STAR: Smartphone-analogous Typing in Augmented Reality

Taejun Kim*
Realiy Labs Research, Meta
Toronto, ON, Canada
taejun.kim@kaist.ac.kr

Amy Karlson
Reality Labs Research, Meta
Redmond, WA, USA
akkarlson@meta.com

Aakar Gupta
Reality Labs Research, Meta
Redmond, WA, USA
aakar.hci@gmail.com

Tovi Grossman
University of Toronto
Toronto, ON, Canada
tovi@dgp.toronto.edu

Jason Wu†
Reality Labs Research, Meta
Toronto, ON, Canada
jsonwu@cmu.edu

Parastoo Abtahi
Reality Labs Research, Meta
Toronto, ON, Canada
parastoo@princeton.edu

Christopher Collins
Reality Labs Research, Meta
Toronto, ON, Canada
chriscollins@meta.com

Michael Glueck
Reality Labs Research, Meta
Toronto, ON, Canada
mglueck@meta.com

Hemant Bhaskar Surale
Reality Labs Research, Meta
Toronto, ON, Canada
hemantsurale@meta.com
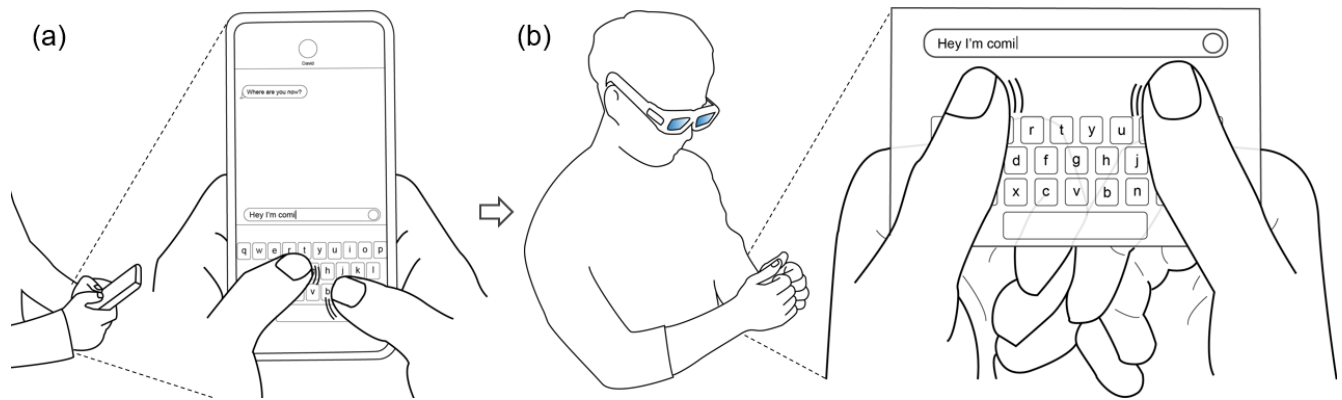
Figure 1: (a) Traditional physical smartphone typing and (b) STAR, a bare-hand, two-thumb text entry method in augmented reality. STAR transfers the two-thumb typing skills learned from using a physical smartphone to bare-hand AR typing.

ABSTRACT

While text entry is an essential and frequent task in Augmented Reality (AR) applications, devising an efficient and easy-to-use text entry method for AR remains an open challenge. This research presents STAR, a smartphone-analogous AR text entry technique that leverages a user's familiarity with smartphone two-thumb typing. With STAR, a user performs thumb typing on a virtual QWERTY keyboard that is overlain on the skin of their hands. During an evaluation study of STAR, participants achieved a mean typing speed of 21.9 WPM (i.e., 56% of their smartphone typing speed), and a mean error rate of 0.3% after 30 minutes of practice. We further analyze the major factors implicated in the performance gap between STAR and smartphone typing, and discuss ways this gap could be narrowed.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**.

KEYWORDS

Text Entry, Augmented Reality, Smartphone Typing, Head-Mounted Display, QWERTY Keyboard

---

*The author is also affiliated with HCI Lab, KAIST, Daejeon, Republic of Korea. This work was done while the author was a Research Intern at Meta.

†The author is also affiliated with Carnegie Mellon University, Pittsburgh, PA, USA. This work was done while the author was a Research Intern at Meta.

---

1 INTRODUCTION

Recent advancements in Augmented Reality (AR) hardware and display technologies (e.g., glasses [24, 46, 58]) have opened up new opportunities for utilizing AR in various fields such as remote work, health care, and education [6, 52, 63]. However, despite the growing need for text entry in AR environments, devising an efficient and easy-to-use text entry solution remains an open challenge.

While researchers have developed several techniques to enable bare-hand text entry for unrestrained AR activities, prior bare-hand AR text entry techniques such as mid-air hand typing [19, 45, 72], eye typing [38, 43], or tapping on a fingertip [69, 70] are unfamiliar to users as they are drastically different from today's typing methods. In addition, common bare-hand AR text entry techniques often lack the haptic feedback delivered when one touches a physical surface. For example, the Microsoft Hololens 2 AR text entry method [19] requires users to touch a mid-air virtual keyboard using their fingers, without any haptic feedback. This lack of surface haptics has been found to impact input accuracy, typing speeds, and user fatigue [2, 13, 28, 32].

Herein, we present STAR, a novel bare-hand AR text entry technique that capitalizes on users' familiarity with two-thumb typing (i.e., both hands holding a smartphone while both thumbs are typing). Due to the ubiquity of smartphones, two-thumb typing is a familiar and rapid form of input for most people [47, 57, 65], even approaching the performance of desktop QWERTY keyboard typing for some users [21, 47]. Our work explores how this familiar skill could be transferred to a bare-hand AR text entry. While it may not be possible to match the typing performance of smartphones with a bare-hand technique, we believe the performance gap between the two can be closed by leveraging the same skills of two-thumb typing.

To use STAR, one forms a "knuckle posture" with their hands (Figure 1b) as if holding a smartphone, which triggers the display of a mini virtual QWERTY keyboard on the user's hand through the Head-Mounted Display (HMD) they are wearing. STAR then enables the user to leverage the haptic feedback of touching their own skin to two-thumb type on this keyboard. If the user wishes to transition onto other tasks, they can seamlessly release the knuckle posture and the keyboard vanishes. The knuckle posture is an explicit mode-switching technique that prevents false activations. Further, STAR is expected to be more socially acceptable compared to other bare-hand text entry techniques because the smartphone-typing gesture naturally conveys the user's state of typing to others [20, 68].

In an empirical evaluation, STAR showed an efficient text entry performance with a mean typing speed of 21.9 words per minute (WPM) and a mean error rate of 0.3%. This is 56% of smartphones typing speed, after only 30 minutes of practice. We further discuss ways to narrow the performance gap between STAR and physical smartphone typing based on the collected typing data and subjective user feedback.

The main contributions are as follows:

- We present STAR, a novel on-skin, smartphone-analogous AR text entry technique that leverages existing typing skills from a physical smartphone.
- We explore the transferability of two-thumb typing skills to AR. In an elicitation study with 29 participants, three different hand postures were discovered for typing on an imaginary smartphone on their skin. These postures, along with other design parameters, were then tested to ensure sufficient skill transfer from smartphone typing to STAR.
- An analysis of the performance delta between STAR and smartphone typing reveals that improved hand tracking and thumb tap sensing may provide an opportunity for STAR to achieve typing performance closer to that of a physical smartphone.

2 RELATED WORK

We first review the literature on large-scale datasets that have been used to characterize smartphone typing skills. Subsequently, we discuss on-body interactions and skill transfer. Lastly, we scrutinize text entry techniques proposed for HMD-based AR and position STAR among them.

2.1 Smartphone Typing Skills

Due to the ubiquity of smartphones, smartphone typing has become a familiar and rapid form of input for most users [47, 57, 65]. Common smartphone text entry techniques involve the coordinated tapping of user's thumbs on a miniature QWERTY keyboard.

Palin et al. [47] presented a large-scale dataset of mobile text entry input collected from a web-based transcription task with 37,370 volunteers (mean age: 24.1). The mean typing speed was 36.2 WPM with 2.3% uncorrected errors and the fastest typists reached over 80 WPM, approaching the performance levels of desktop QWERTY keyboard typing, which has been found to have a mean typing speed of 51.6 WPM [17] and the fastest speed up to 130 WPM [49]. Over 82% of participants used two thumbs to type, which was significantly faster than using one finger (i.e., 37.7 vs. 29.2 WPM). The mean typing speed was similar between genders (i.e., ~36.1 WPM for both men and women) but differed between age groups (i.e., 39.6 WPM for 10-19 years old vs. 26.3 WPM for 50-59 years old). Although it may not be possible to match the same typing performance with bare-hand techniques, STAR aims to narrow the gap by leveraging the pervasive skills of smartphone two-thumb typing.

2.2 On-Body Interactions and Skill Transfer

Using the body surface as an input space has been a long-standing interaction paradigm. For example, Harrison et al. [27] presented Skinput, an interaction using skin as an always-available input surface. Imaginary Phone [26] proposed using the skin on one's hand as a touch input surface and suggested "transfer learning", i.e., by using a physical smartphone, a user inadvertently learns the interface and can then transfer that knowledge to an imaginary interface on the surface of their hand. Gustafson et al. [26] noted that *"the transfer model is viable, even though full accuracy will not be redeemed until higher resolution tracking equipment becomes available"*. STAR leverages this same concept by supporting the transfer of learning from physical smartphone typing to virtual bare-hand typing.

2.3 Text Entry in HMD-based AR

The proposed text entry techniques for HMDs were often motivated by AR and VR applications, with some techniques [19, 38, 60] being

Technique	Modality	Typing Level		Speed (WPM)	Error (%)	Training Amount	Traits
		letter	word				
Controller Pointing [60]	Handheld controllers	✓		15.4	1.0	5 minutes	Need to hold controllers
Word-Gesture [12]			✓	16.4	15.6^C	None	
EyeSwipe [38]	Head + Eye	✓	✓	11.7	1.3	30 minutes	Impacts natural viewing, induce eye fatigue
iText [43]			✓	13.8	1.5^W	72 phrases (4 days)	
GlanceWriter [15]		✓	✓	10.9	2.7^W	10 minutes	
Speech Recognition [53]	Speech		✓	179	4.4	10 phrases	Raise privacy concern
Vulture [45]	Mid-air hand movement		✓	28.1	1.7^W	48 phrases	Induce arm fatigue, lack haptic feedback
ATK [72]			✓	29.2	0.4^W	5 minutes + 45 phrases	
FastType [61]		✓		22.3^P	2.3	15 minutes	
VISAR [19]		✓	✓	17.8	0.6	80 phrases	
ThumbAir [23]			✓	13.7	1.2	140 words + 35 phrases	
QwertyRing [25]	Finger taps on physical surface		✓	20.6	1.3	120 phrases (4 days)	Rely on probabilistic decoder
TapType [62]		✓	✓	$19.2/9.0^L$	$0.6/0.4^L$	35 phrases	
TypeAnywhere [62]			✓	70.6	1.5	120 m. + 80 ph. (4 days)	
PalmType [64]	On-skin finger touch	✓		7.7	1.6	(not specified)	Require users to learn new interaction paradigms, use custom hardware
DigiTouch [66]		✓	✓	16.0	0.9	180 minutes	
TipText [70]			✓	13.3	0.3	30 phrases	
BiTipText [69]			✓	25	0.03	30 phrases	
STAR	**On-skin thumb taps**	✓	✓	**21.9**	**0.3**	**30 minutes**	**Leverages familiar smartphone typing skills**

Table 1: Overview of text entry techniques proposed for, or potentially usable in, HMD-based AR. The table compares these techniques along the following dimensions: modality, supported typing level (i.e., letter-level or word-level), mean typing speed, mean error rate, amount of training needed, and traits of the technique. Error indicates character-level uncorrected error rates unless specified (C denotes character-level corrected error rates, and W denotes word-level uncorrected error rates). P denotes peak typing speed that was measured by a repeated typing of the same word. L denotes mean typing speed with test phrases including Out-Of-Vocabulary (OOV) words, requiring letter-by-letter typing. It is important to note that the performance difference between each method may also be influenced by additional factors such as the input technology (i.e., sensing accuracy), the specific set of test phrases, the probabilistic decoder implemented, or the keyboard layout used.

usable in both domains. Herein, we review text entry techniques proposed for, or potentially usable in, HMD-based AR systems which leverage input modalities such as controller, gaze, speech, mid-air hand movements, finger tapping on physical surfaces, or on-skin touches (Table 1).

2.3.1 Handheld Controllers. Text entry techniques using handheld controllers [7, 12, 60] have been widely adopted in consumer VR applications. Research has shown that controller-based text entry is often fast and easy to learn, with users reaching a mean typing speed of 15.4 WPM after only 5 minutes of practice [60]. While techniques with controllers can work well for VR applications that support a defined activity in a fixed space, the constant need to hold controllers and their limited battery life prevents widespread use in more ubiquitous AR situations.

2.3.2 Eye Gaze. Modern consumer HMDs (e.g., Hololens 2 and Meta Quest Pro) that support eye tracking presents subtle input using gaze-based techniques [15, 22, 38, 43]. EyeSwipe [38], for example, enabled gaze-only gesture-based typing with a mean typing speed of 11.7 WPM and a mean error rate of ~1% after 30 minutes of practice. Recently, Lu et al. [43] proposed using head rotation for

pointing and eye blinking for selection, achieving a mean typing speed of 13.8 WPM and a mean error rate of ~1.5% after 72 phrases of practice over four days. Although text entry with eye and head movements supports subtle interaction, such techniques impact natural viewing and suffer from eye fatigue [5, 14, 31, 39].

2.3.3 Speech. Text entry using speech recognition via the built-in microphones within HMDs has been considered an efficient method for text input [8, 51, 53, 55]. Ruan et al. [53] showed that a deep learning-based speech recognition system could achieve a mean text entry speed of 179 WPM and a mean error rate of ~4.4%. However, speech recognition is known to be unstable in noisy environments [51], and the use of speech raises privacy and social acceptability concerns [55].

2.3.4 Mid-Air Hand Movement. Several mid-air hand typing techniques have been proposed for unrestrained bare-hand AR activities. For example, Vulture [45] enabled mid-air gesture swipe typing using a finger pinch that was tracked with an Optitrack motion tracking system. Although only word-level input was supported with this technique (i.e., letter-by-letter typing was not supported), it yielded a mean typing speed of 28.1 WPM with a mean error

rate of ~2% after practice with 48 phrases. ATK [72] used a Leap Motion sensor to enable mid-air ten-finger typing with a probabilistic tap detection algorithm for each finger. Likewise, although only word-level typing was supported, ATK was found to have a mean typing speed of 29.2 WPM and a mean error rate of 0.4% after practice with 45+ phrases. ThumbAir [23] typing using in-air two-thumb movements has recently shown a mean typing speed of 13.7 WPM with a mean error rate of 1.2% after practicing 140 words and 35 phrases. Dudley et al. [19] proposed VISAR, which used direct finger touch on a mid-air virtual keyboard for both letter- and word-level typing. VISAR was found to have a mean typing speed of 17.8 WPM with a mean error rate of ~0.6% after practice with 80 phrases. It is currently the default text entry method in Microsoft Hololens 2. While these results are impressive, typing in mid-air lacks the haptic feedback that is generated when one touches a physical surface. The lack of surface haptics has been shown to induce significant arm and hand fatigue [2, 13, 28, 32]. In addition, mid-air gestures typically require users to keep their hands raised at eye level, resulting in arm weariness and potential social acceptability concerns.

2.3.5 Finger Taps on Physical Surfaces. There has been active research to enable text entry on any physical surfaces around us, such as tables. Researchers have proposed the use of wearable Inertial Measurement Unit (IMU) devices, such as a ring [25], two wristbands [62], or two five-finger-straps [76] to decode typing sequences during finger tap typing on an imaginary QWERTY keyboard. An example of such an approach is TypeAnywhere [76], which achieved a mean typing speed of 70.6 WPM with a mean error rate of 1.5% after four days of practice. Although these outcomes are remarkable, these methods are mainly suited for word-level input due to their reliance on probablistic decoding at that level. For example, TapType [62] evaluated typing performance using test phrases that include Out-Of-Vocabulary (OOV) words by treating each character as a "word", resulting in a considerable decrease in typing speed from 19.2 to 9.0 WPM.

2.3.6 On-Skin Finger Touch. Researchers have suggested using the skin as an always-available typing surface. Typing on the skin with bare-hand techniques also allows users to utilize their dexterous hand skills in an AR interaction. PalmType used the palm and fingers as input space and mapped each key of the QWERTY layout on separate segments of the skin [64]. While eyes-free typing with an index finger, PalmType achieved a mean typing speed of 7.7 WPM. DigiTouch used the skin on one's finger for bimanual thumb-to-finger touch interaction (i.e., the thumb touched the skin of the fingers on the same hand), achieving a mean typing speed of 16.0 WPM and a mean error rate of ~1% after 180 minutes of practice [66]. TipText [70] and BiTipText [69] used conductive films attached at the first segment of the index finger to support eyes-free text entry (i.e., the thumb tip tapped on the film). Both techniques were limited to entering text only at the word level and did not offer the capability to input characters individually. The unimanual TipText achieved a mean typing speed of 13.3 WPM and a mean error rate of ~0.3%, whereas the bimanual BiTipText achieved a mean typing speed of 25 WPM and a mean error rate of ~0.03%. Both of these results were after practice with 30 phrases.
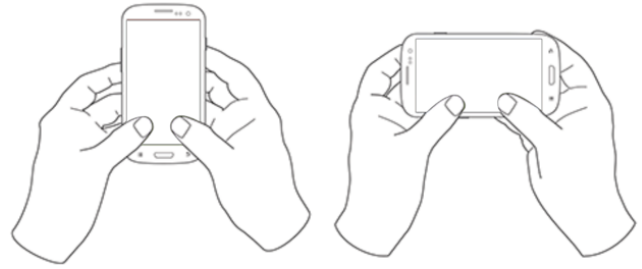
Figure 2: Hand postures during two-handed physical smartphone use that were observed during Hoober's studies (image from [30], with permission).

These techniques drastically differ from conventional typing methods, requiring users to learn new interaction paradigms often with custom hardware. STAR, on the other hand, offers a familiar typing experience to users by leveraging their physical smartphone two-thumb typing skills. Furthermore, STAR supports both word-level and letter-level input, offering practicality in real-life typing scenarios.

3 STAR DESIGN PROCESS

To design a smartphone-analogous typing method, we first surveyed how users typically hold and type on a physical smartphone. According to Palin et al.'s large-scale dataset on mobile text entry [47], over 82% of people used two thumbs to type. In addition, a public observation on how people hold mobile devices [29] found that people "cradled" their mobile phone in their fingers and used both thumbs for two-handed input (Figure 2). Within this context, we established the following **design principles**: 1) users should hold their hands in a similar manner to how they do while holding a mobile phone, 2) the same QWERTY layout should be used, 3) the finger movements used to type should be similar to smartphone typing, and 4) there should be a physical surface to type on as if typing on the smartphone.

Following these principles, during our design phase, we considered cognitive factors, including the intuitiveness of hand posture and physical factors, including the typing surface and keyboard layout and size.

3.1 Elicitation Study: Typing Hand Posture

To explore the appropriate typing skin surface and corresponding hand posture for STAR, we conducted an elicitation study to observe the hand postures that users naturally used while typing on an imaginary smartphone. The elicitation study was run remotely through video calls with 29 participants (19 females, 10 males; age: M = 38 years, SD = 14 years). Participants were instructed to imagine that there was an invisible, imaginary smartphone in their hands to type on. Afterward, they were asked to type the sentence, *"The quick brown fox jumps over the lazy dog"* on an imaginary smartphone in their hands. We observed hand postures from different angles. The study took about 15 minutes, including the introductory instructions and the completion of a demographic survey. All participants were paid.

(a) *Single Palm Surface*
[n=2] (b) *Nested Finger Surface*
[n=20] (c) *Index Finger Surface*
[n=6]
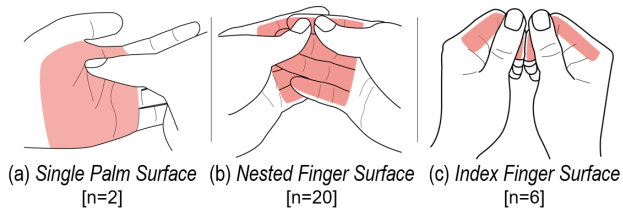
Figure 3: Results of the elicitation study depicting the skin surfaces and hand postures that users would use for imaginary smartphone typing. These are (a) Single Palm Surface with one hand and tapping with a finger from the other hand (n = 2), (b) Nested Finger Surface with both hands folded and tapping with both thumbs (n = 20) and (c) Index Finger Surface with the hands in a symmetric contact and tapping with both thumbs (n = 6). One participant used both thumbs while separating their hands, which did not fit into any category.

During the study, 93% of participants (27 out of 29) used both thumbs for typing, and 96% of them (26 out of 27) maintained contact between both hands while typing. This observation provided evidence that most users perceived typical smartphone typing as utilizing both thumbs while keeping the hands in contact. This is notable considering that users' hands may not always make contact during the use of a physical smartphone (Figure 2).

Many participants (n = 20) used the *Nested Finger Surface* to create a thumb tapping space by placing their hands together in a folded position, with one hand on top of the other hand (Figure 3b). The next most common approach (n = 6) was to utilize the *Index Finger Surface* by making symmetric contact between the hands. In summary, the elicitation study revealed three main candidates for on-skin smartphone-analogous typing. As this research focuses on two-thumb typing, cases involving two thumbs (Figures 3b and 3c) were explored further.

3.2 Prototype Development and Feedback

During the iterative design of the technique, we focused on improving the accuracy, comfort, and efficiency by refining the typing surface, keyboard position, and keyboard size.

3.2.1 Typing Surface. Based on the findings from the elicitation study, we singled out two candidates for on-skin two-thumb typing: the *Nested Finger Surface* and the *Index Finger Surface*. To examine the suitability of each typing surface in realizing STAR, we developed an initial prototype with a Hololens 2 AR HMD. The prototype utilized the HMD's vision-based hand tracking technology for thumb tracking and tap sensing. In the testing setup, a virtual keyboard of smartphone size (Figure 4c, *Original*) was placed in the world space and displayed through the HMD. The participant then manually aligned their hand surface with the keyboard. The tracked position of the thumb's tip, which interacts with the keyboard, was visualized using a small colored sphere.

During an informal pilot, we observed that users experienced difficulty in achieving stable key tapping on the virtual keyboard that is overlain on the *Nested Finger Surface*. This was due to the uneven typing surface caused by the depth difference between the

two hands (i.e., one hand was placed behind the other). Moreover, the middle finger was often positioned slightly behind the index finger, adding uneven depth distance from each thumb. On the other hand, we observed that they were able to make more consistent key presses with better stability on the virtual keyboard that is overlain on the *Index Finger Surface*. This could be attributed to the fact that the distance from the thumb to the surface was more even due to the flat property. Based on the pilot's feedback, we decided to use the *Index Finger Surface* for on-skin thumb typing. Although the *Index Finger Surface* was not the most popular choice from the elicitation study, we were encouraged that multiple participants (n = 6) independently suggested this posture as a canonical representation of imaginary smartphone typing, which also promoted more symmetric and ergonomic thumb-driven typing than the other option.

3.2.2 Keyboard Positioning. When using a physical smartphone, the touchscreen follows a user's hand position as they are holding the device. We tried to mimic this experience by incorporating a *Hands-Following* keyboard (Figure 4b), which updates its position and orientation according to the joints' tracked positions. To compensate for the hand tracking jitter (Figure 5b), a low-pass filter (i.e., 1-euro filter [9]) was applied, and the axis of rotation was restricted for stable alignment between the keyboard plane and the user's *Index Finger Surface*. During an informal pilot, however, we observed that users had difficulty tapping the intended keys as the hand tracking jitter was still causing the keyboard to shake. This was particularly noticeable when the two thumbs were continuously moving to type. Although a strong low-pass filter could alleviate this, it would also cause a delay in achieving timely synchronization between the virtual keyboard and the user's hand surface. We observed that even a tiny amount of unexpected displacement of the keyboard could significantly elevate users' tapping errors, when dealing with the small inter-key distances (1-2 mm). Therefore, we decided to try a *Stationary* keyboard by fixing the keyboard at the initial position where the knuckle posture was made. With the *Stationary* keyboard, a user can reposition the virtual keyboard by making a knuckle posture at a new location.

During an informal test run, we observed that users were able to type more confidently using both thumbs on the *Stationary* keyboard that is overlaid on the *Index Finger Surface*, without worrying about the keyboard unexpectedly moving out of place. Finally, we decided to use the *Stationary* positioning in our final implementation to simulate a stable keyboard alignment.

3.2.3 Keyboard Size. Lastly, we tested different keyboard layouts of varying sizes, starting with the layout of a physical smartphone keyboard (Figure 4c, *Original*). During an informal assessment where users typed phrases they wanted on a *Stationary* virtual keyboard overlain on the *Index Finger Surface*, we observed that users frequently made tapping errors. This is anticipated, given that the keys were 5.5 mm wide with an inter-key distance of 1 mm, and the Hololens' finger tracking showed a tracking jitter of around ± 1 mm (Figure 5b). We then designed larger keyboards that would facilitate more confident thumb tapping while making sure that keys on both ends (e.g., q or p) were within an easily reachable distance. Through an informal test, we examined various key widths (i.e., 6, 7, 8 mm) and inter-key distances (i.e., 2, 3, 4 mm), and decided to use a 6 mm
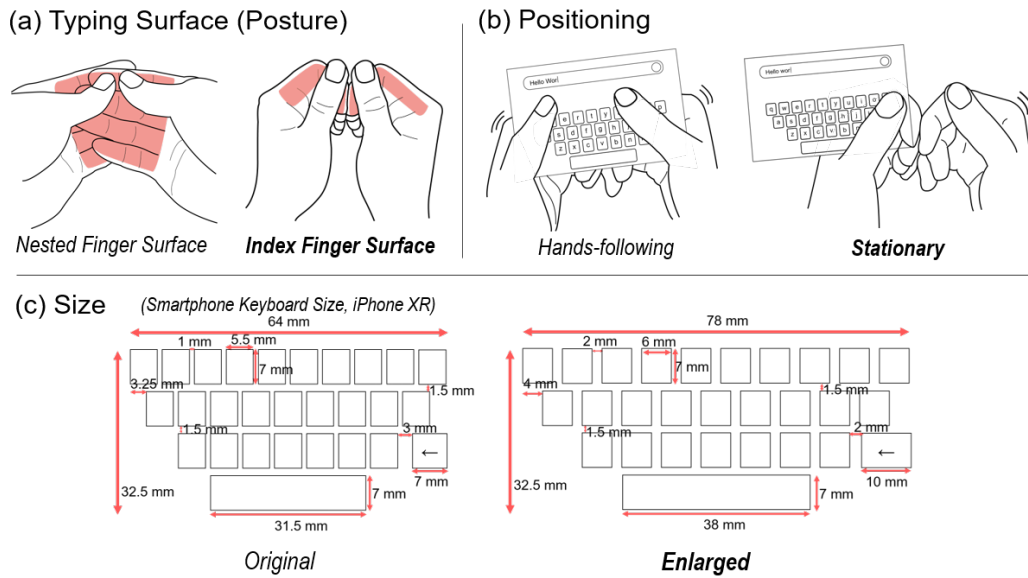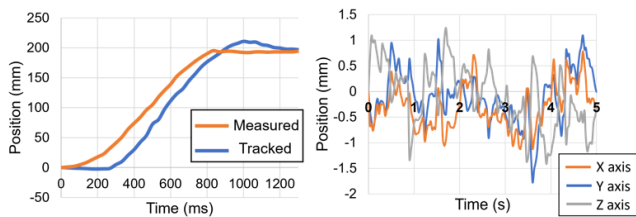
Figure 4: The three design parameters explored during the design phase for STAR: (a) the use of hand surfaces for two-thumb typing such as the *Index Finger Surface* and *Nested Finger Surface,* (b) different Positionings such as *Hands-Following* and *Stationary,* and (c) different Sizes such as *Original* (i.e., the iPhone XR's default keyboard layout) and *Enlarged.* Boldfaced options were chosen for the final STAR design.

key width and 2 mm inter-key distance for the *Enlarged* keyboard size (Figure 4c) in our final implementation. Although we standardized the keyboard size for the controlled experiment, personalizing the keyboard size based on the user's hand size remains a valuable opportunity.

(a) Tracking latency while moving   (b) Jittering of tracked position while not moving

Figure 5: The observed (a) tracking latency and (b) tracking jitter of the thumb tip position from the Hololens 2 hand tracking. The tracking latency was measured while moving the hand approximately 200 mm along a single axis, whereas the measured position was approximated from the recorded video. (i.e., A ruler was put at a fixed position in the recorded video, and the tracked thumb was moved in line with the ruler.) The tracking jitter was measured for 5 seconds while the hand was held stationary on an armrest.

4 STAR IMPLEMENTATION

Based on the explored design parameters, we arrived at a set of design decisions that optimizes usability and key input accuracy: *Enlarged* size, *Index Finger Surface* typing surface, and *Stationary* positioning as highlighted in bold texts in Figure 4. In this section, we describe the main components of our implementation: 1) hand tracking, 2) thumb tap sensing for key registration, 3) visualization details, and 4) word suggestion feature.

In the final implementation (Figure 6), a user can invoke STAR by making a "knuckle posture" as if holding a smartphone. The knuckle posture invokes the virtual keyboard and overlays it on the hands. The user can then perform two-thumb typing on the sides of the index fingers. Once the text entry is complete, the user can exit the typing mode by releasing the knuckle posture to seamlessly transition to other tasks. The knuckle posture is expected to support quick mode switching while preventing unintended activation since the sensory feedback of the hand contact signals the mode change [56].

4.1 Bare-Hand Position Tracking

We implemented STAR under the hand tracking capabilities of the current state-of-the-art AR HMD (i.e., Hololens 2). As modern AR/VR HMDs on the consumer market (e.g., Hololens 2, Meta Quest 2, and HTC Vive) support hand tracking for bare-hand interaction, the tracking fidelity is expected to be more mature over time. However, it currently has a noticeable tracking latency of ~90 ms and a tracking jitter with a range of approximately ±1 mm, as shown in Figure 5. The selection of design parameters, such as *Stationary* positioning, was made to simulate more stable hand

(a) Making knuckle posture

(b) Performing two-thumb typing

(c) Releasing knuckle posture

(d)
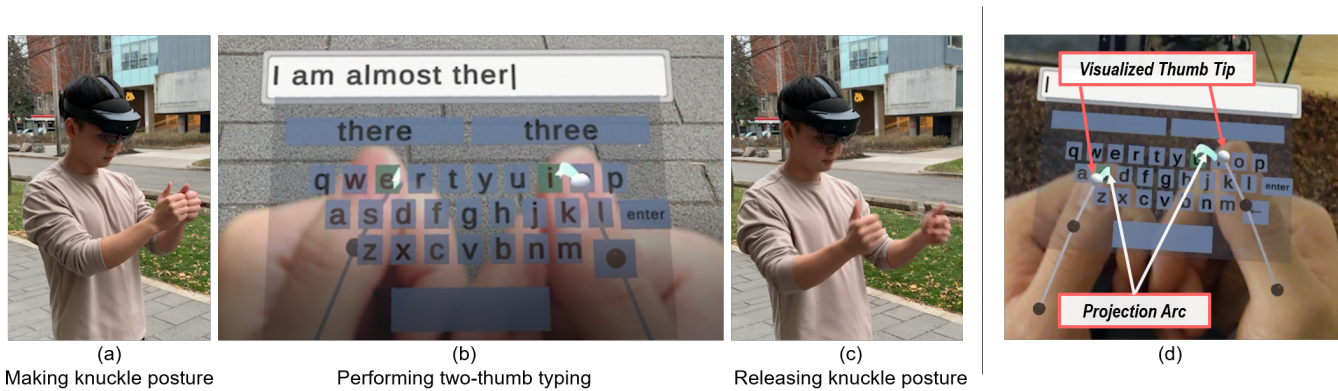
Figure 6: A typical interaction sequence while using STAR. (a) A user triggers STAR typing by making a knuckle posture. (b) The user performs familiar two-thumb typing on their *Index Finger Surface* with visualized thumbs and projection arcs. (c) Finally, the user exits the typing mode by releasing the knuckle posture. (d) The clear view of the projection arc and visualized thumb joints.

tracking under the capabilities of the current state-of-the-art AR HMD (i.e., Hololens 2). Thus, there may be a chance to revisit the design decisions with a more reliable tracking environment.

4.2 Thumb Tap Sensing (Key Registration)

Just like on a physical smartphone, a key tap should be detected whenever the thumb touches the *Index Finger Surface*. We first implemented the detection of key tap (i.e., the contact between a thumb and the *Index Finger Surface*) solely with vision-based hand tracking, as shown in Figure 6. However, because the Hololens 2's finger tracking did not have sub-millimeter level precision, the key registration and thumb-finger contact was often temporally misaligned. This misalignment of haptic feedback led to errors and user complaints. To address this in the study, we developed robust thumb tap sensing beyond what the Hololens 2 could sense. We used thumb-tip-worn capacitive tapes (Figure 7a) to simulate accurate tap detection. Thumb taps register the key with the closest center on the virtual keyboard. Given the continued advancement in hand tracking technology, we anticipate that fully bare-hand thumb tap sensing will achieve comparable performance levels. Further discussion on the realization of bare-handed tracking is in Section 6.1.

4.3 Visualization Details

In see-through (i.e., glasses-based) augmented reality, where virtual pixels are overlain on the physical environment, users often confuse the "depth" of a virtual object [50]. We also observed that users often misjudged the distance between their hands and the virtual keyboard, which resulted in improper hand alignment. To alleviate the depth perception, we visualized the projection arc from the thumb tip to the keyboard plane, and highlighted the hovered key at the end of the arc using color feedback (Figure 6d). In addition, we visualized all three thumb joints and the connections between them above the keyboard, to provide users with a better perception of thumb depth.

During an informal test, we found that users frequently made tapping mistakes when only the *ThumbTip* joint was visualized. This was attributed to the thumb position shifting during curved tapping movements. For example, users first located their thumb straight above an intended key (e.g., k) and executed a thumb tap, resulting in them touching an adjacent key (e.g., j) due to the curved trajectory caused by the thumb's joint flexion (i.e., the human thumb tip does not move in a straight line, but rather an arc [71]). After visualizing all three thumb joints and the expected projection arcs, we observed that users better predicted the key to be selected by their curved thumb tapping movements.

4.4 Word Suggestion Feature

The completion of a word via predictive suggestion is a common-place feature within today's mobile text entry systems. There are three common types of suggestions: auto-completion/correction when the Space is pressed, suggestions that are tapped before one types all the letters in a word, and suggestions that are tapped to make a correction after typing all the letters in a word. Although widely used, forcing auto-completion/correction on every Space could interfere with the evaluation of the technique itself. Therefore, we included the second feature, which allowed participants to use completion by tapping suggested words based on their needs. The words with the highest and second highest probability were suggested on the left and right buttons, respectively.

To identify the two most probable words, a statistical decoding system combined spatial probability and language model probability. First, for each touch point on the keyboard, the spatial probability of each letter was calculated using a bivariate Gaussian Distribution [4] (e.g., t: 0.544, y: 0.432, ..., p: 0.001; the σ value was the distance between the center of each key). By multiplying the spatial probabilities of each touch sample, the probabilities of the possible sequence of characters could be calculated (e.g., thw: 0.425, the: 0.312, thr: 0.266, ..., pmz: 0.000). The decoder then generated a list of words that begin with each candidate sequence of characters (as a prefix) in the language corpus, and multiplied the language

(a) STAR Typing Test (b) Smartphone Typing Test (c) Schematic Diagram
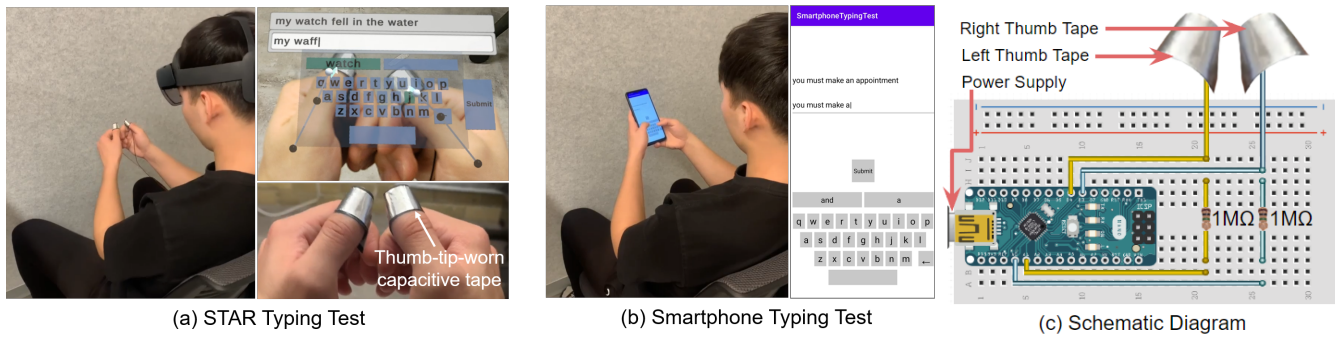
Figure 7: The testing environments used during the evaluation study. (a) In the STAR typing test, the participant put on the AR HMD (Hololens 2) and the thumb-tip-worn capacitive tapes. (b) In the smartphone typing test, the participant held a physical smartphone to type. (c) Schematic diagram includes an Arduino board, two 1MΩ resistors, and capacitive tapes for thumbs.

model probability to calculate the final probability of the word. For the language corpus, Kaufman's lexicon [33] was used, with each word's frequency extracted from Wikipedia corpora [41].

5 STAR EVALUATION

We conducted a user study to evaluate STAR's text entry performance. We also included a physical smartphone typing condition (*Smartphone*) to observe the performance gap between *STAR* and *Smartphone*. As the goal of the study was not to outperform *Smartphone* but to determine how close *STAR* could approach state-of-the-art smartphone typing, *Smartphone* was evaluated before and after the study was completed (i.e., it was not a typical within-subject factor). As the text entry speeds on the two instances of *Smartphone* typing were not significantly different (39.9 vs. 39.0 WPM), they were averaged to form baseline data.

5.1 Participants

Ten participants (i.e., two females, eight males; age: M = 25 years, SD = 4 years) were recruited from a university community to participate in the study. Two participants wore glasses, two participants were left-handed, and eight participants had no prior AR experience. All participants were paid.

5.2 Apparatus

A Microsoft Hololens 2, with a diagonal field of view of 52°, was used to run the Unity-based *STAR* application. For thumb tap sensing, real-time capacitance data from the thumb-tip-worn tape was sent to a PC from an Arduino board through serial communication and was then sent to the Hololens 2 device through wireless TCP communication (Figure 7c). We used distinct thresholding for debouncing the signal. The system identifies the tap engagement when the capacitance surpasses 250 (unit returned from CapacitiveSensor library [1]), and it recognizes the completion of the tap when it falls below 200 units. For the *Smartphone* baseline, a Galaxy A13 5G smartphone was used. The *Smartphone* test application was developed in Android Studio and used the *Original* keyboard layout (Figure 7b).

5.3 Procedure

At the beginning of the study, participants were asked to sit on a chair with no armrest. After completing a consent form, participants were instructed on how to complete the experiment using a video and slides. Participants were asked to roll up their sleeves so they would not cover their wrists and face a wall with a plain background to allow for stable hand tracking. Participants donned the thumb-tip-worn capacitive tape before starting the *STAR* blocks.

On each trial, participants were asked to transcribe a phrase that was randomly generated from the Mackenzie and Soukoreff phrase set [44]. Phrases that contained words that were not in Kaufman's lexicon [33] were excluded. The set of phrases presented was identical for all participants. To transcribe a phrase, participants were instructed to make a "knuckle posture" to open the keyboard and then tap their "*Index Finger Surface*" with a thumb to touch a key. Similar to many other evaluations of text input techniques [19, 47], users could opt to complete a word by tapping on word suggestions, consisting of two options in our design (Figure 7). After each transcription trial was complete, participants clicked on the *Submit* button. They could then check their text entry speed and error rate for that trial.

Each block in the study contained ten transcription trials. Participants completed one block of *Smartphone* before completing the five *STAR* blocks, and then a final block of *Smartphone* after completing the *STAR* blocks. Prior to the first *STAR* block, participants were asked to transcribe three sample phrases to familiarize themselves with the *STAR* application. At the beginning of each block, a preparation stage allowed participants to touch keys on an empty text field without knowing the target phrase to allow them to make minor adjustments to the height/distance of their hands to the keyboard if desired. Once a participant was ready to start transcribing, they pressed the *Show Phrase* key (the *Submit* button shown in Figure 7a was *Show Phrase* before starting a trial) to get the target phrase. Participants were informed that their typing speed would be recorded from the moment they pressed their first letter, so they could memorize the target phrase before starting the transcription if desired.

Participants were instructed to type as quickly and accurately as possible. Between each block, participants removed the HMD

	Block 1	Block 2	Block 3	Block 4	Block 5	Smartphone
Text Entry Speed (WPM)	17.4 (3.1)	18.5 (2.6)	19.9 (2.2)	21.8 (2.8)	21.9 (2.8)	39.4 (7.4)
UER (%)	0.1 (0.2)	0.2 (0.3)	0.3 (0.4)	0.3 (0.7)	0.3 (0.6)	0.5 (0.7)
CER (%)	7.5 (4.8)	9.2 (6.9)	9.0 (5.6)	7.9 (6.2)	8.8 (4.2)	8.3 (3.4)
IKI (ms)	805 (203)	697 (154)	676 (175)	617 (128)	585 (100)	315 (72)
Backspace Usage (count)	3.0 (2.0)	2.9 (2.4)	3.2 (2.3)	2.9 (2.4)	3.0 (1.4)	2.6 (1.2)
Key Press Duration (ms)	148 (25)	138 (22)	140 (21)	138 (20)	132 (17)	84 (9)

Table 2: The means and standard deviations of the text entry speeds, uncorrected error rates (UER), corrected error rates (CER), inter-key intervals (IKI), backspace usage counts, and key press durations during the user study. The performance of each block is reported separately for the *STAR* condition, and the average of the two blocks is reported for the *Smartphone* condition.
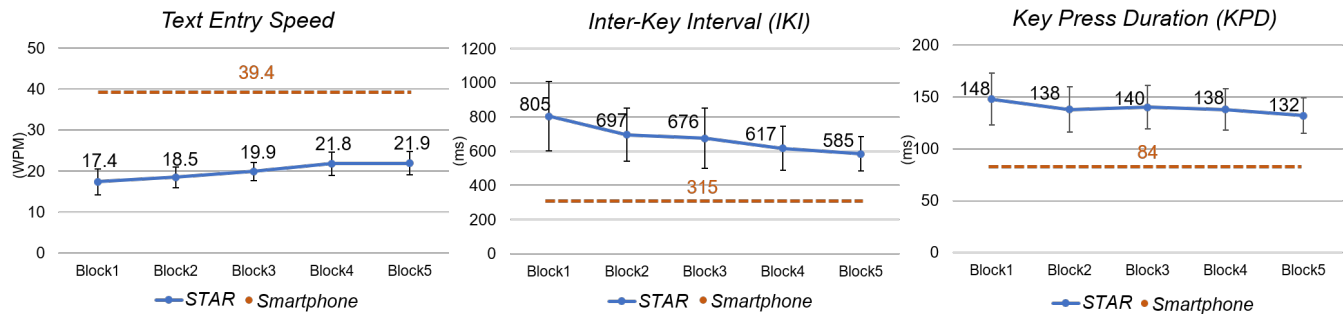
Figure 8: Line plots of mean Text Entry Speed, Inter-Key Interval (IKI)s, and Key Press Duration (KPD) from the evaluation (error bars show standard deviation).

and had a break of at least two minutes. In total, the experiment took approximately two hours to complete.

5.4 Metrics and Analysis

Several metrics were computed to understand the degree to which STAR approached state-of-the-art smartphone typing.

Text entry speed and error rate were computed as they are the two main metrics of performance in text entry research. Text entry speed was measured in Words Per Minute (WPM), where the effective word count was calculated based on the number of transcribed characters minus one divided by a nominal word length of five [67]. The entry duration was measured from the first key input to the last key input for each phrase. We computed three error rate metrics: the Uncorrected Error Rate (UER), the Corrected Error Rate (CER), and the number of times the backspace key was pressed [59]. The UER counted the errors in the submitted text based on the Minimum String Distance [67]. The CER was similar to the UER but also counted backspace usage as an error.

Micro-metrics such as inter-key interval (IKI) [21] and key press duration [18] were also collected to obtain an in-depth analysis of typing behaviors. Inter-Key Interval (IKI) is the time between two subsequent key inputs [21], and can correlate with text entry speed. Key Press Duration (KPD) is the time between key down and key up events [18]. For *STAR*, the KPD was measured from the moment the thumb touched the *Index Finger Surface* to the moment it was released. For *Smartphone*, the KPD was measured from the firing of the key down event to the key up event as measured by the Android API.

To measure the learning effect over blocks, one-way RM ANOVAs were used for metrics with normal distributions (i.e., the text entry speed, IKI, and key press duration metrics) and Friedman tests were used for metrics without normal distributions (i.e., error rate and backspace usage count metrics). For post-hoc comparisons, paired sample t-tests with a Bonferroni correction were performed.

In addition to qualitative metrics, we collected participants' subjective feedback via the post-interview. In addition to their overall experience, we asked their thoughts on the main factors that affected their performance difference between *STAR* and *Smartphone*.

5.5 Results

We first describe the quantitative results (Table 2) and then report on qualitative feedback from our study participants.

5.5.1 Text Entry Speed. The RM-ANOVA revealed significant differences across *STAR* blocks ($F(4,36) = 10.996$, $p < .001$). Post-hoc comparisons revealed that Block 1 (17.4 WPM) was significantly slower than Block 4 (21.8 WPM; $p < .05$) and 5 (21.9 WPM; $p < .001$) and that Block 2 (18.5 WPM) was significantly slower than Block 5 ($p < .005$). As the text entry speeds on the two blocks of *Smartphone* typing were not significantly different (39.9 vs. 39.0 WPM), they were averaged to form a baseline (39.4 WPM). By Block 5, participants were performing at up to 56% of the smartphone typing speed. Note that by the last block, the fastest typist reached 25.0 WPM with *STAR*, whereas the slowest typist reached 16.4 WPM.

5.5.2 Error Rates. The Friedman Test did not find any significant differences in UER across the five blocks of *STAR*. The average

UER was 0.2%, whereas, for *Smartphone*, the average UER was 0.5%. These UERs indicated that participants were able to complete the transcription task with few errors during both conditions.

The Friedman Test did not find any significant differences in CER across the five blocks of *STAR*. The average CER was 8.5%, whereas, for *Smartphone*, the average CER was 8.3%. For backspace usage, the Friedman Test did not find any significant differences across the five blocks of *STAR*. On average, participants used the backspace key 3 times during the *STAR* condition, and 2.6 times during the *Smartphone* condition. These similar CER and backspace usage results indicate that participants exhibited comparable degrees of correction behavior while using both *STAR* and *Smartphone* techniques.

5.5.3 Inter-Key Interval (IKI). The RM-ANOVA found significant differences across blocks for IKI in *STAR* ($F(4,36) = 12.703$, $p < .001$). As the mean IKI for *Smartphone* was 315 milliseconds, by Block 5, participants' IKI was 585 milliseconds, which was 54% of their smartphone performance and is consistent with the ratio of their text entry speed (i.e., 21.9 WPM / 39.4 WPM = 0.56).

5.5.4 Key Press Duration (KPD). The RM-ANOVA did not reveal significant differences across blocks for the time between key presses in *STAR*, with the average KPD across all five blocks being 139 milliseconds. As the mean KPD for *Smartphone* (84 milliseconds) was 1.6 times faster than the *STAR*, participants made 1.6 times faster key presses on the *Smartphone* touchscreen than *STAR*. This observation on KPD is important in that it allows us to take a deeper look at the low-level factors influencing the performance gap. The discussion on this will be continued in Section 5.6.

5.5.5 Subjective Feedback. In reference to their experience, some participants reported that they enjoyed using *STAR*. P3, for example, mentioned *"It was interesting that I am able to type without any on-hand device like a smartphone"* and P6 mentioned *"It was overall a very new and pleasant experience"*. P7 also mentioned *"It took some time to know how to type well, but it worked like smartphone typing once I got used to it."*

In response to the main causes of the performance issues with the techniques, five participants mentioned the limited fidelity of hand tracking. P6 noted that *"the [visualized] thumb couldn't follow my speed when I moved my finger quickly from one key to another ... though it is a slight delay, it makes me keep checking the [visualized] thumb position to avoid mistakes"*. P5 and P6 commented on accidentally touched top-row keys (i.e., q, w, ..., p) while trying to touch the word suggestion button above them. P5 mentioned *"When I quickly tap the word suggestion button, my (tracked) thumb wasn't following enough so the keys on the first row were selected instead"*.

Three participants wanted to use a version where the keyboard followed their hands and automatically aligned. Interestingly, this concept of the *Hands-Following* keyboard was explored during our initial design process but was abandoned due to limited hand tracking fidelity (Section 3.2.2). P7 mentioned *"I had to think not only about typing but also about where my hand surface is relative to the keyboard. I think it slows down the typing"*. P7 also suggested using one-handed thumb swipe gesture typing, as many people do on smartphones today. Lastly, P3 suggested using mid-air gestures to trigger key input, e.g., performing a mid-air thumb left swipe for backspace.

5.6 STAR vs. Smartphone: Performance Delta

STAR was able to achieve text entry speeds that were as high as 56% of the users' physical smartphone typing speed, while maintaining similar error rates in terms of UER and CER with 30 minutes of training. As a text entry technique that leverages the same two-thumb typing skill as a physical smartphone, we believe that it has the potential to achieve even closer performance. Here, we analyze the factors contributing to the performance delta between STAR and smartphone typing using both quantitative performance metrics and subjective feedback.

5.6.1 Hand Tracking. The most crucial factor contributing to the performance gap was the HMD's unreliable hand tracking. As identified by half of the participants, it was necessary for users to keep checking the visualized thumb position to avoid mistakes. This can be confirmed again with quantitative metrics. First, the reported IKIs indicate that users on average spent 1.9 times longer (585 ms) key-to-key input than that of *Smartphone* (315 ms). Users needed to keep monitoring the tracked thumb position to avoid errors, which eventually slowed down their key-to-key movement. Second, the reported KPDs indicate that users on average spent 1.7 times longer (139 ms) unit key press than that of *Smartphone* (84 ms). Although it is presumed that the duration of a general thumb tap would not be much different either on an *Index Finger Surface* or on a touchscreen, STAR showed considerably longer KPDs. We speculate that participants needed to wait for the lagging thumb marker to visually verify that the correct key had been pressed.

5.6.2 Virtual Keyboard Positioning. Another factor influencing the *STAR* performance is the *Stationary* positioning, which was dissimilar to how users employ smartphones. As outlined in Section 3.2.2, the smartphone touchscreen follows the user's hand position as users are holding it. Users can therefore leverage proprioceptive muscle memory when performing two-thumb typing, as each key in the keyboard is always at a fixed position relative to the hands. As we were unable to replicate this experience via a *Hands-Following* keyboard due to the HMD's unreliable hand tracking, participants had to instead align their hands to the *Stationary* keyboard while typing. P7 expressed concerns about having to pay constant attention to the hand position with respect to the keyboard, thus impacting the typing performance.

We again highlight that the limited hand tracking precision of the current state-of-the-art AR HMD was the most significant factor that restricted the performance of STAR. We believe STAR can reach typing performance closer to the smartphone over time with the continued advancement in HMD tracking technology. Incorporating additional sensing hardware that does not impede users' thumb movement (e.g., rings [3, 10, 40, 75], smartwatches [42], or bracelets [16, 54]) may also open an opportunity to realize robust hand and finger tracking for STAR.

6 LIMITATIONS AND FUTURE WORK

Our investigation into skill transfer for typing from smartphone to AR leads to several acknowledged limitations on our experimental

setup due to the sensing technology, as well as opportunities for further investigation of the design space.

6.1 Real-World Deployment

To make STAR usable in real-world applications, the first step will be to achieve robust hand tracking and thumb tap sensing. This may prove difficult using only a HMD since parts of the hand can be occluded from the HMD's perspective. To address this issue, it would be beneficial to incorporate sensing hardware that does not impede the user's thumb typing movements (e.g., rings [3, 10, 40, 75], smartwatches [42], or bracelets [16, 54]). For example, wearing a technology like Electroring [35] could immediately solve the challenge of thumb contact detection, although it would require users to wear rings on both hands. Further investigation of robust sensing techniques using cameras [10, 42], IMUs [40], magnetic fields [3, 11], EMG [54], sound [75], RF signals [36], and pressure [16] may lead to more robust hand and finger tracking for bare-hand text input methods like STAR.

Another crucial aspect to consider is to understand STAR *in-the-wild*. Although STAR was evaluated in a controlled lab study, text entry can occur in diverse situations, such as with primary tasks (e.g., listening to music or during a conversation), with various body poses (e.g., standing, sitting, leaning, or lying), with different activities (e.g., walking, or resting on a desk), or with dynamic real-world background. By evaluating STAR in a more extensive range of environments, we will discover additional opportunities and guidance for further refinement of the technique.

6.2 Practical Standalone Prototype vs. Simulating the Limits of the Idea

During our initial prototyping, we faced the trade-off between a practical standalone implementation that would show the technique's performance with today's standalone hardware and a proof-of-experience level prototype, simulating the limits of the technique with future technologies. While each approach has its respective advantages, we elected a balanced approach. The position sensing is enabled with standalone HMD hardware, while the contact detection is enhanced with the capacitive tapes. We believe this approach provides slightly higher external validity to our results than if we had used high-precision motion tracking equipment (e.g., Optitrack) for position sensing. In particular, several unobtrusive techniques could be utilized to achieve reliable contact sensing [35, 42], whereas replicating motion tracking equipment accuracy is still unachievable with onboard HMD hardware.

However, it is important to note that the performance limits of the STAR method under an ideal technology (e.g., simulation with a motion capture system) have yet to be investigated. Since this aspect was not explored within the scope of this study, it presents an intriguing question for future research endeavors.

6.3 Typing Performance Modeling with Tracking Precision

The hand tracking latency of Hololens 2 was approximately 90 ms, and a tracking jitter was around ± 1 mm. Within this range of tracking performance, the STAR method achieved 56% of the physical smartphone typing speed. To further explore the influence

of tracking latency and error on STAR's typing performance [48], future research could involve systematically manipulating these variables in an experiment and analyzing their specific impact on STAR's performance. This may reveal the level of tracking precision required to reproduce smartphone-level typing performance in practice.

6.4 One-handed STAR

While our research primarily focused on two-thumb smartphone typing, one-handed typing is also prevalent on smartphones [29, 47]. One-handed typing can involve either gesture swipe typing [37, 74] or character-level tap typing. As one-handed tap typing is presumably slower, gesture typing may be more suitable for the one-handed STAR (i.e., the one-handed gesture typing technique on a virtual keyboard overlain on the hand skin surface). Given that previous one-handed gesture typing techniques in AR/VR [12, 34, 40, 45, 73] have been mostly explored using an indirectly mapped cursor visualized on a large keyboard layout through a HMD, it could be an attractive direction for future research to investigate how to transfer smartphone gesture typing skills to on-skin, one-handed AR text entry.

7 CONCLUSION

This research presented a novel bare-hand text entry method that is analogous to physical smartphone two-thumb typing. Unlike earlier techniques that utilize new metaphors and movement patterns, the proposed STAR technique leverages familiar typing behaviors by transferring the same thumb-typing skills we use with physical smartphones to the AR context. The proposed technique was implemented with the current state-of-the-art AR HMD, and the evaluation study showed that it supported efficient text entry performance (i.e., 21.9 WPM), which was up to 56% of participants' physical smartphone typing speeds. As the tracking technologies for HMDs continue their rapid advancement, typing with STAR may approach the level of performance seen in smartphone typing. This progress will open up a promising opportunity for STAR to become the preferred method of ubiquitous AR text entry.

REFERENCES

[1] Arduino. 2022. Capacitive Sensing Library. https://playground.arduino.cc/Main/CapacitiveSensor/, last accessed Dec. 2022.

[2] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George W Fitzmaurice. 2017. Experimental evaluation of sketching on surfaces in VR.. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*, Vol. 17. 5643–5654.

[3] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: Subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 2043–2046.

[4] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In *Proc. Int. Conf. on Human-Computer Interaction with Mobile Devices and Services*. ACM, 251–260.

[5] Ana M Bernardos, David Gómez, and José R Casar. 2016. A comparison of head pose and deictic pointing interaction methods for smart environments. *Int. J. Human-Computer Interaction* 32, 4 (2016), 325–351.

[6] Taryn Bipat, Maarten Willem Bos, Rajan Vaish, and Andrés Monroy-Hernández. 2019. Analyzing the use of camera glasses in the wild. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 8 pages.

[7] Costas Boletsis and Stian Kongsvik. 2019. Controller-based text-input techniques for virtual reality: An empirical comparison. *Int. J. Virtual Reality (IJVR)* 19, 3 (2019).

[8] Doug A Bowman, Christopher J Rhoton, and Marcio S Pinho. 2002. Text input techniques for immersive virtual environments: An empirical comparison. In *Proc. Human Factors and Ergonomics Society Annual Meeting*, Vol. 46. SAGE, 2154–2158.

[9] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2527–2530.

[10] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proc. ACM Symposium on User Interface Software and Technology*. 549–556.

[11] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D input using two magnetic sensors. In *Proc. ACM Symposium on User Interface Software and Technology*. 237–244.

[12] Sibo Chen, Junce Wang, Santiago Guerra, Neha Mittal, and Soravis Prakkamakul. 2019. Exploring word-gesture text entry techniques in virtual reality. In *Extended Abstracts of the 2019 SIGCHI Conf. on Human Factors in Computing Systems*. 6 pages.

[13] Yi Fei Cheng, Tiffany Luong, Andreas Rene Fender, Paul Streli, and Christian Holz. 2022. ComforTable user interfaces: Surfaces reduce input error, time, and exertion for tabletop and mid-air user interfaces. In *Proc. IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)*. IEEE, 150–159.

[14] Nell Chitty. 2013. *User fatigue and eye controlled technology.* Technical Report. OCAD University.

[15] Wenzhe Cui, Rui Liu, Zhi Li, Yifan Wang, Andrew Wang, Xia Zhao, Sina Rashidian, Furqan Baig, IV Ramakrishnan, Fusheng Wang, et al. 2023. GlanceWriter: Writing Text by Glancing Over Letters with Gaze. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–13.

[16] Artem Dementyev and Joseph A Paradiso. 2014. WristFlex: Low-power gesture input with wrist-worn pressure sensors. In *Proc. ACM Symposium on User Interface Software and Technology*. 161–166.

[17] Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. 2018. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.

[18] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. 2019. Performance envelopes of virtual keyboard text input strategies in virtual reality. In *Proc. IEEE Int. Symp. Mixed and Augmented Reality (ISMAR)*. IEEE, 289–300.

[19] John J Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. *ACM Trans. on Computer-Human Interaction (TOCHI)* 25, 6 (2018), 1–40.

[20] Barrett Ens, Tovi Grossman, Fraser Anderson, Justin Matejka, and George Fitzmaurice. 2015. Candid interaction: Revealing hidden mobile and wearable computing activities. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 467–476.

[21] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. 2016. How we type: Movement strategies and performance in everyday typing. In *Proceedings of the 2016 chi conference on human factors in computing systems*. 4262–4273.

[22] Wenxin Feng, Jiangnan Zou, Andrew Kurauchi, Carlos H Morimoto, and Margrit Betke. 2021. Hgaze typing: Head-gesture assisted gaze typing. In *Proc. ACM Symp. on Eye Tracking Research and Applications*. 1–11.

[23] Hyunjae Gil and Ian Oakley. 2023. ThumbAir: In-Air Typing for Head Mounted Displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–30.

[24] Google. 2019. Google Glass 2. www.google.com/glass/, last accessed Jul. 2023.

[25] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. 2020. Qwertyring: Text entry on physical surfaces using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–29.

[26] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proc. ACM Symposium on User Interface Software and Technology*. 283–292.

[27] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the body as an input surface. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 453–462.

[28] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 1063–1072.

[29] Steven Hoober. 2013. How Do Users Really Hold Mobile Devices? https://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php, last accessed Mar. 2023.

[30] Steven Hoober. 2022. Designing for Touch. https://www.4ourthmobile.com/publications/designing-for-touch/, last accessed Mar. 2023.

[31] Shahram Jalaliniya, Diako Mardanbeigi, Thomas Pederson, and Dan Witzner Hansen. 2014. Head and eye movement as pointing modalities for eyewear computers. In *Proc. Int. Conf. Wearable and Implantable Body Sensor Networks Workshops*. IEEE, 50–53.

[32] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 3328–3339.

[33] Josh Kaufman. 2021. Google 10000 English. https://github.com/first20hours/google-10000-english, last accessed Dec. 2022.

[34] Florian Kern, Florian Niebling, and Marc Eriche Latoschik. 2023. Text input for non-stationary XR workspaces: investigating tap and word-gesture keyboards in virtual and augmented reality. *IEEE Trans. on Visualization and Computer Graphics* (2023).

[35] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. 2021. Electroring: Subtle pinch and touch detection with a ring. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 12 pages.

[36] Daehwa Kim, Keunwoo Park, and Geehyuk Lee. 2021. Atatouch: Robust finger pinch detection for a vr controller using rf return loss. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–9.

[37] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A large vocabulary shorthand writing system for pen-based computers. In *Proc. ACM Symposium on User Interface Software and Technology*. 43–52.

[38] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free text entry using gaze paths. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 1952–1956.

[39] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A Lee, and Mark Billinghurst. 2018. Pinpointing: Precise head-and eye-based target selection for augmented reality. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*.

[40] Chen Liang, Chi Hsia, Chun Yu, Yukang Yan, Yuntao Wang, and Yuanchun Shi. 2023. DRG-Keyboard: Enabling subtle gesture typing on the fingertip with dual IMU rings. *Proc. ACM Conf. Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–30.

[41] Winwaed Software Technology LLC. 2012. Calculating Word and N-Gram Statistics from a Wikipedia Corpora. https://www.winwaed.com/blog/2012/04/16/calculating-word-and-n-gram-statistics-from-a-wikipedia-corpora/, last accessed Dec. 2022.

[42] Christian Loclair, Sean Gustafson, and Patrick Baudisch. 2010. PinchWatch: a wearable device for one-handed microinteractions. In *Proc. ACM Conf. Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, Vol. 10. ACM.

[43] Xueshi Lu, Difeng Yu, Hai-Ning Liang, and Jorge Goncalves. 2021. iText: Hands-free text entry on an imaginary keyboard for augmented reality systems. In *Proc. ACM Symposium on User Interface Software and Technology*. 815–825.

[44] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Proc. Extended Abstracts of SIGCHI Conf. on Human Factors in Computing Systems*. 754–755.

[45] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A mid-air word-gesture keyboard. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 1073–1082.

[46] Microsoft. 2019. Hololens 2. https://www.microsoft.com/hololens, last accessed Jul. 2023.

[47] Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. 2019. How do people type on mobile devices? Observations from a study with 37,000 volunteers. In *Proc. Int. Conf. on Human-Computer Interaction with Mobile Devices and Services*. 1–12.

[48] Andriy Pavlovych and Wolfgang Stuerzlinger. 2009. The tradeoff between spatial jitter and latency in pointing tasks. In *Proc. ACM SIGCHI Symp. on Engineering interactive computing systems*. 187–196.

[49] Svetlana Pinet, Christelle Zielinski, F Alario, Marieke Longcamp, et al. 2022. Typing expertise in a large student population. *Cognitive Research: Principles and Implications* 7, 1 (2022), 1–19.

[50] Jiamin Ping, Bruce H Thomas, James Baumeister, Jie Guo, Dongdong Weng, and Yue Liu. 2020. Effects of shading model and opacity on depth perception in optical see-through augmented reality. *J. of the Society for Information Display* 28, 11 (2020), 892–904.

[51] Kathleen J Price, Min Lin, Jinjuan Feng, Rich Goldman, Andrew Sears, and Julie A Jacko. 2004. Data entry on the move: An examination of nomadic speech-based text entry. In *ERCIM Workshop on User Interfaces for All*. Springer, 460–471.

[52] Daniel Röltgen and Roman Dumitrescu. 2020. Classification of industrial Augmented Reality use cases. *Procedia CIRP* 91 (2020), 93–100.

[53] Sherry Ruan, Jacob O Wobbrock, Kenny Liou, Andrew Ng, and James A Landay. 2018. Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones. *Proc. ACM Conf. Interactive, Mobile, Wearable*

and Ubiquitous Technologies 1, 4 (2018), 1–23.

[54] T Scott Saponas, Desney S Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A Landay. 2009. Enabling always-available input with muscle-computer interfaces. In *Proc. ACM Symposium on User Interface Software and Technology*. 167–176.

[55] Nitin Sawhney and Chris Schmandt. 2000. Nomadic radio: Speech and audio interaction for contextual messaging in nomadic environments. *ACM Trans. on Computer-Human Interaction (TOCHI)* 7, 3 (2000), 353–383.

[56] Abigail J Sellen, Gordon P Kurtenbach, and William AS Buxton. 1992. The prevention of mode errors through sensory feedback. *Human-Computer Interaction* 7, 2 (1992), 141–164.

[57] Aaron Smith. 2015. *US smartphone use in 2015.* Technical Report. Pew Research Center.

[58] Snap. 2019. Spectacles 3. https://www.spectacles.com/, last accessed Jul. 2023.

[59] R William Soukoreff and I Scott MacKenzie. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 113–120.

[60] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. 2018. Selection-based text entry in virtual reality. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 1–13.

[61] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 3643–3652.

[62] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[63] Hemant Bhaskar Surale, Yu Jiang Tham, Brian A Smith, and Rajan Vaish. 2022. Arcall: Real-time ar communication using smartphones and smartglasses. In *Augmented Humans*. 46–57.

[64] Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y Chen. 2015. PalmType: Using palms as keyboards for smart glasses. In *Proc. Int. Conf. Human-Computer Interaction with Mobile Devices and Services*. 153–160.

[65] Dan Wang, Zheng Xiang, and Daniel R Fesenmaier. 2016. Smartphone use in everyday life and travel. *Journal of Travel Research* 55, 1 (2016), 52–63.

[66] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. Digitouch: Reconfigurable thumb-to-finger input and text

entry on head-mounted displays. *Proc. ACM Conf. Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–21.

[67] Jacob O Wobbrock. 2007. Measures of text entry performance. *Text entry systems: Mobility, accessibility, universality* (2007), 47–74.

[68] Haijun Xia, Michael Glueck, Michelle Annett, Michael Wang, and Daniel Wigdor. 2022. Iteratively designing gesture vocabularies: A survey and analysis of best practices in the HCI literature. *ACM Trans. on Computer-Human Interaction (TOCHI)* 29, 4 (2022), 1–54.

[69] Zheer Xu, Weihao Chen, Dongyang Zhao, Jiehui Luo, Te-Yen Wu, Jun Gong, Sicheng Yin, Jialun Zhai, and Xing-Dong Yang. 2020. Bitiptext: Bimanual eyes-free text entry on a fingertip keyboard. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 1–13.

[70] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. 2019. Tiptext: Eyes-free text entry on a fingertip keyboard. In *Proc. ACM Symposium on User Interface Software and Technology*. 883–899.

[71] Xin Yi, Chen Liang, Haozhan Chen, Jiuxu Song, Chun Yu, Hewu Li, and Yuanchun Shi. 2023. From 2D to 3D: Facilitating single-finger mid-air typing on QWERTY keyboards with probabilistic touch Mmodeling. *Proc. ACM Conf. Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–25.

[72] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. Atk: Enabling ten-finger freehand typing in air based on 3D hand tracking data. In *Proc. ACM Symposium on User Interface Software and Technology*. 539–548.

[73] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, dwell or gesture? Exploring head-based text entry techniques for HMDs. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 4479–4488.

[74] 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.

[75] Cheng Zhang, Anandghan Waghmare, Pranav Kundra, Yiming Pu, Scott Gilliland, Thomas Ploetz, Thad E Starner, Omer T Inan, and Gregory D Abowd. 2017. FingerSound: Recognizing unistroke thumb gestures using a ring. *Proc. ACM Conf. Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–19.

[76] Mingrui Ray Zhang, Shumin Zhai, and Jacob O Wobbrock. 2022. TypeAnywhere: A QWERTY-based text entry solution for ubiquitous computing. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. 1–16.