and FPS as the ones of the depth images from the RGB camera (12MP, $f/2.2$ aperture) on the TrueDepth camera system for later comparisons between our depth-based approach and prior work using RGB-based approaches. More specifications of depth sensing and RGB sensing on the TrueDepth camera can be found on the product page[1]. We instrumented three iPhone 12 mini devices on three sensor prototypes to simulate computer devices that are deployed on-wrist, on-head, and in-environment (Fig. 4).

4.2 Lip Segmentation

We preserve signal-to-noise ratios (SNR) by eliminating irrelevant information from the input depth frames. Specifically, we perform lip segmentation to extract regions of interest (ROI). In conventional approaches that utilize RGB images as input, the ROI of lips is determined using pre-trained DNN-based face detectors, such as those available in Dlib [26] and MediaPipe [33]. However, to the best of our knowledge, no prior work has explored lip detection solely based on depth data. To locate lips within depth data, we employ transfer learning to fine-tune a pre-trained object detection model, YOLOv7 [57], enabling it to detect lips in depth images.

4.2.1 Foreground Segmentation and Data Conversion. To adapt our data for the YOLOv7 model, we first convert the depth map (i.e., original depth data yielded by the AVFoundation framework) into images. Our initial step involves filtering the depth map using a distance mask for background subtraction, which removes irrelevant background information that could distract the model. To establish the optimal upper threshold for the distance mask, we employed the MediaPipe face detector to identify faces in the RGB image and computed the distances from users in our dataset. We empirically

[1]TrueDepth camera specification: https://support.apple.com/kb/SP829?locale=en_US

determined a distance threshold of 0.5 meters for extracting the foreground, which includes a user's face (Fig. 3 left), before normalizing the image to a range of 0 to 255 (i.e., 8-bit integers). This thresholding and normalization process significantly enhances the resolution of the converted depth image, thereby improving the effectiveness of lip segmentation.

4.2.2 Lip Segmentation Model. We fine-tuned a pre-trained YOLO v7 model [60], pre-trained on traditional visual tasks such as objection detection. As one of the state-of-the-art face detection models, MediaPipe provides stable and accurate facial landmarks and was used as a ground-truth method. Specifically, to prepare an annotated dataset for transfer learning, we used the MediaPipe face detector to perform inference on the paired RGB images in our dataset. We converted predicted pseudo labels into rectangle bounding boxes around a user's lips and formatted them to YOLO-style labels as ground truth for the depth images. We evaluated the fine-tuned YOLOv7 model with ten-fold leave-one-user-out validation with data collected from the previous validation study. The Intersection over Union (IoU) between the predicted and the ground truth bounding box, is defined in the following equation:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{1}$$

IoU is an important performance indicator in object detection. We determine a successful recognition of the lip Region of Interest (ROI) by defining a high IoU threshold of 0.75. Under this condition, our lip segmentation model achieves an Average Precision (AP) of 95.31% (SD = 4.61%) across all participants. The AP-IoU threshold curve is illustrated in Fig. 3 right. Of note, although we used RGB data in the transfer learning pipeline, our trained model does not require this information during inference in speech recognition tasks.

With this model, we implement an automated lip ROI segmentation algorithm that crops out a user's lips using only depth data for further processing. Evidence from our validation study has shown that using the *Mouth* region is sufficient to achieve optimal performance. Therefore, we use an ROI that is slightly larger than the originally detected bounding box. By expanding it with a scaling factor of 1.1, to ensure a full coverage of a user's mouth. In cases where our lip detection model fails to locate the lips in a frame, our algorithm estimates the position and the size of the bounding box using linear interpolation on adjacent frames in the same utterance. The cropped ROI is finally normalized and converted to point clouds for speech recognition.

4.3 Sequence-to-Sequence Speech Recognition

In this work, we developed a sequence-to-sequence speech recognition model *PointVSR*, capable of directly processing point cloud videos as input data and output ***sentences*** or ***command words*** depending on the application layer. Specifically, our sentence recognition includes data processing layers we document in Section 4.3.1, 4.3.2, 4.3.3, and 4.3.4. For word detection, we use the same processing layers and map recognized word primitives to predefined commands in the command set with a heuristic layer, documented in Section 4.3.5.

4.3.1 Depth Image to Point Cloud Conversion. Human speech involves intricate 3D motion around lips and tongues. Depth images store depth data as pixel intensity, or in other words calculate 3D positions from a fixed-perspective 2D plane. This limitation prompted our exploration of alternative representations. Point clouds, on the other hand, are an inherent 3D data structure (X, Y, Z coordinates of points) to represent depth data. Furthermore, point clouds encode depth information that is invariant to their scale changes and rigid transformations, as well as permutations of points inside them [8]. Finally, point clouds could accommodate further encoding of depth data in per-point features such as normal vectors [3]. To convert depth images into point clouds, we employ a transformation that maps each pixel (u, v) in the depth image to its corresponding 3D spatial coordinates (x, y, z). The depth value of the pixel is denoted as $D(u, v)$, which is the distance between the pixel and the camera's optical origin. This transformation relies on the camera's intrinsic parameter, which is a fundamental attribute of a camera. This intrinsic parameter M is represented by a 3×3 matrix:

$$M = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Here, (f_u, f_v) and (u_0, v_0) in the matrix represent the focal lengths along the u, v axes of the depth map and the coordinate of the principal point. Using these intrinsic parameters, we can convert the pixel (u, v) from the depth data into the point (x, y, z) in 3D point clouds using the following transformation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M^{-1} D(u, v) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3}$$

We further calculate normal vectors $\boldsymbol{n}$ as additional features to reflect local geometric properties and orientations at individual point (x, y, z) within a 3D surface [41]. These normal vectors function as descriptors of local features, enabling the effective capture of the spatial attributes inherent in speech. Concatenating with normal vectors enhanced the model's ability to comprehend local surface properties, enabling it to jointly capture the distribution of the lip shape changes and motion cues along the speech actuation, thereby improving the performance and deepening our model's understanding of complex speech dynamics.

To calculate normal vectors at individual points, we denote the input point clouds video as $\boldsymbol{V}$ and its tth frame as a collection of 3D points, further denoted as $\boldsymbol{P_t} = \{p_i | i = 1, ..., N\}$ with N being the number of the points in one frame. We compute the normal vector for every point. Specifically, we find up to 30 adjacent nearest neighbors of the point within the search radius equals 0.1 m, and calculate the principle axis of the adjacent points using covariance analysis as its normal vector. Each individual point $p_i = (x, y, z, n_x, n_y, n_z)$ within our input data is characterized by its spatial coordinates (x, y, z) with additional three dimensions of estimated normal vectors (n_x, n_y, n_z).

4.3.2 Point Cloud Standardization and Alignment. Now that the input point cloud videos $\boldsymbol{V}$ have been organized into the shape of $(6 \times N \times L)$, where N and L denote the number of points in a single

frame and the total frames of an entire spoken speech (i.e., sentences or command words), respectively. Considering the natural diversity in individuals' mouth sizes and shapes, we perform a standardized process for the input point cloud videos. First, we randomly sample the number of points N in each frame to 1024, then normalize each frame of the input point clouds to a unit ball and the furthest point from the centroid of the cloud will land on the unit ball surface. Subsequently, we calculate the centroid of the point clouds of each frame and relocate the point clouds to ensure their centroids are positioned at the origin of our coordinate system. The normal vector features were estimated based on the standardized point clouds. The standardization process can enhance the model's adaptability, facilitating effective generalization across diverse user profiles and thereby ensuring consistent and robust performance across varying mouth shapes and sizes.

We developed an affine transformation as a part of our model, which we hoped to make insensitive and thus generalizable to different sensor perspectives across uses and device locations. To achieve this, we designed a transformation network (TNet) inspired by the model PointNet [45] to predict the affine transformation matrix (Fig. 1). Each frame of point clouds is fed into TNet, and rotated independently using the affine transformation matrix outputted by TNet. It takes the point cloud videos $\boldsymbol{V} \in \mathbb{R}^{6\times1024\times L}$ and regresses to a 3D rotation matrix $\boldsymbol{A} \in \mathbb{R}^{6\times6\times L}$, with which we transform $\boldsymbol{V}$ into aligned point cloud videos $\boldsymbol{V_{aligned}} \in \mathbb{R}^{6\times1024\times L}$. This TNet is trained together with the following feature extraction and sentence decoding part of our model, with the same loss function. The insertion of TNet allows our inference pipeline to be more robust against variances in user face orientations and different device locations.

4.3.3 Spatio-Temporal Features Extraction. To encode temporal information, we introduced another dimension t related to the point (x, y, z, n_x, n_y, n_z), where t means the point is in the tth frame of the point cloud videos. A time series of point cloud frames are then fed into a point 4D convolutional layer proposed in [12] for the feature extraction process (Fig. 1). Unlike convolutions on images where each x-y coordinate is guaranteed to have a pixel value, convolutions on a point cloud frame might come across empty or sparse regions due to occlusion or sensing inhomogeneity. To optimize this in the point cloud 4D convolution, we downsample each frame in the point cloud videos with a spatial subsampling range R_s using the Farthest Point Sampling (FPS) method.

With this method, we identify $N_{anchor} = N/s = 1024/16 = 64$ as anchor points in each frame, s is the sample rate. Using anchor points as centroids, we define a spatio-temporal local region G with the spatial radius R_s which defines the searching area within the current frame and the temporal kernel size R_t which defines the adjacent R_t nearest frames of the current frame $\boldsymbol{P_t}$ (Fig. 1). The point 4D convolution layer then encodes each of these local regions into a feature vector using Equation 4 and a multilayer perception. These feature vectors will append with the anchor points into a max pooling layer as features for the next step. Of note that the max pooling layer also addresses the "unordered" characteristic of point clouds – when represented as sequences, any point can appear at any index position without altering the depth data represented by point clouds.

$$\boldsymbol{F_t^{(x,y,z)}} = \sum_{(x',y',z')\in G} \boldsymbol{W_d} \cdot \left(\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} - \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) + \boldsymbol{W_f} \cdot \begin{bmatrix} n'_x \\ n'_y \\ n'_z \end{bmatrix} \tag{4}$$

In this function, $\boldsymbol{W_d}$ and $\boldsymbol{W_f}$ are the weights for 4D displacement transformation between the surrounding point $(x', y', z')_{t+\delta_t} \in G$ and the anchor point $(x, y, z)_t$. These weights increase the point feature dimensions to improve the feature representation ability. $\boldsymbol{F_t^{(x,y,z)}}$ is the encoded feature vector of the local region with the anchor point (x, y, z) as the centroid.

Subsequently, we employed a video-level spatio-temporal transformer proposed in [13], to search and merge feature vectors extracted from the Point 4D convolution, across the whole point cloud videos (Fig. 1). To generate a global feature representation of the utterance video, we appended a max pooling layer right after the transformer, which effectively combines the localized features into global features. This resulting global feature is fed into two bi-directional Gated Recurrent Unit (Bi-GRU) [6] layers, which are widely used in distinguishing similar visemes with the long temporal context in solving lipreading problems [4, 14, 22, 63, 64, 70]. It enables a comprehensive mapping between lip movements and the corresponding text representation, allowing a point cloud frame to refer to both former and later frames in the time series, thereby enhancing the model's capacity for robust motion modeling throughout the entire speech sequence. This approach ensures that our model effectively recognizes complex speech dynamics by considering both past and future contextual cues. The output from the Bi-GRU layer undergoes a softmax layer to produce probabilities for each token, which we choose to represent as English characters.

4.3.4 Sentence Decoding and Loss Function. We decode the time series of point cloud frames into English characters (including the space character) sequences directly. This selection of primitive tokens improves the generalizability of our system, enabling users to input sentences composed of various combinations of alphabet characters in English. This selection of tokens also allows us to recognize speech with various lengths and thus have a uniform pipeline for recognizing both sentences and command words. With these recognized English characters as intermediate recognition results, we establish a mapping between depth data of speech and textual information. However, one issue that arises from this approach is the mismatch between the number of point cloud frames L and the number of characters in the speech. As to the labeling of ground truth data for training, precisely aligning the durations of visual and character sequences of utterance can be challenging and prone to errors. To overcome this alignment challenge, we employ the Connectionist Temporal Classification (CTC) Loss [15], which is an objective loss function that offers flexible alignment between the input point cloud videos and the target sentence sequences. This flexibility eliminates the need for precise viseme alignments, simplifying the training process considerably, for which many speech recognition systems utilized the CTC approach in the decoding process [2, 4, 34, 36, 43, 63].

In our approach, we use greedy search to decode the output sequence, which selects the most likely character at each time

step, generating a sequence with the highest overall probability according to the model's output probabilities efficiently.

4.3.5 Commands Recognition. As previously described, our command word detection uses the same pipeline as sentence recognition. However, command words are often shorter than sentences, yielding character sequences out of context and thus infeasible to be corrected given context. In response, we implement a heuristic layer for command recognition, mapping character sequences to command words within a predefined command set shown in Table 2. Specifically, once we obtain the predicted character sequences from the pipeline, we compare them with the commands that exist in the command set based on the gestalt pattern matching algorithm [28]. This algorithm is recursively applied to the segments of the sequences and yields the best-matched command as the final output.

5 DATA COLLECTION

In this section, we elaborate on the data collection procedure, including the corpus design, the apparatus used for data recording, and the experiment configuration.

5.1 Sentence Corpus Design

The English language is characterized by its complex grammar, which consists of eight distinct parts of speech: conjunctions, pronouns, nouns, verbs, determiners, adjectives, prepositions, and adverbs. These parts of speech play a fundamental role in the construction of sentences, as they each serve a specific grammatical function that contributes to the overall meaning and structure of the sentence. Each sentence is made up of a combination of these parts of speech, with each word serving a specific grammatical function. In this section, we build a sentence corpus that reflects the common words and structures used in conversations between humans and voice assistants. All the selected words were involved in the list of the most frequently used Alexa sentences according to the research [48] and example sentences from official guides for Siri and Google Home.

This list contains a total of 347 sentences, with 170 from *Siri*, 42 from *Alexa*, and 135 from *Google Home*. Our approach to building a sentence corpus involved first breaking down each sentence into its individual words, and then utilizing the nltk library [31] to identify the part of speech for each word. From there, we calculated the frequency of each word for each part of speech and selected the top 20 words for each category as candidates. We ultimately selected four words from twenty candidates in each category based on their potential to be used in a grammatically correct and semantically meaningful sentence. This involved considering the syntactical rules of the English language, as well as the typical ways in which these words are used in every conversation. Through this process, we were able to create a sentence corpus that reflects the most commonly used words and grammatical structures in speech, while also considering the practical use cases of voice assistants. The finalized sentence corpus is shown in Table 1.

The sentence corpus is a collection of words that are grouped into eight different categories, each containing five words. Drawing inspiration from the GRID dataset [7], our sentences are generated using a straightforward permutation structure: $Conjunction + Pronoun + Noun + Verb + Determinator + Adjective + Preposition + Adverb$. For instance, a resulting sentence from the aforementioned process is *"and what music play every popular from nearby"*. Within each category, there are five distinct word choices, therefore 5^8 possible combinations of words, resulting in a large number of potential sentences which we asked participants to speak in our data collection session (Section 5.3).

Table 1: Sentence corpus in our data collection.

WORD CLASS	COMMON WORDS
Conjunction	And, Or, That, If, Like
Pronoun	What, Me, Who, You, It
Noun	Music, Alarm, Volume, Message, Weather
Verb	Play, Switch, Continue, Set, Listen
Determinator	Every, Another, All, Some, This
Adjective	Popular, Fast, Happy, Upcoming, Warm
Preposition	From, About, Between, Until, After
Adverb	Nearby, Here, When, Back, Why

Table 2: Command set in our data collection.

SCENARIO	COMMANDS
Music and Podcast	Resume, Pause, Previous, Volume Up, Volume Down
Smart Home	Search, Turn On, Turn Off, OK Google, Hey Siri, Alexa
System Control	Start, Confirm, Accept, Cancel, Dismiss, Reject
Call and Text	Hang up, Emergency, Call Mom, Text Dad
Query	What time is it, What is the weather
Application Instructions	Take a Screenshot, Set a Timer, Get Directions Home, Send an Email, Open Twitter, Increase Brightness, Watch Netflix

5.2 Command Set Design

Furthermore, to investigate on silent speech interaction across various smart devices, we utilized a command set consisting of 30 common commands from different scenarios in our daily lives, as detailed in Table 2. Given that the previous sentence recognition models are formed by combining individual command words, resulting in sentences that lack meaningful context for humans, we believe that our command set provides a realistic representation of the types of speech input that participants commonly encounter in their everyday interactions with voice assistant systems.

5.3 Data Collection Session

5.3.1 Participants. To ensure the accuracy and reliability of our model performance, we recruited a group of 10 participants (6 Females) to participate in our data collection process. Additionally, 5 of the participants wore glasses, and 2 participants were native

English speakers without any discernible accents. Sequentially in the data collection, we placed our depth sensing device at three distinct sensor locations (Figure 4):

- On-Wrist: To replicate the conditions resembling the use of smartwatches, we strapped the iPhone to the user's right wrist using Velcro straps.
- On-Head: To simulate scenarios where the device is worn on the head such as VR headset or AR glasses, we affixed the phone to a helmet using a phone holder. The phone was positioned approximately 25 cm in front of the user's face.
- In-Environement: In this setup, we placed the iPhone on a table with a randomized tilt angle and distance, simulating the positioning of depth sensors that could be potentially integrated onto smart devices such as Google Nest.

The data collection process was limited to one hour and partitioned into two distinct parts. The first part focused on gathering data for sentence recognition, while the second part was about the collection of data intended for command recognition. In the first part, we shuffled the sentence corpus and picked 50 unique sentences from the list for each user in each of the three positions (On-Wrist, On-Head, In-Environement) – in total $3 \times 50 = 150$ utterances were recorded for each user. Notably, our approach ensured that these sentences remained exclusive across all participants and all three positions, guaranteeing that no sentence was repeated throughout our study. This also helps to eliminate any potential bias that could be introduced if a user were to encounter the same sentence multiple times. For the second part, participants were required to speak the commands from the command set shown in Table 2. For each command, participants were instructed to read each command three times at each device location. By having users read each command multiple times, we were able to collect a diverse range of data that could account for variations in pronunciations, angles, and other factors.

Participants were not explicitly instructed to remount the device unless they expressed a desire to adjust the sensors for comfort. However, throughout the user study, participants were permitted to use natural postures they felt comfortable with. Data collection from each sensor location took around 20 minutes during which participants took rests. For example, participants lowered their arms down for breaks and raised their arms again to continue the data collection. We also observed participants constantly adjusting their arms and body postures during data collection, which resulted in changes in the relative positions between sensors and users' lips. The only exception was the On-Head sensor location at which the sensor stayed at relatively constant positions with user's lips. However, we did not observe significant differences between this location with the On-Wrist and In-Enrivonment sensor locations later in the result sections. This result proves the robustness of our system where we incorporate a region of interest detection and leverage depth data in concert with TNet to mitigate position variations. Overall, our user study resulted in a cumulative total of $3 \times 30 \times 3 = 270$ utterances for each participant. We measured an average speaking speed over 10 participants across 3 sensor locations of 2.2 words/s (SD = 0.32).

During the user study, we allowed participants to speak naturally. RGB videos from the front-facing camera of the iPhone 12 mini were also collected for later comparative evaluations between our system and prior work based on RGB videos. All the recorded data from the user study underwent a thorough review process to ensure accuracy and reliability. Any human errors were immediately addressed during the study, and broken data (e.g., participants stopped the recording by accident) were checked and removed when the study finished. This process ensured that only high-quality data were retained for the sentence and command recognition models. In total, 1470 sentences and 2673 commands were recorded during the user study.

Figure 4: Three sensor locations in the user: On-Wrist (left), On-Head (middle), and In-Environment (right).

6 EVALUATION

In this section, we assess the performance of our sentence and command recognition. We also compare these results with the outcomes of the state-of-the-art visual speech recognition model that utilizes RGB video data.

6.1 Evaluation Metrics

To evaluate the performance of our pipeline in interpreting spoken sentences using point cloud videos, we used Character Error Rate (CER) and Word Error Rate (WER) as evaluation metrics to measure the system performance. Character Error Rate measures the accuracy of individual character recognition in the predicted sequences with the true sentences. It quantifies the percentage of incorrectly recognized characters in the entire predicted sequence. The CER values were calculated using the following equation:

$$CER = \frac{S + D + I}{N} \tag{5}$$

In this equation, S represents the number of substitutions of incorrectly recognized characters, D represents the number of deletions for missed characters, I represents the number of insertions of extra characters recognized in the predicted sequences, N equals the total number of characters in the true sentences.

WER also evaluates the recognition performance, but it extends the evaluation closer to the application level. Specifically, WER quantified the percentage of incorrectly recognized words in the whole predicted sequences, which were also calculated using the equation 5, but with words as tokens.

Since our model uses characters as tokens for sentence decoding, our model may produce a word that does not exist in English. Therefore, we refine the raw output of the model using the spelling correction module in the TextBlob package [32], which replaces misspelled words with those that exist in the Project Gutenberg eBook's dictionary, while maximizing the frequency of intended

correction word. Both CER and WER metrics are calculated on the auto-corrected texts.

6.2 Comparative Evaluation on Sentence Recognition

To verify the feasibility of using depth sensing as a novel information source for visual speech recognition, we used a conventional RGB-based method as a baseline for comparison. We chose the off-the-shelf model from [35], which has achieved state-of-the-art on public visual speech recognition benchmark LRS3 [1]. The model uses a Conformer [16] as the frontend to encode RGB images along with a hybrid CTC/attention as the decoder. However, since our model uses a pure CTC architecture, we modified the video model by removing the attention loss from its objective function. This modification kept the two models as similar as possible for a fair comparison, as we mainly focused on investigating 1) which sensing modality could enable more accurate silent speech recognition, and 2) whether our model, which combines point 4D convolutional layers with a learnable transformation network, could exploit information from point clouds efficiently.

6.2.1 Within-user Performance. We ran within-user tests to investigate the proposed method's generalization ability to unseen utterances and phrases. Specifically, we conducted a 5-fold validation. Each fold contained 20% of all utterances from every sensor location of every participant after the collected lists of utterances were shuffled. Since we collected an equal number of utterances from each of the three sensor locations for every participant, each fold contained a balanced proportion of utterances from all sensor locations for each participant and overall. It is noteworthy that there were no duplicated utterances included in both the training and testing datasets. Following the same within-user protocols, we trained the baseline RGB model on the paired RGB data collected from our data collection session. Our PointVSR model and the RGB model were trained with the same hyper-parameters, i.e., 250 max epochs with a maximum learning rate of 0.01, adjusted by the OneCycleLR scheduler [49] in PyTorch, and the same spelling correction method was applied to the RGB model.

As shown in Fig. 5 right, overall, PointVSR outperformed the conventional visual speech recognition method (hereafter referred to as VideoVSR) with CER of 4.13% and WER of 8.06%, compared to the CER of 7.95% and WER of 13.02% in VideoVSR method, yielding relative improvements of 48.05% in CER and 38.10% in WER. This significant improvement confirmed our recognition pipeline as a promising method for enabling more precise and reliable silent speech interactions.

To investigate how the recognition accuracy varies among participants, we break down the results for each participant, shown in Fig. 6. We observed that the two native American English speakers, P2 and P6, both had better accuracies than the average. PointVSR achieved the best performance on P10 (WER 5.10%), who is not a native English speaker but can speak English almost as frequently and accent-free as a native speaker. In contrast, P9 had significantly higher CER and WER, which we suspect were caused by their noticeable accent, which likely deviated their lip movement patterns away from the rest of the participants, creating a data minority that posed challenges to deep learning. Therefore, we anecdotally note

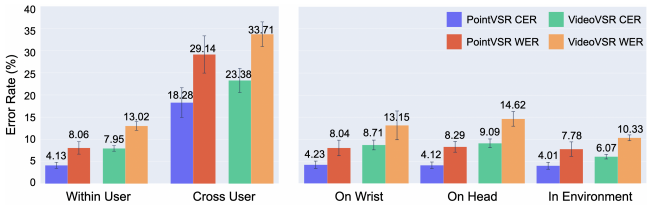

Figure 5: Comparison of sentence recognition performances with different sensing modalities/models, breaking down on within- and cross-user train-test methods (left), and on sensor location (right). Error bars indicate standard deviations across participants.

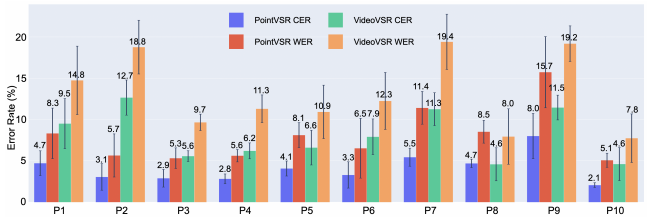

Figure 6: CER and WER from the within-user performance evaluation with error bars indicating standard deviations across folds.

that proficiency in English can be one of the dominant factors in the performance of silent speech recognition, at least given a modest magnitude of data collection. This issue is not unique to PointVSR, as commodity voice recognition systems also are more likely to fail on stronger accents and oftentimes require users to perform speech clearly and loudly. However, we are hopeful that a more sizable data collection could mitigate the data minority problem and yield improved results, as prior deep learning inference systems in HCI have shown.

Depth sensing obtains distance information from the participant's face, thereby contributing highly consistent spatial features across various sensor locations. To verify how this factor affects our method's performance in real-world settings, we conducted a sensor-location analysis. In this section, CER and WER metrics are broken down to three different sensor locations. We also evaluated the RGB-based method under the same protocol for comparisons. The results of these two methods are depicted in Fig 5 left. Overall, PointVSR achieved consistently better performance across the three sensor locations than VideoVSR. Furthermore, PointVSR demonstrated smaller variances of performance across sensor locations compared to those of VideoVSR, indicating a more consistent recognition performance against varying face orientations and sensor distances. The robustness of our method can be particularly advantageous for silent speech applications on wearable and mobile devices, as these devices often are positioned differently relative to a user's face during uses.

6.2.2 Cross-user Performance. In order to gauge PointVSR's performance when generalizing to unseen speakers that do not exist in our dataset, we performed a 5-fold cross-user validation with

each fold containing two participants' data. Fig. 5 right illustrates the results. Both CER and WER increased when moving to cross-user from within-user protocol because speech signals are often highly personalized, varying significantly from person to person, and thus a model can be improved by having training data from a user. The CER and WER measured 18.28% and 29.14% respectively with PointVSR. Still, our method compared favorably to the RGB-based method, achieving a WER and a CER decrease of 4.57% and 5% than VideoVSR, respectively. This generalizability enables our system to work better in an out-of-the-box manner, making it easier for unseen users to access reliable silent speech interactions without the need for calibrations.

In conclusion, PointVSR achieved better and more robust performance than the conventional RGB-based method. Furthermore, our model is much more lightweight than the RGB model – our model has 20 million parameters, which is only 8% the size of the RGB model that has 250 million parameters. This is an equally promising result as our superior performance and indicates that PointVSR, which uniquely leverages depth sensing, is potentially more efficient and easier to train than conventional models using RGB data.

6.2.3 Ad-hoc Analysis of Misrecognized Word in Sentence Recognition. We are interested in whether the misrecognized words would share any common patterns. To investigate this, we used the NIST speech recognition scoring toolkit (SCTK) [40] to analyze the frequencies of the types of errors in a word level on the raw outputs of our model (i.e., without spelling correction). Specifically, we counted the three types of errors including 1) *Substitution* (10.7%) when one letter is incorrectly replaced with another; 2) *Deletion* (0.7%) when a letter is omitted; and 3) *Insertion* (0.2%) when an extra letter is added to a word. We only ran this analysis on the within-user results. Results indicate *Substitution* being the dominant error type, taking up 92.2% of all errors, followed by *Deletion* (6.0%) and *Insertion* (1.7%). Furthermore, we found that *that* is the most confusing word in our vocabulary, where it is misrecognized in 99 out of 293 occurrences. Misrecognition includes words such as *like*, *and*, and *hat* as well as fabricated terms such as "TIAT", "THIT" and "TAND". Those errors are reasonable, as the inaccurately recognized words are highly similar to the target, making it inherently difficult to distinguish. However, by understanding the context using language models in naturally coherent sentences, we assume it is very possible to filter out improbable words or allow users to select from multiple most probable candidates. Additionally, the second most confused word *between* is misspelled as *betwen* for 45 times and *betweeen* for 16 times. This type of error could be solved by common spelling auto-correction.

6.3 Command Recognition

Our command recognition and sentence recognition are built on the same PointVSR model. To avoid duplicated insights from comparisons with the RGB method, we did not conduct a comparative evaluation, but focused on an in-depth analysis of our method's performance correlating with the viseme length of commands.

6.3.1 Command Recognition Evaluation and Results. In addition to sentence recognition, we conducted an evaluation on command recognition, including two rounds of experiments: 5-fold within-user evaluation and cross-user evaluation. For within-user evaluation, we trained the model on 80% of each participant's data (approximately 2139 command utterances from 10 participants) and tested it on the remaining 20% of each participant's data (around 428 command utterances). In cross-user evaluation, we divided the dataset into a testing set, comprising data from two participants, and a training set, composed of data from the remaining eight participants. This division followed a 5-fold cross-validation approach, consistent with the methodology used in cross-user sentence recognition. Both training and testing data in these two evaluations incorporated information from the three sensor locations shown in Fig. 4.

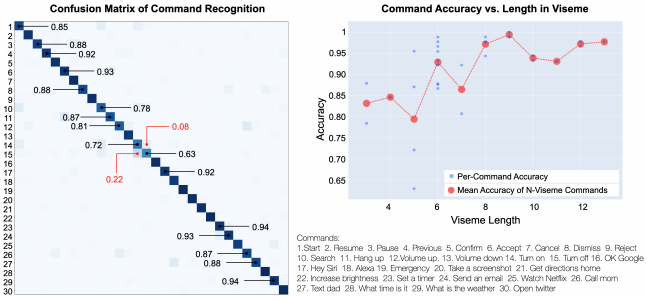

Figure 7: Left: Confusion Matrix of 5-fold within-user command recognition across 30 distinct commands. Accuracy is over 95% unless indicated otherwise. Right: Correlations between viseme length and recognition accuracy.

The confusion matrix described within-user command recognition accuracy is shown in Fig. 7. The average command recognition accuracy equals 91.33% (SD = 1.44%), which serves as strong evidence of our model's effectiveness in accurately differentiating 30 distinct commands of varying lengths. Furthermore, the confusion matrix indicates certain commands are more susceptible to being confused than others, often due to shared viseme sequences, pronunciations, and similar lip movements. For instance, the commands *Turn On* and *Turn Off* were frequently mistaken, likely due to their shared first half *Turn* and the prolonged period required for the lips to form the round shape for the character *O* in both *On* and *Off*. For the cross-user evaluation, the command recognition accuracy decreased to 74.88% (SD = 13.47%) due to user variance which decreased the performance in sentence recognition as well.

6.3.2 Correlation Analysis Between Viseme Length and Recognition Accuracy. We observed that commands such as *Start*, *Pause*, *Search* with short viseme sequences, are more errorful than commands with longer sequences. To better illustrate this phenomenon, we drew a correlation plot of viseme length against command recognition accuracy, as displayed in Figure 7 (right). In this plot, the x-axis represents the viseme length of commands. Note that the length is not a direct conversion from phonemes; rather, it results from the merging of adjacent identical visemes. For example, the viseme sequence of the command *Text Dad*, ['T', 'EH', 'K', 'T', 'T', 'T', 'EH', 'T'], was converted into ['T', 'EH', 'K', 'T', 'EH', 'T'] after we merged the three consecutive viseme 'T'. A clear trend in this plot shows an increase in accuracy as the viseme length of commands increases.

This key finding suggests to future researchers that in the design of silent speech command sets, preference should be given to longer, non-overlapping commands in order to optimize accuracy.

7 DISCUSSION

7.1 Ablation Study

We conducted an ablation study to get deeper insights into PointVSR by measuring how each of its components contributes to the recognition performance. Broadly, our model features four key components: 1) TNet; 2) 4D point cloud convolution layer; 3) Transformer; and 4) Bidirectional GRUs. We performed ablation tests by individually removing one component at a time and evaluating its impact on performance. Following a consistent data split approach within-user 5-fold validation, we opted for a subset of 2 folds from the total 5 and executed the experiments for each ablation test. To make a fair comparison, we kept the structures and parameters of the remaining components the same after one component was removed. We employed zero padding to expand dimensions when removing the 4D point cloud convolution and utilized adaptive pooling when removing the transformer or GRUs to reduce dimensions. Results are summarized in the table 3.

Overall, all components in our model play a vital role in the holistic functioning of the system as we observed significant increases in error rates when any of the components were removed, particularly when excluding either the TNet or GRUs components. In these cases, the word error rate exceeded 90%, indicating a substantial loss in the model's ability to interpret speech based on depth visual cues. Additionally, the 4D point cloud convolution and Transformer components serve as crucial feature extraction components throughout the entire pipeline, when the 4D convolution or the Transformer layer was removed, the model could still capture some aspects of speech, but the error rates increased substantially.

7.2 Handheld Sensor Location

We conducted an additional experiment to explore the *Handheld* sensor location, imitating the common way people hold their smartphones in their hands in daily use. We started by recording a new test dataset that consists of 50 new sentences from P1, following the same sentence composition rules outlined in Section 5.1. During the data collection phase, the iPhone 12 mini was, on average, positioned 22.45 cm away from the user's face, as determined by analyzing the depth map, similar to the use scenario shown in prior work [54]. With this dataset recorded from the handheld sensor location as a test set, we performed two distinct experiments to assess the model's performance in different contexts. The first, within-user cross-location, utilized training datasets from PI with the previous three sensor locations. The second experiment, cross-user cross-location, followed a similar approach as our previous within-user cross-location evaluation, with all data of P1 excluded from the training dataset.

We observed that the within-user cross-location experiment yielded a WER of 7.25% and a CER of 4.17%. Comparatively, the within-user within-location results from the user study averaging On-Wrist, In-Environment, and On-Head locations showed a very similar WER of 8.06% and a CER of 4.13%. When examining the cross-user cross-location scenario, we observed a WER of 28.00% and a CER of 18.99%. These outcomes also align with the earlier results from the cross-user study, where the WER was 29.14%, and the CER was 18.28%. The consistent performance across multiple sensor locations suggests the robustness of our method that could accommodate handheld use scenarios. This result highlights the adaptability of our method to different device form factors, orientations, and head/hand postures, and potentially to factors beyond the ones demonstrated in this work.

Table 3: Error rates of the ablation study where one component was removed at a time.

Ablation	WER	CER
\	**8.06%**	**4.13%**
TNet	90.49%	62.02%
Point Conv	24.43%	14.93%
Transformer	45.44%	30.18%
GRUs	97.44%	73.08%

7.3 Power Consumption and Computational Cost

Power consumption and computational cost are important factors to consider for our method's ecological validity. Our deep learning inference of one spoken sentence in the user study requires 66.66G floating-point operations with 20.43M parameters, which take 621 milliseconds for one sentence (i.e., 150 frames) on a server with four Nvidia RTX A5500 GPUs to complete. Specifically, among these operations, TNet takes 14.62G floating-point operations with 0.81M parameters, 4D convolution 0.80G floating-point operations with 0.56K parameters, Transformer layers take 51.24G floating-point operations with 0.32M parameters and GRUs take 1.38M floating-point operations with 19.28M parameters.

In real-world applications of our method, the power consumption of a system would comprise two components: the data acquisition and the computation. To investigate data acquisition, we recorded 1 hour of continuous operation of iPhone 12 mini's depth camera and analyzed the Powerlog file from the Battery Life profile. The depth camera consumed 110.50 mAh (i.e., 4.96% of iPhone 12 mini battery capacity of 2227 mAh), which is equivalent to 0.42 Wh assuming the working voltage of 3.83 V. The estimated consumption of the computation component falls within the range of approximately 0.86 to 5.41 W for inferencing one sentence with a length of 150 frames. This estimation is based on a linear extrapolation using the results from the model[2] with 5.59G floating-point operations, which requires consumption between 0.072 and 0.454 W. Of note, these numbers are theoretical speculations and should be perceived as an approximation and require further validation through device deployment in the future work.

7.4 Performance Comparison with Trimmed VideoVSR model

We used a state-of-the-art video-based VSR model as a baseline to evaluate our proposed model's effectiveness in learning point

[2]Calculation based on Hugging Face Distilbert model case study: https://machinelearning.apple.com/research/neural-engine-transformers

cloud data. This might not constitute a fair comparison in that video models have often been tailored to large-scale datasets and have a large number of parameters, thus demanding a lot of training data for the performance ramp-up. We conducted an additional series of tests to investigate whether the performance difference we have seen in previous evaluations stemmed merely from our model's smaller size, which makes it easy to train on our relatively small dataset, or truthful superiority future silent speech systems could rely on. Due to the lack of large-scale depth silent speech data, we optimized video models for small-scale data. Specifically, we created a trimmed, 20M version of the VideoVSR model, aligning it in scale with our PointVSR model. Specifically, we used 4 multi-head attention layers instead of 12, and reduced the latent dimension from 768 to 512 and the number of heads from 12 to 4. We then evaluated this model following the same protocols as in section 6.2.1 and 6.2.2. As shown in Table 4, the trimmed VideoVSR model exhibited the highest error rates, indicating that the strengths of our method originate from the superiority of the network architecture in concert with the unique leverage of depth information, rather than the use of small-scale training data.

7.5 Privacy Concerns of Depth Sensing

RGB cameras capture full-color images that can include identifying details such as facial features, skin colors, and clothing. In contrast, depth data contains information about the shape and distance of objects without capturing detailed visual textures or colors. This makes depth sensing less likely to contain information that can be used to identify individuals. Additionally, captures of background can be easily filtered out using distance thresholding in the depth map at hardware and software levels, thereby minimizing the risk of exposing unintentional information. In comparison to RGB-based systems that may require complex neural networks for filtering out irrelevant facial information, our depth-based approach inherently reduces the need for such complex algorithms, contributing to a more straightforward and privacy-conscious methodology. However, we acknowledge that the increasing resolution of depth cameras and the increasing capabilities of deep neural nets could make depth sensing as privacy concerning as other imaging approaches, and thus depth-based silent speech recognition systems should investigate privacy implications in user contexts.

7.6 Example Use Cases

To demonstrate our system, we developed a series of usage scenarios (Fig. 8 and also see Video Figure). Our method, as detailed in the paper, facilitated the recognition of silent speech in a wide array of use scenarios.

Table 4: Comparison between our PointVSR model and the original and trimmed VideoVSR model.

Model	Within-user		Cross-user	
	CER	WER	CER	WER
PointVSR	4.13%	8.06%	18.28%	29.14%
VideoVSR	7.95%	13.02%	23.28%	33.71%
VideoVSR-20M	19.05%	36.62%	39.84%	52.21%

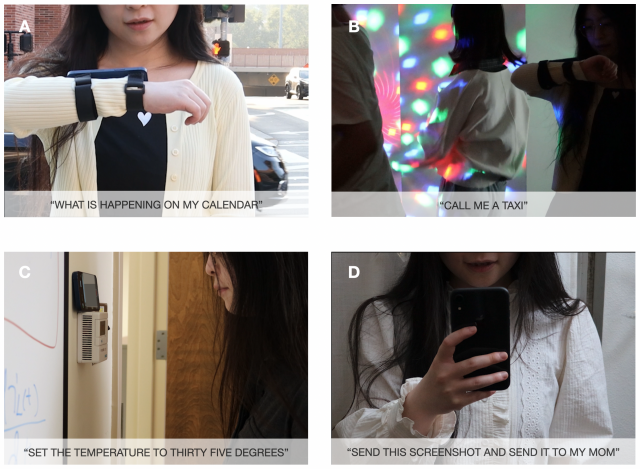

Figure 8: Example use cases that demonstrate our method's reliable performance in a noisy environment (A), and under complicated lighting conditions (B) with a wearable sensor location. Our method also works on deployed devices such as a smart thermostat (C), and can upgrade smartphones to recognize silent commands and sentences (D).

7.6.1 Smartwatch as a Robust Natural Language Interface to AI. We envision a future where smartwatches could serve as a natural and always available voice interface for users to interact with LLM-based AI agents. This requires smartwatches to have reliable sensing performance across a wide range of adversarial noise in both audio and video channels. In noisy environments, audio signals are often polluted, posing challenges for accurate speech recognition. In contrast, depth cameras are capable of capturing precise lip movements, providing a more resilient approach to speech recognition. In the use scenario shown in Fig. 8 (A), a smartwatch prototype, enhanced by our method, achieves accurate and reliable speech recognition results even amidst busy and noisy traffic surroundings. Though video/RGB-based silent speech recognition methods may have similar robustness in acoustically noisy environments, they might be susceptible to noise in the visible light channel and dim lighting conditions, and yield inconsistent performance with various skin tones. As shown in Fig. 8 (B), our system leverages advancements in depth sensing and its robustness against aforementioned noise and can accurately interpret sentences.

7.6.2 Flexible Sensor Location to Enable Device with Various Form Factors. Depth data maintains consistency across different sensor positions and orientations to a greater extent than RGB data, in that spatial characteristics of objects do not depend on the viewing angle, unlike color and texture information, which can change significantly with sensor-user perspective. In this regard, our method could yield more consistent data during constantly changing postures when users use smart devices, especially those deployed in the environment. Furthermore, our method incorporates an alignment process by TNet, contributing to its robustness by accommodating variations in orientation. As illustrated in Fig. 8 (C), users can silently control devices such as AC systems, showcasing the practicality of our system in diverse settings. Beyond smart environment applications, our method can seamlessly integrate with

smartphones shown in Fig. 8 (D) to allow existing applications on smartphones to recognize speech as a natural and intuitive interaction modality.

8 LIMITATION AND FUTURE WORK

As with all technologies, our proposed one has limitations and so does the evaluation of our work, which we acknowledge here to inspire ideas and encourage future work.

Demographic variety Foremost, the demographic variety of our participant pool is modest, which limited further insights we could draw regarding demographical factors on our proposed visual speech recognition. We suspect that our technique will share strengths as well as weaknesses as ones of depth sensing for our hitchhike on commodity depth sensing infrastructure. On the positive side, depth sensing can be as robust as FaceID, which has been proven successful across demographical variances in authentication tasks. Robustness against user variance is an important promise of our proposed technique to make visual speech recognition reliable and ultimately equitable across society. Nonetheless, studies with larger numbers and a more diverse set of participants are needed in our future work.

Deployment in the wild Additionally, further insights could be drawn by deploying our system in the wild, allowing its users to speak natural languages at will, and with a wide variety of factors of environments (e.g., vibrational noise, sun exposure) and user features (e.g., body posture, face feature). This would require our system to be implemented on a device with a proper cloud computing scheme or to be entirely standalone by running on the device, both of which require further system engineering which we plan to do in the future.

Other depth sensors Furthermore, our hardware selection is limited to the TrueDepth camera, a high-end depth camera that provides high-resolution depth data with high SNR. However, not all smart devices can afford to equip this sensor, limiting the scalability of this work to some extent. We acknowledge that future work could investigate mid- or low-end depth cameras with lower-resolution depth data or data with higher noise floors (i.e., low SNR) to learn about the performance of our technique on a wider spectrum of depth cameras. In this work, the input depth frames are downsampled to 1024 points (i.e., lower resolution by a factor of at least 3 than the original data), which imitates lower-resolution depth data. However, the effect of higher noise floors remains to be tested in the future.

Impovement with more training data Our model is more parameter-efficient while outperforming the conventional RGB model in recognition tasks. The lightweight model has lower computational costs and is thus easier to distribute on edge devices. However, since there are no existing large-scale point cloud datasets for speech recognition tasks, we were unable to assess our model's capability when scaling up the number of its parameters together with its training data. Recent research on machine learning has proven Transformers are data-efficient and scalable [20, 66], and we anticipate increasing the complexity of the model (e.g., using more heads and larger latent space dimensions for the multi-head attention architecture) should enhance the model's capability to fit more training data and leave this work for future explorations.

Beyond speech recognition Finally, depth data of human speech could lead to a wider array of use cases beyond speech recognition, such as health care, education, and embodied AI. Though establishing a dataset of human speech depth data is not within the scope of this work, we release the depth data we have collected under the approval of our institute's IRB, as well as the source code of our system implementation to facilitate the growth of this raising field of research. Our open-source repository is held at: https://github.com/hilab-open-source/WatchYourMouth.

9 CONCLUSION

We present *Watch Your Mouth*, the first silent speech recognition approach using depth as the sole source of information. We propose this approach to enrich the sensing capability of a wide variety of smart devices, allowing for speech-based interactions to be utilized in a highly socially acceptable and privacy-preserving fashion. We developed custom signal preprocessing and DL pipelines. Validations and studies conducted in this research show promise in using depth sensing for command and sentence recognition at three distinctive device locations: On-Wrist, On-Head, and In-Environment. Results indicate the superiority of our approach over conventional RGB-video-based silent speech recognition, decreasing within-user CER and WER by 3.82% and 4.96%, and cross-user CER and WER by 5% and 4.57% respectively.

REFERENCES

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. LRS3-TED: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496* (2018).

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2020. Asr is all you need: Cross-modal distillation for lip reading. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2143–2147.

[3] Ahmad Kamal Aijazi, Paul Checchin, and Laurent Trassoudaine. 2013. Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. *Remote Sensing* 5, 4 (2013), 1624–1650.

[4] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando De Freitas. 2016. Lipnet: Sentence-level lipreading. *arXiv preprint arXiv:1611.01599* 2, 4 (2016).

[5] Hrvoje Benko, Andrew D Wilson, and Federico Zannier. 2014. Dyadic projected spatial augmented reality. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 645–655.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[7] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. 2006. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America* 120, 5 (2006), 2421–2424.

[8] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. 2021. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems* 23, 2 (2021), 722–739.

[9] Peter B Denes and Elliot Pinson. 1993. *The speech chain*. Macmillan.

[10] Nathan Devrio and Chris Harrison. 2022. DiscoBand: Multiview Depth-Sensing Smartwatch Strap for Hand, Body and Environment Tracking. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[11] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, et al. 2020. DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 829–843.

[12] Hehe Fan, Yi Yang, and Mohan Kankanhalli. 2021. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14204–14213.

[13] Hehe Fan, Yi Yang, and Mohan Kankanhalli. 2022. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2022), 2181–2192.

[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] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. 369–376.

[16] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*. 5036–5040. https://doi.org/10.21437/Interspeech.2020-3015

[17] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 441–450.

[18] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 453–462.

[19] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 637–644.

[20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).

[21] 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.

[22] Minsu Kim, Joanna Hong, Se Jin Park, and Yong Man Ro. 2021. Multi-modality associative bridging through memory: Speech sound recollected from face video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 296–306.

[23] 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.

[24] 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.

[25] 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*. European Language Resources Association, Marseille, France, 6866–6873. https://aclanthology.org/2022.lrec-1.741

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

[27] John Kominek and Alan W Black. 2004. The CMU Arctic speech databases. In *Fifth ISCA workshop on speech synthesis*.

[28] Giuseppe Lancia and Ramamoorthi Ravi. 1999. GESTALT: Genomic steiner alignments. In *Annual Symposium on Combinatorial Pattern Matching*. Springer, 101–114.

[29] Fabio Lavagetto. 1995. Converting speech into lip movements: A multimedia telephone for hard of hearing people. *IEEE Transactions on Rehabilitation Engineering* 3, 1 (1995), 90–102.

[30] 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* (Reims, France) *(AH2019)*. Association for Computing Machinery, New York, NY, USA, Article 1, 9 pages. https://doi.org/10.1145/3311823.3311831

[31] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028* (2002).

[32] Steven Loria. 2018. textblob Documentation. *Release 0.15* 2 (2018).

[33] 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).

[34] Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic. 2023. Auto-AVSR: Audio-visual speech recognition with automatic labels. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[35] Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2021. End-To-End Audio-Visual Speech Recognition with Conformers. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7613–7617. https://doi.org/10.1109/ICASSP39728.2021.9414567

[36] Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2022. Visual speech recognition for multiple languages in the wild. *Nature Machine Intelligence* 4, 11 (2022), 930–939.

[37] Hiroyuki Manabe, Wataru Yamada, and Hiroshi Inamura. 2014. Tag system with low-powered tag and depth sensing camera. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 373–382.

[38] 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.

[39] Iain Matthews, Timothy F Cootes, J Andrew Bangham, Stephen Cox, and Richard Harvey. 2002. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 2 (2002), 198–213.

[40] National Institute of Standards and Technology. 2021. SCTK, the NIST Scoring Toolkit. https://github.com/usnistgov/SCTK.

[41] Daoshan OuYang and Hsi-Yung Feng. 2005. On the normal vector estimation for point cloud data from smooth surfaces. *Computer-Aided Design* 37, 10 (2005), 1071–1079.

[42] 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.

[43] Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Georgios Tzimiropoulos, and Maja Pantic. 2018. Audio-visual speech recognition with a hybrid ctc/attention architecture. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 513–520.

[44] 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.

[45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[46] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017).

[47] Zhou Ren, Jingjing Meng, and Junsong Yuan. 2011. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In *2011 8th international conference on information, communications & signal processing*. IEEE, 1–5.

[48] 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.

[49] Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, Vol. 11006. SPIE, 369–386.

[50] Themos Stafylakis and Georgios Tzimiropoulos. 2017. Combining residual networks with LSTMs for lipreading. *arXiv preprint arXiv:1703.04105* (2017).

[51] 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* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 696, 21 pages. https://doi.org/10.1145/3544548.3581465

[52] 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* (Virtual Event, Germany) *(ETRA '21 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 13, 6 pages. https://doi.org/10.1145/3448018.3458011

[53] William H Sumby and Irwin Pollack. 1954. Visual contribution to speech intelligibility in noise. *The journal of the acoustical society of america* 26, 2 (1954), 212–215.

[54] 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*. 581–593.

[55] Wei Sun, Franklin Mingzhe Li, Congshu Huang, Zhenyu Lei, Benjamin Steeper, Songyun Tao, Feng Tian, and Cheng Zhang. 2021. Thumbtrak: Recognizing micro-finger poses using a ring with proximity sensing. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*. 1–9.

[56] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Bjoern Hartmann, and Tovi Grossman. 2019. Loki: Facilitating remote instruction of physical tasks using bi-directional mixed-reality telepresence. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 161–174.

[57] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

[58] Jingxian Wang, Chengfeng Pan, Haojian Jin, Vaibhav Singh, Yash Jain, Jason I. Hong, Carmel Majidi, and Swarun Kumar. 2020. RFID Tattoo: A Wireless Platform for Speech Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 155 (sep 2020), 24 pages. https://doi.org/10.1145/3369812

[59] Andrew D Wilson and Hrvoje Benko. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings*

of the 23nd annual ACM symposium on User interface software and technology. 273–282.

[60] Kin-Yiu Wong. 2022. yolov7. https://github.com/WongKinYiu/yolov7.

[61] Robert Xiao, Teng Cao, Ning Guo, Jun Zhuo, Yang Zhang, and Chris Harrison. 2018. LumiWatch: On-arm projected graphics and touch input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.

[62] Robert Xiao, Chris Harrison, and Scott E Hudson. 2013. WorldKit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 879–888.

[63] Kai Xu, Dawei Li, Nick Cassimatis, and Xiaolong Wang. 2018. LCANet: End-to-end lipreading with cascaded attention-CTC. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 548–555.

[64] Shuang Yang, Yuanhang Zhang, Dalu Feng, Mingmin Yang, Chenhao Wang, Jingyun Xiao, Keyu Long, Shiguang Shan, and Xilin Chen. 2019. LRW-1000: A naturally-distributed large-scale benchmark for lip reading in the wild. In *2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019)*. IEEE, 1–8.

[65] Shang Zeng, Haoran Wan, Shuyu Shi, and Wei Wang. 2023. MSilent: Towards General Corpus Silent Speech Recognition Using COTS MmWave Radar. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 39 (mar 2023), 28 pages. https://doi.org/10.1145/3580838

[66] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. 2022. Scaling Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12104–12113.

[67] Ruidong Zhang, Hao Chen, Devansh Agarwal, Richard Jin, Ke Li, François Guimbretière, and Cheng Zhang. 2023. HPSpeech: Silent Speech Interface for Commodity Headphones. In *Proceedings of the 2023 ACM International Symposium on Wearable Computers*. 60–65.

[68] Ruidong Zhang, Mingyang Chen, Benjamin Steeper, Yaxuan Li, Zihan Yan, Yizhuo Chen, Songyun Tao, Tuochao Chen, Hyunchul Lim, and Cheng Zhang. 2021. SpeeChin: A Smart Necklace for Silent Speech Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–23.

[69] Ruidong Zhang, Ke Li, Yihong Hao, Yufan Wang, Zhengnan Lai, François Guimbretière, and Cheng Zhang. 2023. EchoSpeech: Continuous Silent Speech Recognition on Minimally-obtrusive Eyewear Powered by Acoustic Sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[70] Yuanhang Zhang, Shuang Yang, Jingyun Xiao, Shiguang Shan, and Xilin Chen. 2020. Can we read speech beyond the lips? rethinking roi selection for deep visual speech recognition. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*. IEEE, 356–363.

[71] Yuhang Zhao, Elizabeth Kupferstein, Brenda Veronica Castro, Steven Feiner, and Shiri Azenkot. 2019. Designing AR visualizations to facilitate stair navigation for people with low vision. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 387–402.

[72] Yuhang Zhao, Sarit Szpiro, and Shiri Azenkot. 2015. Foresee: A customizable head-mounted vision enhancement system for people with low vision. In *Proceedings of the 17th international ACM SIGACCESS conference on computers & accessibility*. 239–249.

[73] Jian Zhu, Cong Zhang, and David Jurgens. 2022. Phone-to-audio alignment without text: A semi-supervised approach. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8167–8171.