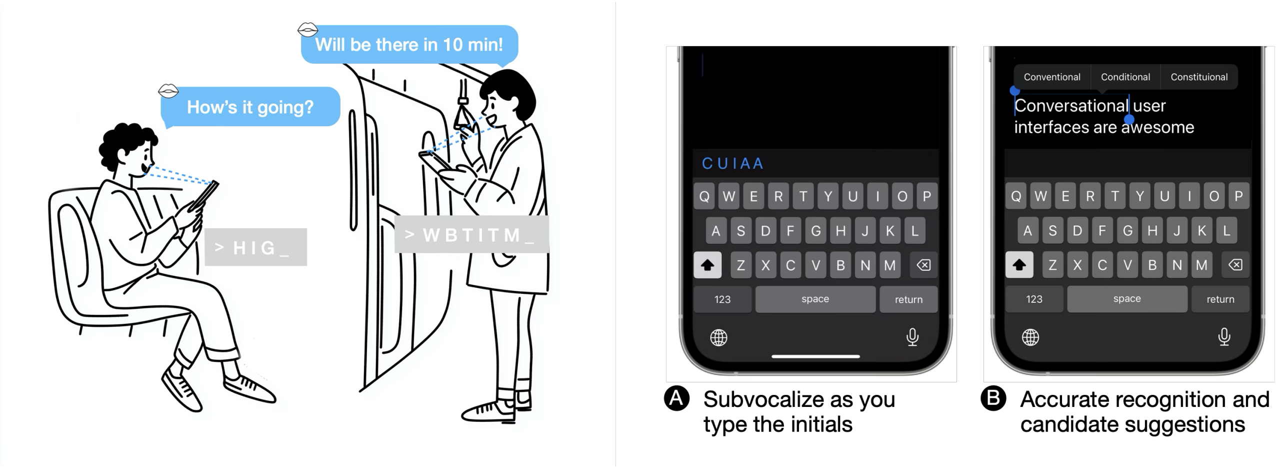
Multimodal Silent Speech-based Text Entry with Word-initials Conditioned LLM

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

Shitao Fang
The University of Tokyo
Interactive Intelligent Systems Laboratory
Tokyo, Japan
fst@iis-lab.org

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



A Subvocalize as you type the initials

B Accurate recognition and candidate suggestions

Figure 1: LipType enables efficient and private text entry on mobile devices in various daily scenarios. The system performs high-performance silent speech recognition and candidate suggestion using only initial letters as complementary information, providing effortless and faster text input experiences even in one-handed situations.

Abstract

Although exhibiting great potential in enabling seamless communication between humans and conversational agents, large vocabulary recognition is still challenging for silent speech interfaces. In this research, we propose a novel interaction technique that combines silent speech and typing to enable more efficient text entry while preserving privacy. This technique allows users to use abbreviated phrase input while still ensuring high accuracy by leveraging visual information. By fine-tuning a large language model with a visual speech encoder, we condition the models to decode the speech content with word initials as hints. Evaluations on existing datasets show that our model can reduce the Word Error Rate from 20.3% to 9.19%, compared to state-of-the-art visual speech recognition models. Results from a user study demonstrated significant improvements in input speed and keystroke saving. Participants reported that our prototype, *LipType*, leads to an overall lower perceived workload, particularly in the effort and physical demand dimension.

CUI '25, Waterloo, ON, Canada

ACM ISBN 979-8-4007-1527-3/25/07
https://doi.org/10.1145/3719160.3736612

CCS Concepts

• **Human-centered computing** → **Text input**; **Interaction techniques**; **Sound-based input / output**; • **Computing methodologies** → **Speech recognition**.

Keywords

Text Entry, Multimodal Interface, Silent Speech, Touch Input

ACM Reference Format:
Zixiong Su, Shitao Fang, and Jun Rekimoto. 2025. Multimodal Silent Speech-based Text Entry with Word-initials Conditioned LLM. In *Proceedings of the 7th ACM Conference on Conversational User Interfaces (CUI '25), July 08–10, 2025, Waterloo, ON, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3719160.3736612

1 Introduction

Efficient text entry on mobile devices, which typically relies on virtual on-screen keyboards, remains a challenge due to small screen sizes and the lack of haptic feedback. These constraints often lead to reduced typing speed and accuracy and pose significant challenges to efficient text input. Previous research in HCI has been putting significant effort into solving this issue by proposing gesture-based typing techniques [5, 57], auto-correction methods [59], and predictive input methods [27], and other multimodal schemes [46, 61]. Recently, the emergence of powerful Large Language Models (LLMs), which are capable of processing and performing complicated text-based tasks, has enabled more complicated text-based tasks such as co-programming with AI [34], robot control [36], and cognitive-behavioral therapy [13]. However, it also elicits an urgent demand for efficient and seamless interactions with conversational agents through natural language-based communication.

In this work, we propose a novel technique that integrates silent speech with traditional typing to enable more efficient text entry while preserving user privacy. Drawing from Baddeley's model of working memory and the phonological loop theory [3], which underlines that our short-term memory involves a subvocal rehearsal process, we explore the possibility of natural and intuitive interaction with simultaneous silent speech and typing input. Silent speech interaction leverages non-acoustic signals to infer the content of speech, which is a promising alternative for voice interfaces by providing rapid input while minimizing privacy concerns, especially in situations where it is inappropriate to speak aloud (e.g., in public or during a meeting) or speech recognition is not reliable (e.g., in noisy places). Researchers have explored different sensing modalities such as ultrasound sensing [24], EMG [21], RFID sensing [51], EEG [37], and video cameras [44] to obtain speech-related signals by tracking the movement of articulators, and developed methods to decode human speech from such data. The video-based method, or lip-reading, can provide the best recognition performance because of its high spatial and temporal resolutions, and the prevalence of cameras has made it easy to access by end users and produced large datasets. However, current research indicates that we are still far from practical lip-reading systems that ensure high accuracy. The state-of-the-art model on open datasets marks a Word Error Rate (WER) around 20.3% [28], resulting in a significant gap between speech recognition models such as OpenAI's Whisper [38] (2.7%). The reason is multi-dimensional, such as insufficient data and inherent ambiguity of lip movements. Furthermore, voiced speech and subvocalized speech show a different pattern in lip movements [44], leading to a considerable performance loss to lip-reading models when they are actually used in a silent manner.

With the abundant visual information captured from the camera on a smartphone, we anticipate that the initial letter of each word in the sentence would provide essential information that significantly alleviates the uncertainty of silently uttered speech, enabling high-performance lip-reading. Furthermore, given that the average length of English words is approximately 5 letters [6], ideally, typing only word initials would save about 80% keystrokes. Reducing the number of keystrokes is valuable in text entry tasks because it leads to lower motor costs and higher input efficiency. To verify the feasibility of this idea, we propose a multimodal machine learning approach that combines visual and text input, which thereby leverages decoder-only LLMs to reconstruct the speech content conditioned by the word initials. Through offline evaluations on public lip-reading datasets, we found that our approach leads to a much lower 6% Word Error Rate (WER) compared to the state-of-the-art visual ppeech recognition (VSR) model's 20%. This substantial improvement encouraged us to build a prototype, *LipType*, on iOS to further investigate the usability of multimodal silent speech and typing interaction. The prototype is intended to serve as an alternative to traditional typing methods and provide fast and accurate text entry functions, supporting a wide range of applications. We integrated necessary functionalities, including candidate suggestion and automatic correction, and conducted a user study in comparison with the traditional typing-only method. During the study, we measured the system's performance in three dimensions: input speed, keystroke savings, and error rate. Furthermore, NASA-TLX and System Usability Scale (SUS) tests were used as subjective assessments to measure perceived workload and usability. Overall, with our prototype, LipType, participants were able to enter text faster, achieving a 33.10% higher Word Per Minute (WPM) compared to the traditional typing method, averaged across two-handed and one-handed conditions. By providing only word initials as text input, LipType saved 67.59% keystrokes, with the correction process taken into account. Participants also reported that they perceived lower physical and temporal demand while spending less effort using LipType, but the mental demand became higher instead. This result is within our expectations as typing word initials during text entry introduces additional learning costs, as users have little experience with this novel input method. We believe that both the mental demand and the user's performance will be further improved during day-to-day use, envisioning a future where the bandwidth of human-AI communication can be boosted by our approach with no privacy concerns or social acceptance issues.

This work makes four key contributions. To summarize, we

1. proposed a new multimodal training paradigm that uses an LLM to take word initials and videos as input for silent speech recognition,
2. performed a thorough offline test to evaluate the performance of the word initials conditioned VSR model and find implications,
3. developed a prototype on mobile devices that assists users in typing word initials while speaking, integrated with candidate suggestion and auto-correction features,
4. conducted a user study with traditional typing as the baseline, measuring the proposed method's performance in terms of input speed and workload.

2 Related Work

To bridge the gap between conventional typing and silent speech research, we first review related work in text entry techniques on mobile devices and silent speech recognition methods and interfaces.

2.1 Text entry techniques on mobile devices

Improving the efficiency of text entry on mobile devices with constrained keyboard size has been a longstanding focus of HCI research. Early work pioneered optimizing the keyboard layout [4, 32, 56] and designing gesture-based typing techniques [5, 57]. As machine learning technology advances, researchers have developed more sophisticated models to decode keyboard input with errors [35] or recognize on-keyboard gestures for text editing. Furthermore, language models (LMs) are trained on large corpora and used to suggest erroneous texts [8, 58] or perform automatic corrections at the phrase level as the user keeps typing [59]. Those keyboard-only techniques have less learning cost and don't require additional equipment. However, they still rely on massive keystrokes and fall short in input speed. In response, T. Li et al.

developed a specialized LM to decode abbreviated input and save keystrokes [27]. Similarly, SkipWriter enables abbreviated handwriting input [54] with an LLM decoder to save motor movements. However, the abbreviated forms of words contain substantial ambiguity, which provides limited recognition accuracy and leads to additional effort due to error correction.

Seeking better efficiency, efforts to integrate multiple sensors and input channels for text entry have also provided magnitude insights. For example, TAGSwipe [26] combines gaze swiping and touch and uses a button to confirm selection or indicate the start and the end of a swipe gesture. Hummer [17] pushes forward this idea for hands-free text entry by using humming sounds to replace touch input. Multimodal voice-based techniques are the most related to our work, which benefits from the natural, fast, and easy-to-learn nature of human speech. B. Suhm et al. [46] proposed a system to enable cross-modal repeating with pen gestures to correct recognition errors, which outperformed keyboard and mouse input. EyeSayCorrect [61] allows the user to locate the misrecognized words with gaze and perform corrections by respeaking. K. Sim investigated the approach of augmenting speech modality with touch events [41, 42], which adopts a similar route to our work. They thoroughly discussed the possible integration forms of speech and touch, including an initial letter of word vs. boundary of sentences, touch keyboard vs. gesture keyboard, and isolated input (word by word) vs. continuous input, and found the error rate was reduced when decoding speech with touch events. This body of research provided valuable insights into the design space of multimodal speech and touch text entry interface as silent speech and speech share similar implications in many aspects. For example, touch input can result in slower speech speed, and fuzzy letters will multiply the unreliability of early-stage speech recognition models. However, since most results were acquired from simulated data, the usability of such interaction in real world remains understudied. Furthermore, the rapid evolution of Automatic Speech Recognition (ASR) models in recent years has pushed forward the speech recognition performance to human levels, which makes it less effective to incorporate touch input during the decoding stage. In contrast, silent speech holds the promise of seamless and privacy-preserving communication between humans and computers, but the inherent ambiguity of visual speech remains a significant barrier to large-vocabulary continuous recognition. In this work, we aim to develop a high-performance recognizer to enable real-world feasibility, thereby conducting a detailed user study to shed new light on the usability and implications of the combination of silent speech and touch input.

2.2 Silent Speech Recognition Interfaces and Methods

Silent speech interfaces (SSIs) have explored a range of sensing methods to capture the underlying biosignals associated with speech production. SSIs are designed to enable confidential and natural communication by interpreting speech-related signals without relying on audible sound. A variety of sensing techniques have been investigated over the years. For example, electromagnetic articulography (EMA) [12, 15, 40] has been used to track the movements of speech articulators, while ultrasound imaging offers a means to visualize vocal tract and tongue dynamics [19, 23–25]. Other approaches include capturing subtle acoustic signals like non-audible murmur (NAM) and ingressive speech [48, 49], which involve sensors tuned to pick up sounds that are typically too faint for conventional microphones. Further sensor modalities extend to placing capacitive sensors inside the mouth to directly measure articulator movements [22] and using surface electromyography (sEMG) to capture the electrical activity of facial muscles during speech [21]. Yet, among all these approaches, lipreading-based SSIs have attracted significant research interest due to their non-intrusive nature and potential for mobile deployment. Besides RGB cameras [44, 47], infrared cameras [60] and depth cameras [52] are used for low-light environments and privacy protection. However, most research focuses on command-level recognition [43, 45], as continuous word-level recognition is challenging, and even large models with Transformer backbones [50] can suffer more than 50% WER in real-world applications. Therefore, we consider that it might not be feasible to pretrain a one-size-fits-all lip-reading model that has robust recognition with only video input. In response, we propose using a multimodal approach to complement silent speech with word initials as priors with minimal additional effort and take advantage of the latest advances in natural language processing research.

3 Initials Conditioned LLM for Visual Speech Recognition

3.1 Model Architecture

Large language models trained on a vast amount of text data have shown great potential in understanding human language, and recent research on integrating speech recognition models with LLMs has suggested its effectiveness in assisting speech dictation. We draw inspiration from recent papers in ASR [29] and build a new learning pipeline fuse video and initial letters embeddings to condition the LLM decoder. As shown in Figure 2, we freeze a pretrained VSR model to encode lip videos and only train a lightweight linear projector to align the visual speech embeddings to the same latent space of the LLM's text embeddings. We borrow the best open-source VSR model from Auto-AVSR [28], which produces a 768-dimension feature vector at each time step t, i.e., a video frame. Since the sample rate of a video (usually 25 fps in VSR research) is much higher than the speech content represented as text tokens and could dominate the prompt, a downsampling layer of factor 2 is used to unfold the feature vectors, resulting in a visual embedding of shape $[T/2, 1536]$.

For the LLM module, we use the LLaMA 3.2 model family [11], keeping the learned parameters with a low-rank adapter (LoRA) [18] as the only trainable module, so as to preserve its knowledge and mitigate overfitting. The visual embeddings V are inserted into a prompt template following the LLaMA 3 prompt format[1]: "< $|start\_header\_id|$ >user< $|end\_header\_id|$ > <V>, transcribe the lip video to text with the initial letters <I>< $|eot\_id|$ >< $|start\_header\_id|$ >assistant< $|end\_header\_id|$ >", where <I> represents the initial letters, concatenated with spaces in upper case, and < $|eot\_id|$ > stands for the end of the turn. We empirically set

[1] See https://www.llama.com/docs/model-cards-and-prompt-formats/meta-llama-3/ for details

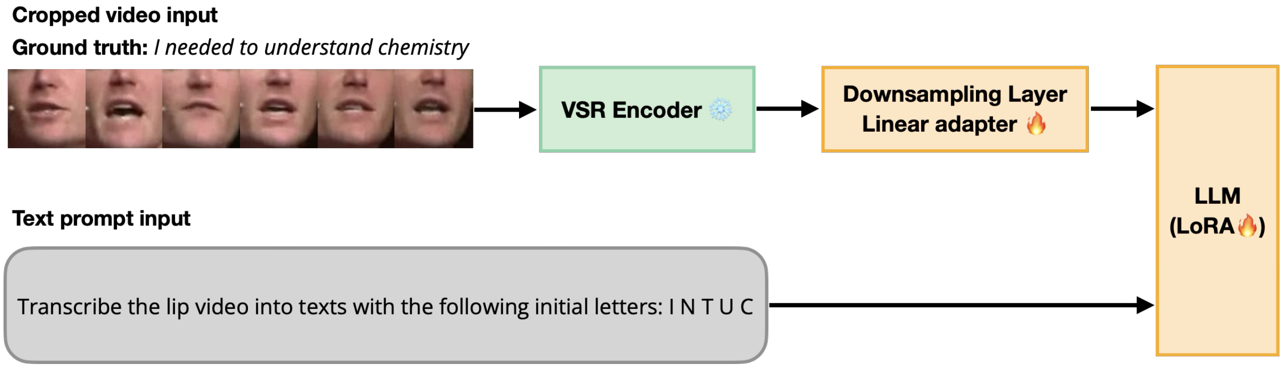

Cropped video input
Ground truth: I needed to understand chemistry
VSR Encoder
Downsampling Layer
Linear adapter
LLM
(LoRA)
Text prompt input
Transcribe the lip video into texts with the following initial letters: I N T U C

Figure 2: The architecture overview of the VSR-LLM integration conditioned by initial letters.

the LoRA adapter's rank and alpha to 128 with a dropout rate of 0.1, and use a hidden size of 2048 for the projector of the VSR encoder.

3.2 Lip-reading Datasets and Training Details

To train the proposed model, we use publicly available lip-reading datasets LRS2 [1] and LRS3 [2], which in total provide 818 hours of video data. Those datasets are extracted from the BBC TV program and TED talks and are widely used in VSR research for model training and benchmark purposes. Our preprocessing procedure splits long videos into clips with a maximum duration of 8 seconds to match the expected input scenarios, where the user speaks sentence by sentence so they can perform corrections if needed. We segment the lip region on the speaker's face using the Retinaface [9] landmark detector, following the same procedure in the Auto-AVSR paper. While the word initials are extracted from the transcriptions to facilitate subsequent data loading. Note that we excluded clips containing numbers that present as Arabic numerals, as they usually read as multiple words, making it difficult to accurately translate them into written-out forms in order to find the corresponding initial letters. Furthermore, the texts are formatted in lowercase, except for the word "I" and abbreviations containing it (e.g., "I'm," "I've," etc.). As a result, 31490 out of 478460 samples are screened out.

For training, we use both the pre-training and training subsets of LRS3, leaving the test set exclusively for evaluation. The model is trained for 30 epochs using the Adam optimizer [10], with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is 1e-3 and is reduced gradually by a CosineAnnealingLR scheduler [7]. Both the training and inference pipelines use the PyTorch and Huggingface's Transformers [53] frameworks.

3.3 Offline Model Evaluation

The test set of the LRS3 dataset was used to evaluate the model's performance and compare it with the Auto-AVSR model as a baseline. We also investigated how the initial prior, the size of the LLM, and different prompt alternatives would impact the model's performance. Table 1 shows the model performance in Character Error Rate (CER) and Word Error Rate (WER) measures, along with the accuracy of generating corresponding word initials.

3.3.1 Effect of Word Initials as Priors.

By adding word initials in the prompt as conditions, our model achieved a WER of 9.33%, which is a 54.04% relative improvement compared to the baseline model. This result suggests that combining an LLM decoder with VSR models can efficiently leverage the prior information from word initials to generate more accurate texts. Next, we conducted an ablation study by removing the other text prompts (i.e., the *transcribe the lip video to text with the initial letters:* part) and found that the WER dropped slightly to 9.47%. We still decide to keep those extra prompts for our prototype implementation because they are always fixed and only cause a small part of computational costs.

To investigate if the improvements come from the LLM alone, we then excluded all text prompts, including the word initials. In this case, the model only achieved a WER of 27.9%, confirming the effect of word initials as priors. We also noticed that, in this case, the performance is worse than the baseline model, although they share the same pretrained video encoder. However, the LLM decoder in our model is only fine-tuned on the LRS2 and LRS3 datasets, while the whole baseline model is trained with a much larger dataset with 5 times longer video hours. Unfortunately, since the authors did not open the additional dataset that they acquired by transcribing unannotated videos, we were unable to investigate how much our model would improve with more training data. Even so, the promising results with word initials indicate that our method is highly data-efficient and has huge room to improve as more data becomes available.

3.3.2 Effect of Model Size.

We build alternatives using LLaMA-3.2-3B-Instruct and LLaMA-3.2-1B-Instruct models [11]. The WER reached 9.33% for the 3B model and 15.5% for the 1B model, indicating that neural networks with more learnable parameters can provide better performance. Even larger models are not studied due to excessive training costs and latency in real-time inference scenarios. We also tested the inference time for each model size. On a desktop PC with an RTX 3090 GPU, the 3B model takes 367 ms to decode a 5s video (150 frames) using a beam size of 10, while the 1B takes 226 ms. Considering that the response times on mobile devices for visual stimuli range in 320 ± 43 ms [55], we believe using the 3B model can have little impact on the input speed while significantly reducing the errors.

Table 1: CER and WER performance of different models. The last two rows are ablation test results. *Ours − prompts* means no extra prompts were used except for word initials. *Ours − word intials* means the model only uses visual speech embeddings to decode.

Method	Pretrained LLM	CER(%)	WER(%)	Initials Accuracy (%)
Auto-AVSR [28]	-	14.52	20.3	-
Ours	LLaMA-3.2-1B-Instruct	10.1	15.5	75.3
Ours	LLaMA-3.2-3B-Instruct	**5.91**	**9.33**	99.3
Ours − prompts	LLaMA-3.2-3B-Instruct	6.07	9.47	**99.5**
Ours − word initials	LLaMA-3.2-3B-Instruct	18.6	27.9	50.4

3.3.3 Performance in Following Word Initials. To investigate how well the model learned to follow the word initials, we calculate the frequency of recognition results that strictly match the prompt. Without word initials, the model only achieves an accuracy of 50.4%. The 3B model achieves remarkable performance with an accuracy of 99.1%, while the 1B model falls short in this evaluation and only achieves an accuracy of 75.3%.

The results from the offline evaluation and analysis suggest that LLM can understand and follow the initials hint in a late fusion way, where the features from video and text are simply concatenated in the decoding stage. Since the 3B model has shown the best performance and works in nearly real-time, we choose to use this model to build a research prototype and investigate its usability in practice.

4 LipType: Multimodal Input Method using Silent Speech and Typing

LipType integrates the initials-conditioned VSR model in a mobile app to enable an efficient and fast text input experience. The following design factors are considered when developing our prototype.

4.1 Subvocalize-as-you-type

As shown in Figure 3, we use Swift and the Xcode IDE to implement a notepad-like app on iOS that also provides fundamental text entry and editing functions. When the user types the first character in a new sentence, the system is activated and starts to use the front camera to capture images of the user's face. After finishing both typing and speech input, the user presses the return key on the bottom right corner of the keyboard to query recognition results. Note that the speech and word initials input don't necessarily have to be strictly synchronized with each other, and our design is intended to exclude non-speech parts in the video for better efficiency and accuracy. The typed initial letters are shown in uppercase with spacing to facilitate verification. We use a socket connection protocol [16] to send the JPEG-compressed frames along with the word initials over local WiFi networks in real time. On a GPU server, the lip regions are extracted using the same Retinaface [9] detector and saved in a buffer for later inference.

4.2 Initials-constrained Beam Search

As discussed in Section 3.3.3, although our model has shown promising performance in decoding silent speech with word initials, there is still a small possibility that the generated text doesn't perfectly match the condition. Therefore, we develop an initials-constrained beam search algorithm by only allowing certain tokens at each decoding stop with a straightforward implementation: for each step t, the model can only generate a token $T \in A$, where it either adds a new word that fulfills the word-initials constrain (i.e., starts with a *space* character and followed by the next initial letter) or continues to be part of the previous word (i.e., starts with any other character except for *space*). We use regex representations to find the allowed token for each alphabet as the initial letter in advance to avoid additional computational costs during inference. During beam search, the probabilities of not allowed tokens are suppressed to $-inf$. In this way, the generated texts are always forced to follow the word initials, and we found that WER was reduced from 9.33% to 9.19% in the offline test. Although this improvement in recognition performance is marginal, it is important to eliminate unwanted behaviors for a reliable user experience. We empirically set the beam size to 10 and the maximum number of new tokens to generate to 20 by trading off the additional latency and accuracy.

4.3 Editing with Candidate Suggestions and Auto-correction

When recognition errors occur in the transcribed text, rather than deleting and retyping the entire word, the user can tap on the word to query for candidates and then select the appropriate replacement from the suggestion bar above the keyboard. During this process, the model preserves the existing visual and prompt embeddings and then performs a prefix-constrained beam search that leverages preceding words as prior context to improve accuracy. This method provides unique efficiency in LipType, because the word initials not only narrow the search space but also indicate the exact number of words in the sentence, ensuring a one-to-one correspondence between the recognized and intended words. Therefore, users can quickly spot errors and apply corrections. If the intended word is not among the model's suggestions, the user will need to use typing as a fallback.

Once corrections are completed, the model performs another prefix-constrained beam search to update the remaining words as ambiguity decreases. This automatic update is triggered either when a candidate is selected or when the user finishes typing a word (as indicated by pressing the space key).

5 User Study

We conducted a user study with text entry tasks to investigate the proposed method's performance in supporting efficient input on

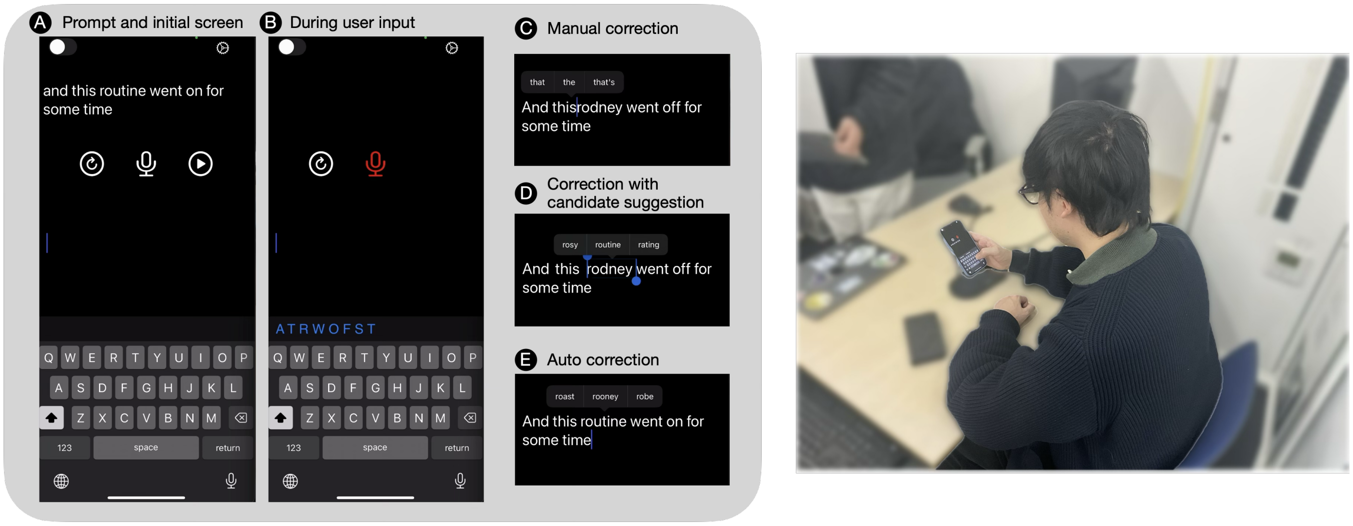

A Prompt and initial screen
B During user input
and this routine went on for some time
ATRWOFST
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return
C Manual correction
that the that's
And thisrodney went off for some time
D Correction with candidate suggestion
rosy routine rating
And this rodney went off for some time
E Auto correction
roast rooney robe
And this routine went on for some time

Figure 3: The prototype interface and the user study setup. (A) The prompt of phraseset and the initial screen before starting to input. There is a redo button, a microphone button that indicates the recording status, and a next button to proceed to the next phrase. (B) When the user starts to type, the prompt will be hidden, and the word initials will be shown in the toolbar at the top of the keyboard. The initial letters are editable during this process. The microphone button turns red to indicate that the speech is being recorded via the camera. (C) The user selects a misrecognized word to solicit candidates. If the desired word is not suggested, the they need to manually type it. (D) The user accepts the suggested candidate. (E) Following manual or candidate correction, the model will perform a new beam search with the confirmed prefix and update the remaining words.

smartphones. Furthermore, we expect that LipType can substantially save keystrokes, which can serve as an efficient input method even in one-handed situations. This feature makes the proposed interface not only friendly to handicapped people but also useful when the user is experiencing situational disabilities, e.g., when they are holding bags with one hand or cooking in the kitchen. Therefore, we evaluated LipType against traditional typing (also referred to as Type in this paper) in four text entry conditions: 1) Two-handed LipType, 2) Two-handed Type, 3) One-handed LipType, and 4) One-handed Type.

5.1 Apparatus and Participants

This study received approval from the university's Institutional Review Board (IRB). All participants provided informed consent by signing an IRB-approved consent form prior to their participation. Data was collected using an iPhone 15 Pro running iOS 18.2 with a 6.1-inch screen, a common size for modern smartphones. The device's camera was configured to capture images at a resolution of 1280 × 720 pixels and a frame rate of 30 fps. The deep learning backend was hosted on a desktop PC equipped with an NVIDIA RTX 3090 GPU, running Ubuntu 24.04. We recruited 16 participants (10 males and 6 females, aged between 22-30) from the local community through word-of-mouth. Of these, three were native or bilingual English speakers, while the remaining participants self-reported intermediate English proficiency. All participants indicated frequent experience with virtual keyboard typing on touchscreen smartphones for daily or work-related tasks.

We constructed our phrase set by randomly selecting utterances without replacement from the LRS3 test set. This dataset comprises TED talk transcripts, reflecting the vocabulary and structure of natural spoken communication. To approximate everyday text entry tasks such as SMS dictation, where speech input has demonstrated the best efficiency with a small amount of text to be entered [20], we further refined our study corpus to contain phrases with a constrained word count. For each session, we selected 6 phrases for each length ranging from 3 to 10 words, totaling up to $6 \times 8 = 48$ phrases. Texts are shown in lowercase, and most punctuations are removed except for the apostrophe mark, as we focus more on the word-level performance. We also As shown in Figure 3, the phrases are prompted on the top of the interface at the beginning of each trial. To simulate real scenarios where the participant types out the words in their mind, we hide the prompt as soon as the user presses the virtual keyboard. In this way, they had to remember the whole sentence before starting to type and could copy the word initials by watching the prompt. This is necessary as it ensures that the process of thinking of the initials is taken into account when measuring the input time.

There are three buttons in the middle of the text area for controlling the experiment process. The redo button resets the timer as well as the recorded frames, which is used when the participant fails to follow the rules or to say the phrase correctly. The microphone button turns red when the system is capturing frames for lip-reading to indicate the recording status. The next button submits the current result and moves to the next trial.

Finally, to counterbalance the experimental conditions and mitigate interaction effects, we employed a 2 (posture: two-handed vs. one-handed) × 2 (input method: Liptype vs. traditional typing) factorial design. Note that although there are four distinct conditions, we did not counterbalance them as a single set (i.e., $P(4,4)$); instead, we treated the two factors separately. With 16 participants, this resulted in 4 subjects per order (i.e., 16 ÷ 4 = 4).

5.2 Procedure

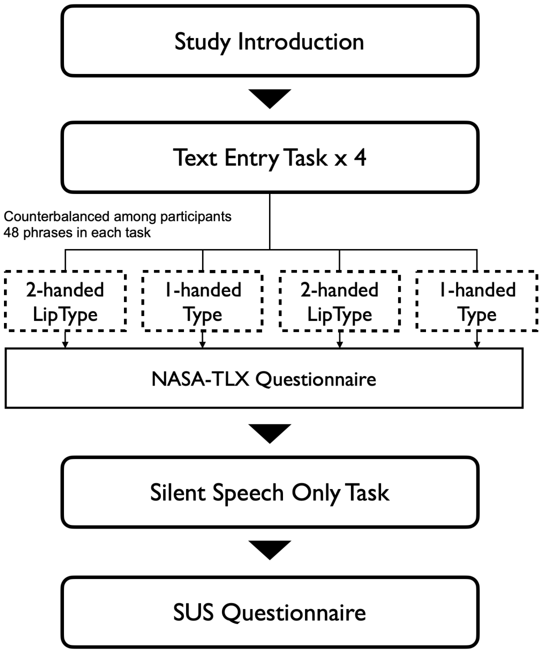


Figure 4: User study procedure shown in a diagram.

The whole user study was conducted in a lab environment with good light conditions. At the beginning of the study, we gave a brief introduction to the participants and collected consent forms. Next, the participant was asked to sit on a chair with armrests. Before each session, they had 5 minutes to learn to use the input method under the instructions from the experimenter. For practice purposes, we used a different corpus than the phrase sets mentioned before to avoid participants becoming familiar with or memorizing the stimulus that would be used in the test phase. Since the prompt was hidden during input, participants might struggle with remembering the phrase and fail to finish. Therefore, they were allowed to use the redo key to check the prompt again and start over. However, the prompt would not be hidden anymore if they failed three times in a row in one trial. The prompt would also be visible again for verification when the participant pressed the return key to indicate completion, but they could still correct errors, if any. Finally, we asked the participants to perform the task as fast and accurately as they could. Four sessions were carried out, and each corresponded to one of the input conditions. Note that for the one-handed sessions, participants only used their dominant hand (all participants were right-handed) to perform the task. However, if it was too difficult (fatigue, inability to reach certain keys, etc.) for them to continue, they were allowed to put the phone on a magnetic holder but still use only one hand to grasp and tap on the screen.

After each session, the participant was asked to complete a paired NASA TLX questionnaire for workload evaluation. Finally, a SUS usability test was conducted and subjective feedback was collected to find qualitative insights.

We were also curious whether the word initials input would slow down the user's speaking speed and further lead to a performance drop in our VSR-LLM model, especially compared to speech-only input. Therefore, we used another 48-phrase corpus, which was also randomly selected from the LRS3 test set following the same protocol, and asked the participant to silently articulate the phrases without typing. Since we only wanted to measure the speech WPM, neither recognition nor correction was performed in this session.

5.3 Quantitative Measures and Results

During the study, we used the following measures and logged the corresponding data for calculation. All results are reported in Table 2 and illustrated in Fig 5.

5.3.1 *Word Per Minute.*

WPM is a common measure of the input speed. We recorded the timestamp of the first key event as the start time and defined the finish time as the moment when the text was last modified. With those timestamps, we calculate the input time in seconds for each trial. Given that in HCI literature, the definition of a "word" in the context of WPM is five characters (including spaces) [31], the WPM is defined as

$$\text{WPM} = \frac{Number\ of\ Characters}{5} \times \frac{60}{Input\ Time}$$

First, we confirmed that the data within each factor combination could pass the Shapiro-Wilk normality test. We then performed two-way ANOVA and found a significant main effect for the Input Method factor ($F(1,60) = 14.07$, $p = .0004 < .001$, $\eta_p^2 = .19$) but no significant interaction effect $F(1,60) = 0.25$, $p = .621$, $\eta_p^2 = .004$. Given the higher average WER for LipType, we conclude that LipType results in faster input speed regardless of whether one or two hands are used. To further investigate the impact of LipType in each posture, independent t-tests were conducted separately, and significant differences were found for both two-handed ($t = 2.15$, $p = .039 < .05$, $d = 0.76$) and one-handed conditions ($t(15) = 3.23$, $p = .003 < .01$, $d = 1.14$). The effect of Input Method was even stronger for the latter condition, suggesting that LipType led to a larger performance gain when users typed with one hand. Furthermore, while one-handed typing is 13.8% slower than two-handed typing, this gap is only 6.98% for LipType.

5.3.2 *Keystrokes Per Character (KSPC).*

KSPC is an important indicator of the given text entry technique's efficiency, and it is defined as the number of keystrokes, on average, to generate each character of text [30]. Ideally, the KSPC of the mini QWERTY keyboard should be equal to 1 if we set aside the shift keys and other punctuations. In this study, we consider all screen touch events, including keyboard taps, cursor relocation, and candidate query/selection as keystrokes. Therefore, KSPC is defined as

$$\text{KSPC} = \frac{Number\ of\ Keystrokes}{Number\ of\ Characters}$$

A two-way ANOVA indicates that LipType required significantly fewer keystrokes per character than Type, where $F(1,60) = 584.22$,

Table 2: The quantitative results of WPM, KSPC, and Uncorrected WER. Numbers in the parentheses indicate the standard deviation.

	Two-handed		One-handed	
Input Method	LipType	Type	LipType	Type
WPM	42.96 (13.30)	33.98 (9.98)	40.23 (13.25)	28.52 (5.87)
KSPC	0.34 (0.07)	1.13 (0.09)	0.37 (0.12)	1.03 (0.17)
Uncorrected WER (%)	2.64 (2.20)	0.78 (0.13)	2.74 (2.98)	0.16 (0.34)

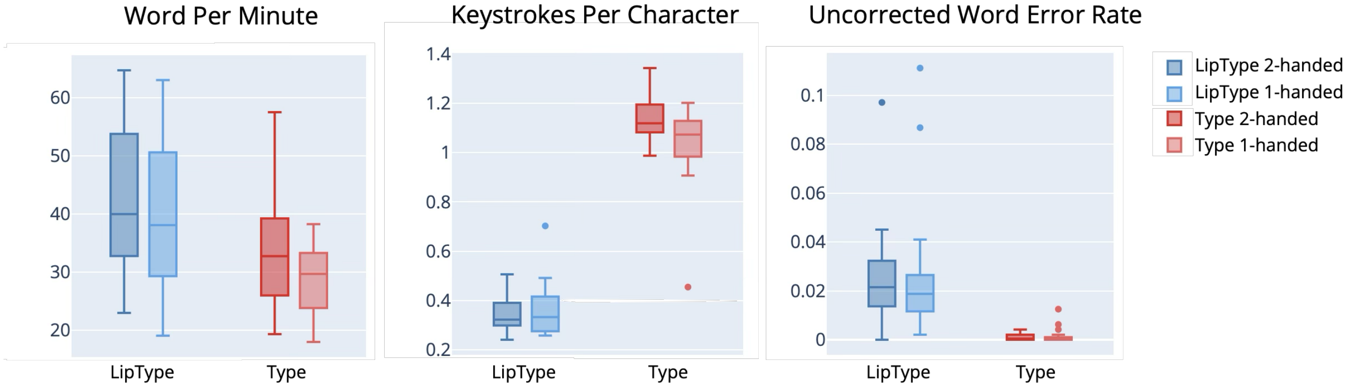


Figure 5: Box plots showing the distributions of the WPM, KSPC, and Uncorrected WER metrics. Circles represent outliers that fall below 3× IQR below Q1 or above Q3.

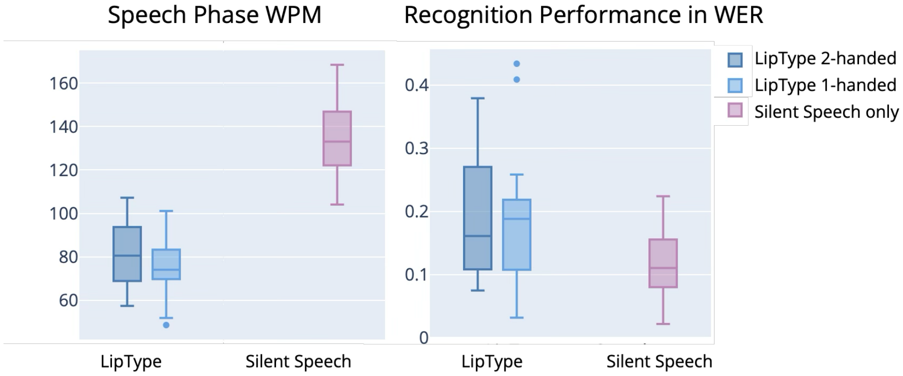


Figure 6: Box plots illustrating the speech input WPM and recognition performance of LipType and silent speech-only conditions. Circles represent outliers that fall below 3× IQR below Q1 or above Q3. The silent speech WER was computed in an offline manner after collecting data from the speech session.

$p = 1.27 \times 10^{-32} < .001$, $\eta_p^2 = .0.91$. Overall, LipType saved around 67.59% keystrokes across two postures. Since there are also significant interaction effects between Input Method and Posture ($F(1, 60) = 4.53$, $p = .037 < .05$, $\eta_p^2 = .07$), we performed a post-hoc pair-wise Tukey HSD to examine the differences among conditions. The result implies significant differences between LipType and Type for both two-handed and one-handed conditions ($p < .001$), confirming that our system consistently outperforms the traditional type-only input method in terms of efficiency, even in constrained scenarios.

5.3.3 *Uncorrected Word Error Rate.*

Although we asked the participants to make sure the input is accurate before proceeding to the next trial, they might fail to find all errors. Especially for LipType, we also logged the initial recognition results to compute the uncorrected WER along with the corrected WER. With S as the number of substitutions, D as the number of deletions, I as the number of insertions, and C as the number of correct words, WER can be calculated with the following equation:

$$\text{WER} = \frac{S + D + I}{S + D + C}$$

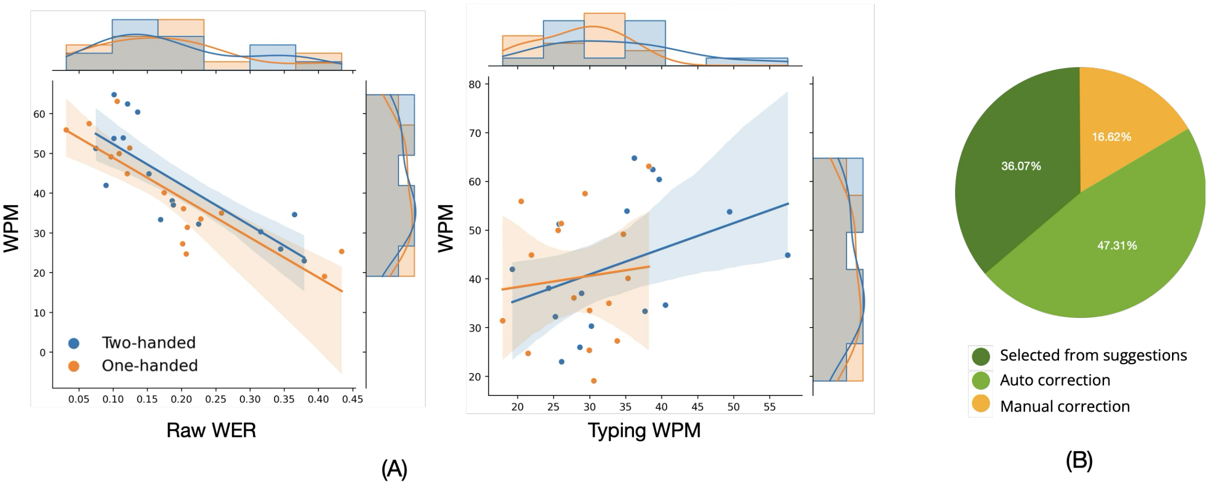


Figure 7: (A) Joint plots with marginal histograms between Raw WER vs. WPM and Typing WPM vs. WPM for LipType. The straight lines represent the linear regression functions with 95% confidence interval of the regression fit plotted in shadows. (B) The proportion of three types of corrections, where the error is 1) corrected by selecting from candidate suggestions, 2) automatically corrected when the user performed a suggested or manual correction, 3) manually corrected using the keyboard.

Via a 2-way ANOVA, we found that the uncorrected WER for LipType is significantly higher than Type with $F(1, 60) = 30.64$, $p < .001$, $\eta_p^2 = .34$. No significant interaction was found between Input Method and Posture ($F(1, 60) = 0.0003$, $p = .98$, $\eta_p^2 = 5.1\times10^{-6}$), suggesting that the effect of Input Method on uncorrected WER was consistent across both one-handed and two-handed conditions. Unlike conventional typing, where users naturally verify and correct each word during input, LipType requires users to review the entire sentence retrospectively, resembling the behavior of voice input interfaces. Previous research also revealed a fast memory decay of speech production [33], and consequently, users are more likely to overlook the errors in the dictated text with LipType, resulting in higher uncorrected WER. This limitation is not unique to LipType. Instead, similar results are reported in previous research comparing speech input and typing [39].

5.3.4 Speech Speed and Recognition Performance. To investigate whether typing the word initials and speaking at the same time will slow down the user's silent speech, we calculated the WPM during the speech input phase for LipType sessions and silent speech only sessions. I.e., the recognition time and correction time were excluded (Fig. 6). A one-way ANOVA test implies significant differences between two-handed LipType, one-handed LipType, and Silent Speech only ($F(2, 45) = 52.57$, $p = 2.8 \times 10^{-12} < .001$, $\eta^2 = .71$). A further Tukey's HSD post-hoc analysis shows that both two-handed LipType and one-handed LipType were significantly slower than the silent speech only condition ($p < .001$) but have no significance to each other ($p = .65$). This result also indicates that the word initials are easy to input even using only one hand.

Next, we ran an offline test to see if the slowed-down speech would result in degraded recognition performance for our model, where the raw LipType WER before correction is reported. As a result, the LipType WER was 18.89% ± 10.60%, higher than speech-only's 12.23% ± 5.65%. This is because the dataset that is used to train our model consists of TV program videos, where the speaker usually talks at a higher speed, and their face was captured from a fixed camera at a distance. As a text entry method, we can easily infer the ground truth as the user uses LipType, thereby enabling model personalization to fill the gap between the distributions of train and test data.

5.3.5 Effect of Recognition Accuracy and Typing Performance. We investigate how the recognition accuracy of our model and the user's typing performance would impact the user's input speed. Fig 7 (A) shows the distribution between each of the two factors. We calculate the Pearson Correlation Coefficient and found strong negative correlation between Raw WER (the raw output from our VSR-LLM model before corrections) and WPM for both two-handed ($r = -0.799$, $p < 0.001$) and one-handed conditions ($r = -0.845$, $p < 0.001$). This implies that the input speed can highly rely on the recognition accuracy of silent speech, as lower WER results in fewer correction operations. On the other hand, the user's conventional typing skills (represented by their WPM in Type sessions) show a moderate positive correlation to their input speed with LipType in two-handed conditions ($r = -0.396$, $p = 0.129 > 0.05$) and a very weak positive correlation in one-handed conditions ($r = -0.102$, $p = 0.707 > 0.05$) with no significance found. Therefore, we infer that the user's performance with LipType relies less on their typing skills. With LipType, even people who have limited typing experience on mobile devices can achieve comparable input speed to experienced typing users.

5.3.6 Proportions of Correction Types. In Fig 7 (B), we illustrate the proportions of each correction type, namely those that were 1) corrected by selecting from candidate suggestions, 2) automatically corrected when the user performed a suggested or manual correction, 3) manually corrected using the keyboard. Among all 1763

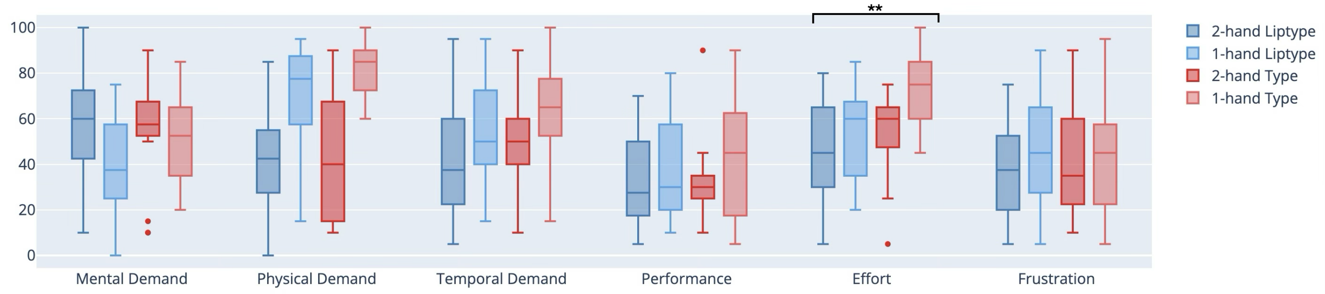


Figure 8: Distribution of NASA-TLX workload ratings across different tasks, visualized using box plots. Lower scores represent lower perceived workload. Asterisk markers indicate the significance level (: $p < 0.01$).**

corrections performed by 16 participants, auto-corrections had the highest ratio (47.31%), while only 16.62% of the errors needed to be manually corrected. This promising result shows that the word initials not only helped the initial decoding of silent speech, but also largely narrowed down the candidate scope during the later editing process, ensuring efficiency by saving both input time and keystrokes.

5.4 NASA TLX Ratings

In Figure 8, we summarize the participants' subjective workload ratings for each task, where lower scores indicate less perceived workload and demand. Across both two-handed and one-handed conditions, LipType consistently received better (lower) ratings than Type for Physical Demand, Temporal Demand, and Effort. Two-way ANOVAs revealed a significant main effect of input method on Effort, with LipType rated as less effortful ($M = 49.22, SD = 22.58$) than Type ($M = 62.97, SD = 19.67$), $F(1, 60) = 7.34, p = .008 < .01, \eta_p^2 = .11$. The interaction effect for Effort was not significant ($F(1, 60) = 1.37, p = .25 > .05, \eta_p^2 = .02$), indicating this effect was consistent across typing postures. Although the main effect of input method on Physical Demand did not reach statistical significance ($F(1, 60) = 1.79, p = .185, \eta_p^2 = .03$), LipType was rated as less physically demanding on average ($M = 54.53, SD = 26.98$) compared to Type ($M = 61.72, SD = 28.78$). Notably, the relatively higher F-value for Physical Demand compared to other non-significant workload dimensions suggests a trend in this direction, although the dominant factor for Physical Demand was the typing posture, as indicated by a significant main effect ($F(1, 60) = 44.02$, $p = 1.05 \times 10^{-8} < .001, \eta_p^2 = .42$). No significant main effects of input method were found for Mental Demand ($M_{LipType} = 47.81$, $SD = 23.89$; $M_{Type} = 52.81$, $SD = 21.51$), Temporal Demand ($M_{LipType} = 49.53$, $SD = 26.86$; $M_{Type} = 56.56$, $SD = 22.98$), Performance ($M_{LipType} = 35.16$, $SD = 21.35$; $M_{Type} = 38.44$, $SD = 23.12$), or Frustration ($M_{LipType} = 40.94$, $SD = 23.12$; $M_{Type} = 42.34$, $SD = 24.82$). Despite the lack of statistical significance for these dimensions, the consistently lower means for LipType across all workload categories suggest a tendency towards a reduced overall workload compared to Type.

5.5 System Usability Scale Scores and Subjective Feedback

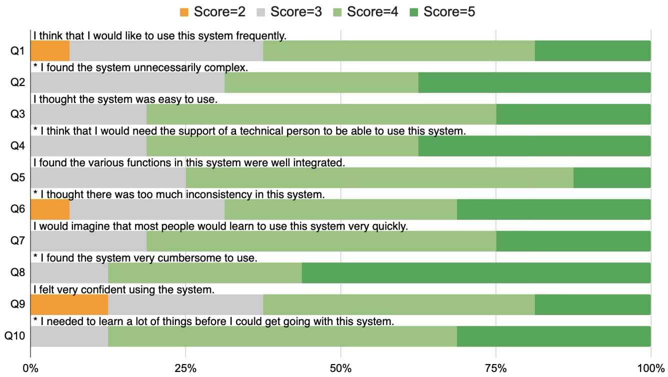


Figure 9: The overview of the SUS results. For better visualization, we reversed the score for negatively worded questions (Q2, Q4, Q6, Q8, and Q10, indicated with asterisk markers).

The user study concluded with a SUS test, and we noted down if the participant had any additional feedback. As shown in Fig 9, the result suggests generally positive feedback on usability with an average score of 72.625 (SD=9.54), which means it is considered highly usable and easy to learn. P8 expressed, *"I really want to use this system (LipType) when I'm commuting on the subway."* P10 felt that their English language proficiency was limited and had a negative impact on the performance: *"I can speak and type much faster if I use my native language (Chinese)."* On the other hand, the model achieved the best raw WER on P1, who is a native English speaker, and they were very confident using the system and said that *"I would like to use it as my default app for texting."* We also received concerns about privacy issues when the participant learned how the system works. *"I don't want to turn on the front camera in public* (P5)." However, after explaining that only the mouth area of the video was used for recognition, they changed their perspective but insisted: *"I want to make sure it doesn't upload my data to some remote cloud servers."*

All participants fully understood how to use the system after the experimenter's demonstration and a short period of practice. P14 said, *"I found this interaction very natural to me because I usually read aloud in my mind when typing."* However, some other participants reported that they felt their speaking speed was slowed down when trying to recall and type the initial letters. We believe that although LipType, as a novel text entry method, definitely introduces additional learning costs, it aligns with human nature and can serve as a day-to-day technology.

6 Discussion, Limitations, and Future Work

We discuss insights from the system development and user study, outline the limitations of current implementation, and shed new light on the directions for future work.

6.1 The Future of Silent Speech for Text Entry

Despite silent speech interfaces have been studied for years, text-entry applications are still challenging due to limited recognition performance. Lip-reading based methods are promising because of rich training data; however, the best available model produces an order of magnitude more errors compared to the best ASR model [38]. The inherent difficulty of lip-reading stems from the ambiguity of visemes and the manifold data distribution resulting from different camera angles. Therefore, we consider the possibility that eventually, lip-reading models will face a lower upper limit of recognition performance, making it not feasible to build a general model that can work out-of-the-box for everyone and achieve sufficient recognition performance as a standalone input method.

In this work, we take a novel approach to integrate additional input channels, namely initial typing, to exploit silent speech despite its ambiguity. LipType not only provides a practical method to enable private and efficient text input with current lip-reading models, but it also has the potential to bring personalized models into reality without placing an extra burden on users. Text entry tasks that happen during daily activities, such as messaging, document writing, and note-taking, are usually double-checked by the user. The ground truth can be easily acquired by assuming the final results are mostly correct. In this way, we no longer need a dedicated data collection process, which can be tedious and time-consuming, as it is possible to collect sufficient data during day-to-day use and build a model that is specialized for a single person. As the recognition performance improves over time, eventually, word initials can be omitted when the model has a high confidence and only required in a post-hoc manner. We envision this human-in-the-loop process is the key to democratizing silent speech without pain.

6.2 Recognition with Fuzzy Word Initials and Numbers

One major limitation of our user study is that we asked the participants to enter the word initials perfectly, and they had to redo the trial if errors were found. Especially for longer sentences, it can be difficult to always make sure the initial letters are correct. To achieve a better user experience, we need to enable the model to tolerate such noise to mitigate friction. We set off by case-studying the data from the user study to find frequent errors in the word-initial input. As shown in Table 3, we show examples of Substitution, Delete, and Insert errors when the user fails to type the right initials (those are not final results as the user re-tried after an error was detected). Typical substitution cases seem due to the unpronounced letter in the word (e.g., the letter "K" in "know") or the consonant at the end of the word. Delete errors share a less common pattern, simply caused by missing one or more words in the sentence. Most insert errors appear when the user accidentally typed out the abbreviated word (letter "A" in "they're" and letter "I" in "there's"). We attempt to cope with this problem by synthesizing fuzzy word initials in the training data.

Specifically, with a probability of 0.25, we randomly remove a letter, substitute a letter with its neighboring letter, or duplicate a letter in the ground truth. The new model was trained following the same procedure as described in 3.2, and we performed two tests on the LRS3 test set. For the first run, there was no alternation for the initial letters, and for the second run, at least one letter for each sentence was altered using the same data augmentation. Results show that the model achieved a WER of 13.83% on altered initial letters while keeping the same level of 9.43%. This promising result suggests that the model can be trained to tackle fuzzy initials with little harm to normal input. We envision that by expanding the training data corpus, it is possible to enable flexible input, where only a small part of word initials as hints will provide sufficient accuracy. The user only needs to disambiguate words that have the same visemes, e.g., indicate the letter "b" for the word "bark" to avoid "park", and leave alone the rest of the words.

Furthermore, in this work, we excluded Arabic numerals from both the training dataset and the corpus used in the user study. However, numbers originally written out in words were retained, allowing the model to still recognize and process numerical information to some extent, and the user study results still capture the effect of numerical text entry. Nonetheless, it is important to acknowledge that the current model is trained on limited numerical data and is unable to generate Arabic numerals. In future work, we plan to leverage advanced methods such as LLMs to accurately convert numbers into written English words. This approach will enable the extraction of corresponding word initials, which expands the available training data and holds the potential to support Arabic numeral text input.

6.3 Longitudinal and In-the-wild Study

The text entry task highly relies on the user's level of experience. Therefore, to fully understand the implications of multimodal silent speech and typing input, a long-term out-of-the-lab study is important. Most people have become familiar with conventional typing on mini QWERTY keyboards due to the popularization of smartphones. However, the user study in this work only demonstrated our prototype's usability in the hands-on stage. Future work demands a thorough study on the learning curve and potential risks of prolonged use, which should be done by tracking the user's performance on a weekly scale. One of the biggest barriers is that the current model is still heavy and can only run on high-end GPU machines, which are only available in locations with reliable local network connections. With the advancement of efficient LLMs for edge devices, we plan to optimize our model for on-device inference

Table 3: Typical errors in word-initial inputs.

Sentence	User input	Correct Initials	Error Type
I don't know	IDN	IDK	Substitution
Thank you very much	TKVM	TYVM	Substitution
She found us we found her disease	SFUFHD	SFUWFHD	Delete
They had her scanned inside out	THHSI	THHSIO	Delete
They're not my children	TANMC	TNMC	Insert
There's no structure	TINS	TNS	Insert

using techniques such as quantization [14] and knowledge distillation. Following the validation of models suitable for deployment on resource-constrained devices, it will be crucial to conduct an in-the-wild study to gain insights into performance and usability in real-world scenarios.

We used the most commonly used datasets, LRS2 and LRS3, to train the embedding projection layers and LoRA adapter in our model. They are at scale and contain data from diverse speakers and backgrounds; however, TED talks are well-rehearsed, presented in front of an audience, and timed, thus not fully capturing the nature of spontaneous social communication. Note that although we also used the LRS3 corpus in the user study, video data was recorded on the phone during the user's typing and should provide a closer approximation to real-world scenarios. We consider the domain gap between the training data and actual use cases of our system to be a principal factor contributing to the observed performance decline when comparing the results from the offline experiments with those from the user study. In addition, varying lighting conditions or blurry videos are not studied in this work. Future work should also focus on building dedicated visual speech datasets to improve the model's generalizability and establishing reliable benchmarks for text entry tasks.

7 Conclusion

In this paper, we introduced a novel multimodal text entry system that integrates silent speech recognition with word-initial input on mobile devices. By leveraging a word-initials-conditioned LLM fused with a visual speech encoder, our approach significantly improves silent speech recognition accuracy while ensuring efficient and private text entry. Through extensive offline evaluations, we demonstrated that our method achieves a remarkable Word Error Rate (WER) reduction from 20.14% to 9.19%, outperforming state-of-the-art VSR models.

We further developed a functional mobile prototype of LipType to investigate its usability in text entry tasks. Our user study revealed that LipType enables significantly faster input speed compared to conventional typing on virtual keyboards, especially in one-handed use cases where it achieved an average WPM of 41.80, a 35.32% improvement over traditional typing. Additionally, LipType achieved an average keystroke savings of 67.59% while maintaining a manageable error rate. Participants reported lower physical demand and effort when using LipType, confirming the system's efficiency and potential for daily text entry tasks. Moreover, the successful integration of word initials in the input process demonstrated its effectiveness in disambiguating lip-reading outputs, further validating our design choices.

Despite its advantages, LipType's limitations include the requirement for precise word-initial input, which could be mitigated by enabling tolerance for minor input mistakes. Additionally, real-world deployment necessitates model optimization for on-device inference to ensure lower latency and enhanced privacy. Future work will explore long-term user adaptation, personalized model fine-tuning, and cross-linguistic applications to expand LipType's usability.

In conclusion, this work presents a promising step toward practical silent speech-based text entry systems. By combining efficient multimodal input with LLMs, our approach enhances both speed and usability while maintaining privacy, paving the way for future silent speech interaction on mobile devices.

Acknowledgments

This work was supported by JST Moonshot R&D Grant Number JPMJMS2012 and JSPS KAKENHI Grant Number 24KJ0775. We would further like to thank the participants for their involvement in our user study and the anonymous reviewers for their valuable and constructive feedback.

references

[1] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2018. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence* 44, 12 (2018), 8717–8727.

[2] 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).

[3] Alan D Baddeley, Susan E Gathercole, and Costanza Papagno. 2017. The phonological loop as a language learning device. *Exploring working memory* (2017), 164–198.

[4] Tom Bellman and I Scott MacKenzie. 1998. A probabilistic character layout strategy for mobile text entry. In *Graphics Interface*, Vol. 98. 168–176.

[5] Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge, and Shumin Zhai. 2012. Bimanual gesture keyboard. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 137–146.

[6] Vladimir V Bochkarev, Anna V Shevlyakova, and Valery D Solovyev. 2015. The average word length dynamics as an indicator of cultural changes in society. *Social Evolution and History* 14, 2 (2015), 153–175.

[7] PyTorch Contributors. 2024. CosineAnnealingLR. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingLR.html. Accessed: 2025-02-27.

[8] Wenzhe Cui, Suwen Zhu, Mingrui Ray Zhang, H Andrew Schwartz, Jacob O Wobbrock, and Xiaojun Bi. 2020. Justcorrect: Intelligent post hoc text correction techniques on smartphones. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 487–499.

[9] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. 2020. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5203–5212.

[10] P Kingma Diederik. 2014. Adam: A method for stochastic optimization. *(No Title)* (2014).

[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

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

[13] Cathy Mengying Fang, Phoebe Chua, Samantha Chan, Joanne Leong, Andria Bao, and Pattie Maes. 2024. Leveraging AI-Generated Emotional Self-Voice to Nudge People towards their Ideal Selves. *arXiv preprint arXiv:2409.11531* (2024).

[14] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323* (2022).

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

[16] Robbie Hanson. n.d.. CocoaAsyncSocket. https://github.com/robbiehanson/CocoaAsyncSocket. Accessed: February 5, 2025.

[17] Ramin Hedeshy, Chandan Kumar, Raphael Menges, and Steffen Staab. 2021. Hummer: Text Entry by Gaze and Hum. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 741, 11 pages. https://doi.org/10.1145/3411764.3445501

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

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

[20] Razan Jaber and Donald McMillan. 2020. Conversational user interfaces on mobile devices: Survey. In *Proceedings of the 2nd Conference on Conversational User Interfaces*. 1–11.

[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] 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 *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19.

[23] 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.

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

[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*. 6866–6873.

[26] Chandan Kumar, Ramin Hedeshy, I Scott MacKenzie, and Steffen Staab. 2020. Tagswipe: Touch assisted gaze swipe for text entry. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–12.

[27] Tianshi Li, Philip Quinn, and Shumin Zhai. 2023. C-PAK: correcting and completing variable-length prefix-based abbreviated keystrokes. *ACM Transactions on Computer-Human Interaction* 30, 1 (2023), 1–35.

[28] 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.

[29] Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, et al. 2024. An Embarrassingly Simple Approach for LLM with Strong ASR Capacity. *arXiv preprint arXiv:2402.08846* (2024).

[30] I Scott MacKenzie. 2002. KSPC (keystrokes per character) as a characteristic of text entry techniques. In *International Conference on Mobile Human-Computer Interaction*. Springer, 195–210.

[31] I Scott MacKenzie and R William Soukoreff. 2002. Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction* 17, 2-3 (2002), 147–198.

[32] I Scott MacKenzie and Shawn X Zhang. 1999. The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 25–31.

[33] Brinda Mehra, Kejia Shen, Hen Chen Yen, and Can Liu. 2023. Gist and Verbatim: Understanding Speech to Inform New Interfaces for Verbal Text Composition. In *Proceedings of the 5th International Conference on Conversational User Interfaces*. 1–11.

[34] Sadia Nowrin and Keith Vertanen. 2023. Programming by Voice: Exploring User Preferences and Speaking Styles. In *Proceedings of the 5th International Conference on Conversational User Interfaces*. 1–13.

[35] Tom Ouyang, David Rybach, Françoise Beaufays, and Michael Riley. 2017. Mobile keyboard input decoding with finite-state transducers. *arXiv preprint arXiv:1704.03987* (2017).

[36] Akhil Padmanabha, Jessie Yuan, Janavi Gupta, Zulekha Karachiwalla, Carmel Majidi, Henny Admoni, and Zackory Erickson. 2024. Voicepilot: Harnessing LLMs as speech interfaces for physically assistive robots. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–18.

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

[38] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*. PMLR, 28492–28518.

[39] Sherry Ruan, Jacob O Wobbrock, Kenny Liou, Andrew Ng, and James Landay. 2016. Speech is 3x faster than typing for english and mandarin text entry on mobile devices. *arXiv preprint arXiv:1608.07323* (2016).

[40] Paul W Schönle, Klaus Gräbe, Peter Wenig, Jörg Höhne, Jörg Schrader, and Bastian Conrad. 1987. Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract. *Brain and Language* 31, 1 (1987), 26–35.

[41] Khe Chai Sim. 2010. Haptic voice recognition: Augmenting speech modality with touch events for efficient speech recognition. In *2010 IEEE spoken language technology workshop*. IEEE, 73–78.

[42] Khe Chai Sim. 2012. Speak-as-you-swipe (SAYS) a multimodal interface combining speech and gesture keyboard synchronously for continuous mobile text entry. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. 555–560.

[43] Zixiong Su, Shitao Fang, and Jun Rekimoto. 2022. LipLearner: Customizing silent speech commands from voice input using one-shot lipreading. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–3.

[44] 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

[45] Zixiong Su, Xinlei Zhang, Naoki Kimura, and Jun Rekimoto. 2021. Gaze+ Lip: rapid, precise and expressive interactions combining gaze input and silent speech commands for hands-free smart TV control. In *ACM symposium on eye tracking research and applications*. 1–6.

[46] Bernhard Suhm, Brad Myers, and Alex Waibel. 2001. Multimodal error correction for speech user interfaces. *ACM transactions on computer-human interaction (TOCHI)* 8, 1 (2001), 60–98.

[47] 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.

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

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

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[51] Jingxian Wang, Chengfeng Pan, Haojian Jin, Vaibhav Singh, Yash Jain, Jason I Hong, Carmel Majidi, and Swarun Kumar. 2019. RFID tattoo: A wireless platform for speech recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–24.

[52] Xue Wang, Zixiong Su, Jun Rekimoto, and Yang Zhang. 2024. Watch Your Mouth: Silent Speech Recognition with Depth Sensing. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–15.

[53] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[54] Zheer Xu, Shanqing Cai, Mukund Varma T, Subhashini Venugopalan, and Shumin Zhai. 2024. SkipWriter: LLM-Powered Abbreviated Writing on Tablets. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[55] Kyle T Yoshida, Joel X Kiernan, Allison M Okamura, and Cara M Nunez. 2023. Exploring human response times to combinations of audio, haptic, and visual stimuli from a mobile device. In *2023 IEEE World Haptics Conference (WHC)*. IEEE, 121–127.

[56] Shumin Zhai, Michael Hunter, and Barton A Smith. 2002. Performance optimization of virtual keyboards. *Human–Computer Interaction* 17, 2-3 (2002), 229–269.

[57] Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 97–104.

[58] Mingrui Ray Zhang, He Wen, and Jacob O Wobbrock. 2019. Type, then correct: Intelligent text correction techniques for mobile text entry using neural networks. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 843–855.

[59] Mingrui Ray Zhang and Shumin Zhai. 2021. PhraseFlow: Designs and empirical studies of phrase-level input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[60] 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.

[61] Maozheng Zhao, Henry Huang, Zhi Li, Rui Liu, Wenzhe Cui, Kajal Toshniwal, Ananya Goel, Andrew Wang, Xia Zhao, Sina Rashidian, et al. 2022. Eyesaycorrect: Eye gaze and voice based hands-free text correction for mobile devices. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*. 470–482.