TypeRing: Ten-Finger Surface Typing with Dual IMU Rings

Zhuojun Li
Department of Computer Science and Technology, BNRist, Tsinghua University
Beijing, China
lizj23@mails.tsinghua.edu.cn

Chun Yu*
Department of Computer Science and Technology, BNRist, College of AI, Tsinghua University
Beijing, China
chunyu@tsinghua.edu.cn

Zhe He
Department of Computer Science and Technology, BNRist, Tsinghua University
Beijing, China
hz23@mails.tsinghua.edu.cn

Lubin Wang
Weiyang College, Tsinghua University
Beijing, China
wanglb23@mails.tsinghua.edu.cn

Chengwen Zhang
Department of Computer Science and Technology, BNRist, Tsinghua University
Beijing, China
zcw25@mails.tsinghua.edu.cn

Mingyuan Du
Department of Computer Science and Technology, Tsinghua University
Beijing, China
dmy23@mails.tsinghua.edu.cn

Weinan Shi*†
Department of Computer Science and Technology, Tsinghua University
Beijing, China
swn@tsinghua.edu.cn

Yuanchun Shi*‡
Department of Computer Science and Technology, Tsinghua University
Beijing, China
shiyc@tsinghua.edu.cn

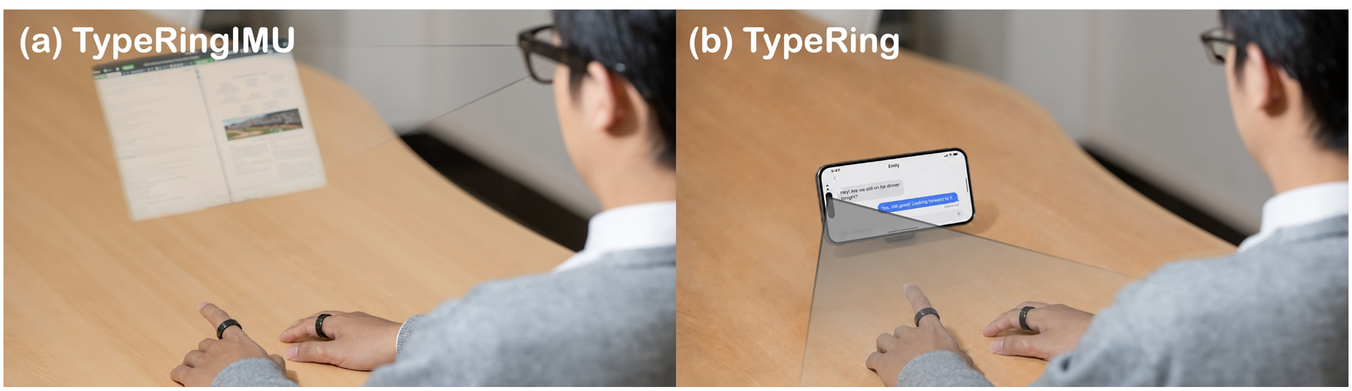


Figure 1: Demo scenarios of our method. We present a novel ten-finger surface typing method with dual IMU rings, comprising two variants: (a) TypeRingIMU, which relies solely on two IMU rings to enable ten-finger typing on any flat surface. The user is typing on the table with visual feedback provided through AR glasses; (b) TypeRing, which enhances TypeRingIMU via optical-inertial fusion, leveraging the front-facing camera available on mobile devices. The user is typing in front of a smartphone, expanding its interaction space to the table.

Abstract

Ten-finger surface typing promises a portable alternative to physical keyboards and touchscreens, making it well-suited for mobile scenarios, but remains challenging to balance sensing cost with input performance. We ask whether it can be enabled with two IMU rings, without relying on bulky wearables or continuous vision processing. We first present TypeRingIMU, which detects rapid ten-finger touches and identities with an F1-score of 98.4%, enabling efficient ten-finger typing on any flat surface using only two IMU rings. We further enhance it via optical-inertial fusion that opportunistically detects touch positions from IMU-triggered touch events, leveraging the front-facing camera already available on mobile

*Also with Key Laboratory of Pervasive Computing, Ministry of Education.
†Corresponding author.
‡Also with Qinghai University.

UIST '26, Detroit, MI, USA

ACM ISBN 979-8-4007-2856-3/2026/11
https://doi.org/10.1145/3830398.3830512

devices, namely TypeRing. It removes fingering constraints, thus improving input naturalness and accuracy. Our method achieves 28.6 WPM input speed (63.5% of physical keyboards), with top-1 and top-12 accuracies of 88.5% and 98.1%, significantly outperforming current methods and demonstrating strong potential for efficient, convenient, and user-friendly mobile text input.

CCS Concepts

• **Human-centered computing → Text input**.

Keywords

Ring, IMU, Touch Detection, Optical-Inertial Fusion, Text Input

ACM Reference Format:
Zhuojun Li, Chun Yu, Zhe He, Lubin Wang, Chengwen Zhang, Mingyuan Du, Weinan Shi, and Yuanchun Shi. 2026. TypeRing: Ten-Finger Surface Typing with Dual IMU Rings. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST '26), November 02–05, 2026, Detroit, MI, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3830398.3830512

1 Introduction

Typing is one of the most fundamental input techniques in human-computer interaction [7, 18, 42], yet the most familiar approaches to users - physical and touchscreen keyboards - are constrained by portability and interaction space, making them unsuitable for mobile text input. Ten-finger surface typing, where users type on any common flat surface as if on a keyboard, offers a promising alternative. It preserves the familiarity and efficiency of ten-finger typing while eliminating the need for a physical keyboard or touchscreen. These advantages make it a promising method for mobile text input, aligning with the long-standing HCI vision of enabling users to type anywhere.

Achieving ten-finger surface typing, however, remains highly challenging due to sensing difficulties. The system must detect rapid and subtle finger touches with finger identities [9, 32, 44] or positions [26, 36, 40, 45], and decode intended text from them. Existing approaches struggle to meet these requirements. Vision-based methods provide rich touch information but demand continuous image processing, which is computationally expensive and vulnerable to occlusion [32, 36, 45]. IMU-based methods are computationally efficient but often require bulky hardware [47, 57] or impose strict fingering constraints [44] (i.e., the standard mapping rule that each key should be tapped by which finger), limiting input naturalness. As a result, it remains difficult to balance sensing cost with input performance, leaving ten-finger surface typing an open challenge for mobile scenarios.

We ask whether it can be solved with two IMU rings. We first present **TypeRingIMU**, which uses only two IMU rings worn on index fingers to support ten-finger surface typing. We design a lightweight touch detection model that performs a 5-category classification within each hand, distinguishing thumb, index, middle, ring/pinky (treated as a single class to balance data distribution and reduce sensing difficulty) finger touches and negative samples, with an F1-score of 98.4%. The IMU window of our model is 100 ms. Thus, it supports rapid touches up to 10 times per second within each hand, allowing fast finger motions during typing. Furthermore, unlike prior work that enforces standard fingering to map finger identities to characters, our approach leverages probabilistic mappings learned from user data and gradually personalizes them with continued use, reducing learning difficulty. As a result, TypeRingIMU achieves a text input speed of 22.7 WPM (50.4% of physical keyboards), with top-1 and top-12 word-level decoding accuracies of 81.7% and 97.9% (top-N means the probability of target word being in the top-N word candidates). It outperforms the state-of-the-art IMU-based method (TapType [44]: 19.2 WPM, 25.8% of their users on physical keyboards) with minimal sensing cost of only two IMUs.

Building on this foundation, we further introduce **TypeRing**, which enhances TypeRingIMU through optical-inertial fusion. We leverage an RGB camera placed in front of the user's hands to detect horizontal touch positions. Since many existing mobile devices are already equipped with front-facing cameras (e.g., smartphones and tablets), this design has strong potential to be applied in mobile text-entry scenarios. The decoder further leverages touch positions to predict the input word, rather than using finger identities. This design eliminates the fingering constraints of TypeRingIMU, thereby improving both input naturalness and decoding accuracy. More importantly, our fusion strategy requires the camera only to process keyframes associated with IMU-detected touches, which account for only 6.3% of all frames from the 30-FPS camera in our experiment, significantly reducing computational cost. With this improvement, TypeRing achieves a text input speed of 28.6 WPM (63.5% of physical keyboards), with top-1 and top-12 accuracies reaching 88.5% and 98.1%, highlighting its potential as a natural and efficient mobile text input method.

In summary, our contributions are threefold:

(1) We propose a lightweight IMU-based finger touch event and identity detection model that recognizes rapid finger touches with an F1-score of 98.4%, enabling efficient ten-finger surface typing with two IMU rings.
(2) We design an optical-inertial fusion-based finger touch position detection algorithm triggered by IMU-detected touch events. It avoids continuous vision processing, thus reducing computational cost.
(3) We present the first dual-IMU-ring solution for ten-finger surface typing. It includes TypeRingIMU, which emphasizes simplicity and portability, and TypeRing, which leverages finger touch position to eliminate the typing fingering constraints, thereby enhancing input naturalness and accuracy. These two variants complement each other and allow users to flexibly choose according to different mobile scenarios.

2 Related Work

2.1 IMU-based Touch Recognition

Detecting finger touches, which trigger character input for text entry, is easily supported by physical keyboards or touchscreens. However, detecting touch events on arbitrary surfaces is non-trivial. One promising solution is to use IMUs, as they are sensitive to fine-grained finger movements. This problem falls within the scope of IMU-based hand gesture recognition [15, 48, 56], where prior work explored various wearable devices such as gloves [31, 34], smartwatches [20, 33, 44, 52, 53, 58], rings [12, 13, 27, 39, 50, 55], combined devices [24], and other specialized devices (e.g., Tap Strap [47, 57]) to detect hand and finger gestures.

Specifically, detecting finger touches on surfaces for ten-finger text input poses stricter requirements than general gesture recognition, including: (1) the device must be lightweight enough for continuous use in mobile scenarios; (2) the method must support high-frequency touch detection for rapid typing motions; (3) the algorithm must also distinguish touch finger identities (e.g., index finger) to enable accurate text decoding. Most IMU-based gesture recognition methods use relatively long temporal windows (e.g., 0.5-2 s [24, 53, 58]) to capture motion features, which are unsuitable for rapid typing motions. The most related work is TapType [44], which uses two IMU wristbands with a 1600 Hz sampling rate and an 80 ms window length to detect finger touches and identities with a cross-user F1-score of 86%. However, wrist-worn sensors are farther from the fingers, and the high sampling rate is not suitable for continuous use in mobile deployment. In contrast, our work places IMU rings on the index fingers to sense finger motions more directly. We limit the sampling rate to 200 Hz and use a 100 ms window length, which reduces the sensing cost and also support rapid touches up to 10 times per second within each hand.

2.2 Optical-Inertial Fusion

Optical and inertial sensors exhibit complementary characteristics: cameras capture global spatial information but suffer from occlusion and high computational cost, while IMUs capture fine-grained motion with low cost but lack spatial awareness. This complementarity has motivated optical-inertial fusion in various interactive sensing tasks, including but not limited to navigation [5, 8, 38], localization [6, 49, 54], image stablization [2, 22, 41], motion capture [3, 16, 17, 54], and hand tracking [21, 23, 25, 33, 43, 52].

For surface typing, sensing can be decomposed into two tasks: detecting finger touch events and estimating touch positions, which naturally align with optical-inertial fusion. Pure IMU-based methods [44, 47] rely only on finger identities for decoding and often require standard QWERTY typing fingering, which is not suitable for non-expert users. Vision-based methods [9, 32, 36, 45] provide spatial information but suffer from occlusion. TapID [33] partially addressed this issue by combining an IMU wristband with cameras. SurfaceXR [52] further supports various surface interactions by combining an IMU smartwatch with Meta Quest, but it is not designed for ten-finger typing. Building on this idea, we explore a lightweight optical-inertial fusion design using dual IMU rings and a front-facing camera for surface typing, which notably improves sensing performance and avoids occlusions by the hand palms. Furthermore, we introduce an event-triggered touch position detection framework, drastically reducing the computational cost.

2.3 Surface Typing

Text input is one of the most fundamental human-computer interaction task. Prior work has shown that users can transfer their QWERTY typing muscle memory to surface typing [10, 26, 28, 37, 40]. Therefore, once touch events and positions are detected, users' intended text can be decoded using probabilistic language models.

Surface typing methods typically decode text using probabilistic language models based on the Bayesian algorithm [11]. Finger identity-based approaches [9, 32, 44, 47] map each finger to characters but lack spatial information and often require standard QWERTY fingering. In contrast, touch position-based methods [13, 28, 35, 37, 40] leverage spatial touch information to improve decoding accuracy. Recent works such as ResType [26] further enhance decoding through augmented Bayesian models. Building on these approaches, we explore two complementary solutions: TypeRingIMU, a pure IMU-based method using probabilistic finger-to-character mappings, and TypeRing, which further incorporates touch positions through optical-inertial fusion to improve decoding accuracy and relax fingering constraints.

3 Method

3.1 Framework Design

We aim to enable ten-finger touch typing on any flat surface with two IMU rings. This problem is decomposed into three sub-tasks: (1) ten-finger touch event detection and finger identification using dual IMU rings, (2) an optional camera module for touch position detection, and (3) text input decoding with either finger identities or touch positions, as shown in Figure 2. As a result, our method enables ten-finger surface typing with or without touch position sensing (depending on using scenarios), forming the dual-system design of TypeRingIMU and TypeRing.

In the first sub-task (Figure 2a), the IMU module detects touch events and classifies finger identities based on vibration patterns captured at the index fingers. In the second sub-task (Figure 2b), once a touch event is detected, the camera module retrieves a keyframe to estimate the horizontal touch position. This event-triggered strategy avoids continuous vision processing and reduces computational cost. Finally, recognized finger identities or touch positions are fed into an augmented Bayesian decoder for word prediction (Figure 2c). We also integrate ring-mounted touchpads to support deletion and candidate selection.

3.2 Touch Detection and Finger Identification

3.2.1 Design. We design a CNN-based classifier to detect touch events and finger identities. To support rapid typing motions with low latency, we adopt a short-window design. The IMU sensor samples at 200 Hz, and each input window contains 20 frames (100 ms). Positive samples are defined as finger touches occuring near the center of the window (± 10 ms for robustness), and overlapping touches within a window are not supported. Therefore, it enables high-frequency tap detection up to 10 times/s within each hand (taps across two hands are independent). The limited temporal context introduced by the short window increases sensing difficulty, making touch detection more challenging.

Moreover, since users' typing behavior is unevenly distributed across fingers, with index and middle fingers contributing the majority of touches (40.6% and 39.4%), we place the IMU rings on the index fingers to maximize sensing capability. Since ring and pinky touches are less frequent and exhibit lower separability, we merge them into a single class. They only account for 13.9% of all finger touches, which is a reasonable trade-off between sensing difficulty

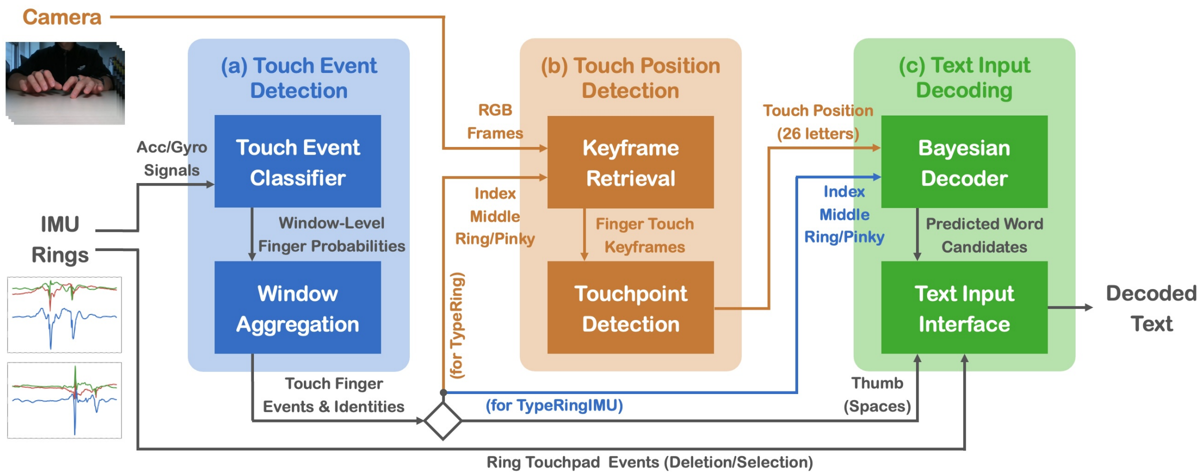


Figure 2: Our method consists of three modules: (a) touch event detection and finger identification from IMU data; (b) touch position detection triggered by IMU detected touch events; (c) text input decoding either from finger identities or touch positions. The interaction is further completed by ring touchpad-enabled deletion and candidate selection. Blue paths are used only by TypeRingIMU, while orange paths are used only by TypeRing. Gray paths are shared by both systems.

and input performance. The final classifier predicts five categories: negative (non-typing), thumb, index, middle, and ring/pinky.

3.2.2 Touch Detection Model. The network takes a 6×20 IMU window as input and outputs a 5-class probability over finger identities. It is built on customized Inception blocks with 375K parameters, suitable for deployment on mobile devices. Detailed architecture is provided in Appendix A.1. For real-time inference, we apply a sliding-window strategy on the 200 Hz IMU stream, extracting a 20-frame window every two frames, resulting in a real-time processing frame rate of 100 FPS. To suppress duplicate detections from overlapping windows, we aggregate predictions and trigger a touch event only when the same positive label is observed for 4 consecutive windows.

3.3 Touch Position Detection

3.3.1 Design. In our framework, the camera is placed in front of the user's hands, simulating the front-facing camera commonly available on mobile devices. Touch position detection is triggered by IMU-detected touch events, which provides prior knowledge of contact timing. It also simplifies the problem, compared to determining touch events and positions simultaneously. The system processes only keyframes corresponding to detected touches rather than continuously analyzing all frames, reducing computational cost compared to vision-only approaches.

3.3.2 Keyframe Retrieval. To refine the touch timestamp within each IMU window, we adopt a rate-of-change score (RCS) method [44]. Given acceleration data $\boldsymbol{a}_t \in \mathbb{R}^3$ over a window of length T, RCS is computed as $RCS_t = \frac{1}{d}RCS_{t-1} + |\|\boldsymbol{a}_t\|_2 - \|\boldsymbol{a}_{t-1}\|_2|$ ($RCS_0 = 0.0$), where $d = 2.0$ in our implementation. The contact timestamp is determined as $t_c = \arg\max_t RCS_t$, and the corresponding camera frame is retrieved as the touch keyframe. Unlike prior work that uses RCS for event filtering, we use it only for temporal localization after touch detection, improving robustness to false triggers.

3.3.3 Touchpoint Detection Algorithm. Touchpoint detection is based on two observations: (1) the touching finger exhibits stronger motion than other fingers and the background, enabling segmentation via frame differencing; (2) under a front-facing view, the touchpoint corresponds to the lowest fingertip in the image.

The algorithm is demonstrated in Figure 3. Given the refined timestamp t_c, we retrieve the current frame F_1 and a previous frame F_0 at $t_c - 100$ ms. Both frames are converted to the YCrCb color space as F_0' and F_1' and differenced as $dF = F_1' - F_0'$. The moving finger is segmented by thresholding the Cr and Cb channels of dF, followed by morphological filtering to remove noise. The touchpoint is then defined as the lowest point of the segmented mask.

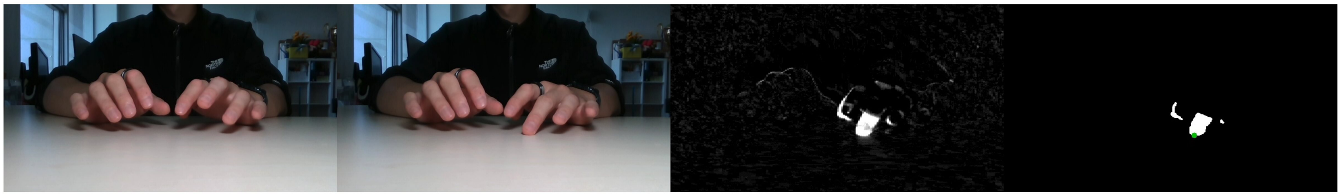
3.4 Text Input Decoding

3.4.1 Augmented Bayesian Decoder. We decode text with an augmented Bayesian decoder [26], which combines a touch model and a language model. Given a word W and touch sequence I, decoding follows $P(W|I) \propto P(I|W)P(W)$, where $P(I|W)$ denotes the touch model and $P(W)$ the language model. Following ResType [26], the touch model integrates both absolute and relative components:

$$\begin{aligned} P_{abs}(I|W) &= \prod_{k=1}^{n} P(i_k|w_k) \\ P_{rlt}(I|W) &= P(i_1|w_1) \cdot \prod_{k=2}^{n} P(i_k - i_{k-1}|w_{k-1}, w_k) \end{aligned} \tag{1}$$

where w_k and i_k denote individual character and touch respectively. They are combined with linear weighting $P_{aug}(I|W) = \lambda \cdot P_{abs}(I|W) + (1 - \lambda) \cdot P_{rlt}(I|W)$.

Unlike touchscreen-based methods, our camera view compresses the vertical dimension. Therefore, we model touch positions only



(a) Previous Keyframe at t_c-100ms (b) Current Keyframe at t_c (c) Keyframe Difference (d) Finger Mask and Touchpoint

Figure 3: Touch position detection demo. (a)&(b) the previous and current keyframes; (c) the keyframe difference, showing 10.0 × (the Cr - Cb channel) in the YCrCb color space; (d) the mask of the moving finger and the detected touchpoint.

along the horizontal (X) axis using a 1D Gaussian distribution, improving robustness to viewpoint variations. For TypeRingIMU, touch positions are replaced by finger identities, and the touch model is derived from the conditional probability of finger identities given letters learned from typing data. The decoding framework remains identical. For the language model, we adopt a unigram-bigram word model [13]. It contains 10000 English words, covering most common expressions in English. Out-of-vocabulary (OOV) words are not supported.

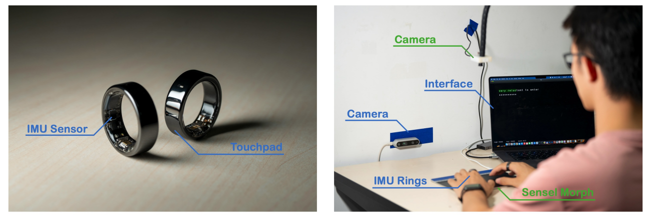
For interaction, we use non-thumb fingers to input letters, and use thumbs to input spaces. Entering a space triggers Bayesian word prediction and displays the top-12 candidates. We integrate a touchpad on each IMU ring for input control: the right-hand touchpad cycles through candidates, and the left-hand touchpad performs word deletion, completing the text input loop.

3.4.2 Personalization. To account for cross-user variability, we design an online personalization strategy for the touch model. During use, confirmed input text is treated as ground truth to update the conditional probability distribution of finger identities or touch positions given letters. The personalized model is linearly combined with a pre-trained general model, with the personalization weight gradually increased over time.

4 Study One: Data Collection

To implement and evaluate our system, we construct a comprehensive surface typing dataset consisting of two parts: (1) a large-scale ten-finger surface typing dataset for model training and evaluation, and (2) a negative gesture dataset to improve the robustness of touch event detection.

4.1 Apparatus



(a) The Dual IMU Rings (b) The Apparatus Setup

Figure 4: The apparatus setup in the user studies. (a) dual IMU rings; (b) a multiple-sensor platform for collecting the typing and ground truth data. Sensors marked in green are used only for data annotation.

Figure 4 illustrates the data collection setup. We first customized two IMU rings worn on the index fingers, each integrating a 6-axis IMU sensor (ICM-42688) and a touchpad (similar to τ-Ring [46], which is open to use for research). The IMU samples 3-axis accelerometer and 3-axis gyroscope data at 200 Hz and streams data to a host device via BLE in real time. The touchpad supports finger touch and release events, which are used for deletion and candidate selection during text input. For the camera, we use a RealSense D435 placed in front of the user to simulate a front-facing mobile camera, when users place their mobile devices on the surface and interact with them. Only the RGB stream is used, downsampled to 360 × 640 at 30 FPS. The camera is positioned approximately 6 cm above the table surface, approximating the height of front-facing cameras on horizontally placed mobile devices. For annotation and evaluation, we employ a Sensel Morph touchpad to capture multi-finger touch positions and a downward-facing camera to determine finger identities using Mediapipe [29]. These sensors are used solely for data labeling and are not involved in the runtime algorithms.

To synchronize all sensors, the biggest challenge is the IMU rings, since the are connected to the host computer via BLE, which have different time sources. To address this, after a ring connects, we estimate a linear clock mapping through repeated timestamp exchanges between rings and the host computer, which takes no longer than 30 s and can reach about 1 ms precision. Since cameras and the touchpad is wired to the host computer, their timestamps can be directly synchronized.

4.2 Typing Data Collection

To construct the typing dataset, participants perform blind ten-finger surface typing by imagining a QWERTY keyboard on the touchpad and transcribing English phrases. All sensors are synchronized and connected to a host computer for data recording. Since our final algorithms have not been implemented yet, finger touches are detected by the touchpad during typing. Each time, a target phrase (only containing 26 English lowercase letters and spaces) is displayed on the screen. Below the target phrase, an input area shows the input characters, which contains only asterisks, each generated by a finger touch as visual feedback. Users are required to ensure that their intended input to be aligned with the target phrase. They are regarded as the ground truth characters for finger touches. If they feel any error, the entire phrase are required to be deleted by left swiping and typed again. The typing fingering is not restricted, allowing participants to follow their own habits.

We recruited 32 participants (21 males & 11 females, aged 19-31, M=24.8, SD=3.1). Each completed 10-20 sessions, with 20 phrases per session randomly sampled from the MacKenzie phrase set [30]. In total, we collected 387 sessions comprising 7740 phrases and 259726 finger touches with synchronized IMU and camera data. Touch distributions across fingers are 16.25% (thumb), 36.50% (index), 33.42% (middle), 11.67% (ring), and 2.19% (pinky), supporting our design choice of wearing the rings on index fingers.

4.3 Negative Data Collection

To improve the robustness of touch event detection, we construct a negative gesture dataset to capture non-typing motions that may be confused with finger touches. We predefined 29 common gesture categories (e.g., in-air tapping, waving, clenching) and additionally allowed participants to contribute their own gestures.

We recruited 70 participants (33 males & 37 females, aged 19-31, M=23.8, SD=3.3) to perform gestures from these categories. We applied two types of data recording sessions: (1) participants follow a counter, performing one gesture at a time every 2 seconds for one minute; (2) participants repeat a gesture continuously for one minute. This design simulates discrete gestures and continuous motions respectively, which are both common during typing. In total, we collected 92041 IMU samples as negative data. The detailed gesture distribution is provided in Appendix Table 4.

5 Implementation

5.1 Data Preprocessing

We extract annotated touch samples from the dataset to support model training and evaluation, including: (1) touch timestamps, (2) touchpad ground-truth positions, (3) synchronized 20-frame IMU windows centered at each touch, (4) corresponding input characters, (5) touch finger identities, and (6) camera-detected touch positions.

Touch timestamps and positions are directly obtained from the Sensel Morph touchpad. IMU windows are cropped around each touch moment. Target characters are aligned with touch events by replaying the typing recordings and excluding deleted inputs. Finger identities are annotated by detecting fingertip locations from the downward-facing camera using MediaPipe Hands [29], and aligning them to the touchpad coordinates via linear transformation. For evaluating camera-based touch positions, detected positions are aligned to the touchpad coordinates and compared against touchpad ground truth. The distribution of camera-detected touch positions is further used as the touch model in the augmented Bayesian decoding algorithm.

5.2 Touch Event Detection

5.2.1 Implementation. We construct the IMU dataset from the typing and negative gesture data described in Section 4. To unify training across hands, all left-hand IMU signals are mirrored to the right-hand coordinate system due to their symmetric placement. The final dataset contains 259726 positive touch samples and 280522 negative samples, including both gesture data (92041) and non-touch IMU segments sampled from typing sessions (188481) to reduce false triggers in realistic scenarios.

We evaluate the model under cross-user settings by splitting users into training and validation groups and performing 8-fold cross-user validation. A separate test set is not used, as real-time performance is further evaluated in Study Two (Section 6). The models are trained on an NVIDIA 4090 GPU, with a batch size of 512 for 200 epochs. The model with the best validation F1-score is adopted for subsequent offline and realtime evaluations.

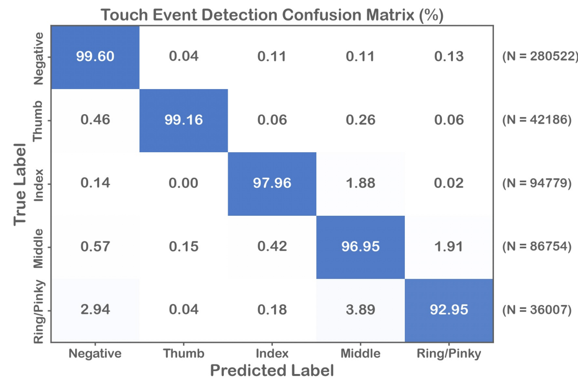


Figure 5: The confusion matrix of the touch event detection model. All percentages are normalized within each row.

5.2.2 Evaluation. Under 8-fold cross-user validation, the model achieves an average F1-score of 98.39%. For comparison, TapType [44] has a cross-user F1-score of 86%. Since it is designed for six positive classes (five fingers and palm touches), and handles negetive samples by a separate thresholding method, this comparison is only contextual. It partially demonstrates the superiority of our dual-IMU-ring design. As shown in Figure 5, our model achieves good performance recognizing negative samples and thumb touches with over 99% accuracy. Errors mainly come from confusion between middle and ring/pinky touches, or ring/pinky touches being unrecognized.

However, these errors have limited impact in practice. First, ring/pinky touches account for only 13.9% of all touches. Second, for TypeRing (with camera), the IMU module is only required to provide accurate touch timestamps for camera triggering. It could thus be treated as a 3-class problem (negative, thumb, other fingers), where the F1-score increases to 99.39%. Specific finger identities are only useful for TypeRingIMU. In conclusion, our method achieves the state-of-the-art performance of detecting ten-finger touch events and identities only using two IMUs, ensuring reliable touch recognition in typing scenarios.

5.3 Touch Position Detection

5.3.1 Implementation. Following Section 3.3, we first determine the touch timestamp t_c using the RCS-based method, and retrieve two keyframes F_0 (t_c − 100 ms) and F_1 (t_c). We compute their difference dF in the YCrCb color space, segment the moving finger region by thresholding the Cr and Cb channels, and define the touch position as the lowest point in the segmented mask. Denote the Cr and Cb channels in dF as d_r and d_b. Thresholds on the Cr and Cb differences are determined empirically from annotated typing images. We randomly sampled 132 images from the dataset, annotated moving finger and background regions, and analyzed their distributions in the Cr and Cb channels. After that, we fit a linear decision boundary to separate the regions, resulting in: $2d_r + d_b \geq 24$ and $d_b \leq 0$. We further apply one time of erosion and one time of dilation with a 3×3 kernel to remove noises in the finger mask. We use the lowest point in the final mask as the touch position, which achieves satisfying performance on the dataset.

Finger	Mean (cm)	Median (cm)	95% (cm)	1cm-acc (%)
Thumb	0.85	0.34	5.07	87.72
Index	0.30	0.22	0.69	98.82
Middle	0.35	0.30	0.77	98.69
Ring	0.60	0.46	0.94	96.32
Pinky	0.68	0.60	1.04	93.76
Non-Thumb	0.38	0.29	0.82	98.12

Table 1: The touch position errors in the X direction.

5.3.2 Evaluation. We evaluate detected touch positions against touchpad ground truth. As a result, the mean X-axis errors for thumb, index, middle, ring, and pinky fingers are 1.27 cm, 0.30 cm, 0.36 cm, 0.60 cm, and 0.68 cm respectively. The mean Y-axis errors are 2.62 cm, 2.06 cm, 1.54 cm, 1.63 cm, and 1.54 cm respectively, which are substantially larger due to perspective compression in the front-facing view. Therefore, we use only horizontal (X-axis) positions in the touch model to improve robustness.

The X-axis touch position errors of each finger are shown in Table 1. We define a new metric, 1cm-accuracy, computed as the percentage of errors within 1 cm. Excluding thumbs, which are often occluded and mainly used for space input, the mean X error is 0.38 cm with a 1cm-accuracy of 98.12%. The touch positions provide more reliable spatial information beyond finger identities.

5.4 Text Input Decoding

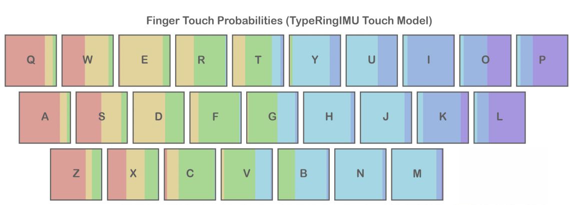


Figure 6: The touch finger distribution. Within each key area, each color represents a finger identity, and the area of the color indicates the finger touch probability.

5.4.1 Implementation. To implement the augmented Bayesian decoder (Section 3.4.1), we estimate letter-conditioned touch models from the typing dataset, including the finger identity distribution (Figure 6) and the touch position distribution (Figure 7).

For TypeRingIMU, we compute the conditional probability of finger identities given input letters. We found the typing behavior of most users deviates significantly from the standard fingering and exhibit substantial cross-user variability. Therefore, unlike previous work (e.g., PinchType [9], TypeNet [32], TapType [44]), we do not enforce standard fingering for text input. Instead, we derive a 32-user generic finger probability distribution and gradually personalize it based on individual user input.

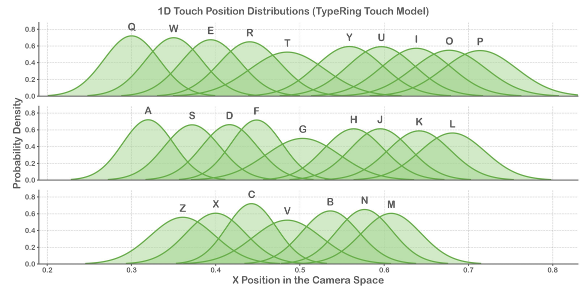


Figure 7: The 1D touch position distribution. Touches corresponding to a letter are modeled as a 1D Gaussian distribution. Letters within each row are separated for better clarity.

For TypeRing, we model the touch position of each letter as a 1D Gaussian distribution along the X axis in the camera space. By incorporating continuous touch positions, TypeRing further relaxes fingering constraints. The absolute touch model is also combined with the relative model following ResType [26]. The optimal weights are set to 0.4 (absolute) and 0.6 (relative). For TypeRingIMU, we use only the absolute model, as the relative finger model does not improve decoding accuracy in our evaluation.

System	Top-1 (%)	Top-5 (%)	Top-12 (%)
TypeRingIMU-G	73.25	92.00	96.00
TypeRingIMU-P	81.65	95.50	97.91
TypeRing-G	80.83	95.24	97.17
TypeRing-P	88.48	97.14	98.09

Table 2: The top-N text decoding accuracy of TypeRingIMU and TypeRing, with each using the general (-G) or personalized (-P) touch models respectively.

5.4.2 Evaluation. We replay the full typing process of the dataset and evaluate word-level decoding accuracy using both general (-G) and personalized (-P) touch models. As shown in Table 2, TypeRing consistently outperforms TypeRingIMU, demonstrating the benefit of incorporating touch positions. Personalization further improves decoding accuracy for both systems.

6 Study Two: Evaluation

We conduct a text input user study to evaluate the real-time performance of our method, including a main study to assess the typing performance and a follow-up study to evaluate robustness.

6.1 Design and Procedure

The user study aims to evaluate the real-time typing performance of TypeRingIMU and TypeRing and also contextualize them with prior surface typing methods. The apparatus was identical to the data collection setup (Section 4.1). Ground truth sensors are retained only for evaluation, but are not involved in the text input algorithms. Wrist postures during typing are not restricted. Participants are allowed to type and rest freely following their own habits.

6.1.1 Interaction. During the experiment, an English phrase is displayed on the laptop screen. The user wears two IMU rings on index fingers and performs ten-finger surface typing. Text input is performed using eight non-thumb fingers to enter letters and thumbs to enter spaces. Before a word is completed, the input is displayed as asterisks since the exact tapped letters are not yet determined. After a space is entered, the system predicts the top-12 candidate words using the Bayesian decoder, with the top candidate automatically selected and replacing the asterisks. Users can switch candidates using the right-hand ring touchpad, while the left-hand touchpad performs word deletion. Input errors are not explicitly modeled. Users can correct them by deleting the word and input again. Details of the interaction loop are shown in Appendix A.2.

6.1.2 Procedure. We recruited 16 participants from campus (aged 21-31, M=24.8, SD=2.4). The study adopted a within-subject design. To obtain users' baseline typing performance, participants first completed transcription tasks on smartphones, tablets, and physical keyboards (order counterbalanced). For each device, participants transcribed 10 English phrases randomly sampled from the MacKenzie phrase set [30] as quickly and accurately as possible.

After that, we evaluated the text input performance of TypeRingIMU and TypeRing (order counterbalanced). Each system consisted of six sessions of text transcription tasks, with 10 phrases per session. For each system, the touch models (finger identity model for TypeRingIMU, and touch position model for TypeRing) are gradually personalized across the six sessions. The first session used the general touch model learned from the typing dataset, while subsequent sessions incorporated personalized models derived from participants' previous sessions through linear weighting. The weights of the personalized models are set to 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0, allowing the decoder to progressively adapt to each user's typing behavior. At the end of the experiment, participants rated all input methods and provided subjective feedback.

6.2 Evaluation Results

Here we report the input performance of TypeRingIMU and TypeRing. Statistical analyses are conducted using repeated measures ANOVA (parametric) or Wilcoxon signed-rank tests (non-parametric), with Bonferroni correction applied as appropriate. Significance levels are annotated with '–', '$*$', '$**$', and '$***$', representing $p \geq 0.05$, $p < 0.05$, $p < 0.01$, and $p < 0.001$ respectively.

6.2.1 Text Input Speed. The text input speed is measured in words per minute as $WPM = \frac{|L| - 1}{T} \times 60 \times \frac{1}{5}$ [1], where L is the input character count, T is the input time in seconds. As shown in Figure 8, both systems achieve a satisfying input speed, and TypeRing significantly outperforms TypeRingIMU across all sessions. After practice (Session 6), TypeRingIMU reaches 22.72 ± 7.07 WPM, while TypeRing achieves 28.64 ± 8.10 WPM, improving typing speed by 26.1% ($p < 0.01$). Since participants are new to the systems, these results may not represent the ceiling performance.

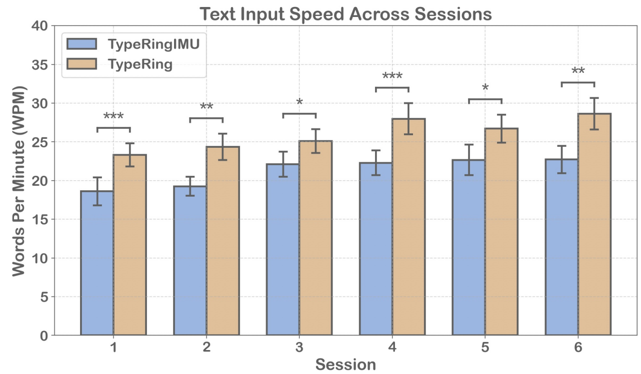


Figure 8: The text input speed of TypeRingIMU and TypeRing across sessions. Error bars indicate standard error.

For comparison, participants' input speeds on smartphones, tablets, and physical keyboards are 37.06 ± 8.80, 32.03 ± 7.14, and 45.11 ± 12.84 WPM respectively. At session 6, the relative speeds of TypeRingIMU compared to them are 61.3%, 70.9%, and 50.4%, and the relative speeds of TypeRing are 77.3%, 89.4%, and 63.5%. The relative speed of TypeRing to tablets is the highest, indicating its potential as an alternative to them.

Table 3 compares our system with prior surface typing methods. To eliminate participant bias, we primarily compare the relative typing speed, which is defined as the input speed of each work, compared with the reported input speed on physical keyboards of their users. TypeRingIMU substantially outperforms TapType [44] with a similar sensing modality (dual IMUs), and achieves relative speed comparable to camera-based systems such as TouchInsight [45] and StegoType [36]. While touchscreen-based methods (e.g., TOAST [40] and ResType [26]) achieve higher speeds due to precise touch sensing, they cannot support typing on common flat surfaces. For QwertyRing [13], the ring-based typing design is similar to ours, but the IMU ring is worn on the middle phalanx of the index finger, which is closer to the fingertip thus reducing the sensing difficulty. It is designed for one-finger touch typing, which also differs from our ten-finger typing scenario. Overall, our system provides a favorable trade-off between sensing cost and typing performance for mobile surface typing scenarios.

6.2.2 Errors and Correction Effort. We measure the word-level error rates using the uncorrected error rate (UER) and the corrected error rate (CER). UER is defined as the number of incorrect words after the input process over the total number of words. Across all sessions, the UER of both systems remains low (around 2-3%), with no significant differences between methods. At Session 6, the UERs of TypeRingIMU and TypeRing are $2.04\% \pm 2.85\%$ and $1.87\% \pm 2.61\%$. The UERs of smartphones, tablets, and physical keyboards ($2.41\% \pm 3.51\%$, $2.96\% \pm 3.00\%$, and $3.30\% \pm 4.35\%$) are comparable, indicating that all methods achieve similar input quality. Compared with prior work 3, the UER of our method still remains low and acceptable.

CER measures the effort required for correcting errors during typing. Corrections occur through either word deletion or candidate switching. Across all sessions, TypeRing shows significantly lower deletion, selection, and overall CER than TypeRingIMU ($p < 0.05$,

Name	Year	Sensing Modality	Typing Mode	Absolute Speed	Relative Speed	UER
TOAST [40]	2018	large touchscreen	ten finger	44.6 WPM	65.2%	-
QwertyRing [13]	2020	1 IMU ring (middle phalanx)	index finger	20.59 WPM	-	1.50%
TapType [44]	2022	2 IMU wristbands	ten finger	19.2 WPM	25.8%	-
ResType [26]	2023	Sensel Morph touchpad	ten finger	41.26 WPM	86.7%	1.55%
TouchInsight [45]	2024	Meta Quest hand tracking	ten finger	37.0 WPM	54.8%	2.9%
StegoType [36]	2024	Meta Quest hand tracking	ten finger	42.4 WPM	56.9%	7.0%
TypeRingIMU (Ours)	2025	2 IMU rings (proximal phalanx)	ten finger	22.72 WPM	50.4%	2.04%
TypeRing (Ours)	2025	2 IMU rings + 1 RGB camera	ten finger	28.64 WPM	63.5%	1.87%

Table 3: Usage and typing speed compared with related work. "-" means not reported.

$p < 0.01$, and $p < 0.001$, respectively). At Session 6, the deletion rates of TypeRingIMU and TypeRing are 17.70%±8.20% and 13.10%± 9.21%; the selection rates of them are 13.82% ± 7.22% and 5.34% ± 5.26%; and the CERs of them are 31.52%±11.88% and 18.44%±10.60%. The results indicate that TypeRing substantially reduce the effort required for error correction.

For conventional devices, we only report deletion rates, as they do not support candidate switching. They are computed in word-level. Continous backspaces are counted only once to be more comparable with our method. The deletion rates of smartphones, tablets, and physical keyboards are 12.63%±10.99%, 11.32%±11.62%, and 12.97% ± 5.77% respectively, which are comparable to TypeRing at Session 6 (13.10%).

To further understand the reduction in correction effort, we analyze the time distribution of the typing process at Session 6, normalized to inputting 10 words (50 characters). TypeRingIMU requires 22.56±6.40 s for input, 3.58±2.33 s for deletion, and 3.47±3.09 s for candidate selection, while TypeRing requires 19.07 ± 3.88 s, 2.33 ± 1.55 s, and 1.18 ± 1.37 s respectively. TypeRing reduces input, deletion, and selection time by 15.5% ($p < 0.01$), 34.9% ($p < 0.05$), and 66.0% ($p < 0.001$), indicating that the speed improvement mainly comes from reduced time spent on word correction operations, consistent with the CER results.

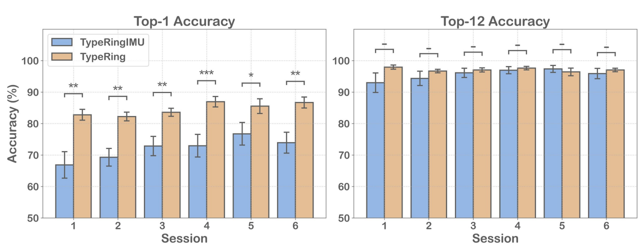


Figure 9: Top-1 and top-12 decoding accuracies of our method. Error bars indicate standard error.

6.2.3 Decoding Accuracy. We visualize the top-1 and top-12 decoding accuracies in Figure 9. At Session 6, the top-1 accuracies of TypeRingIMU and TypeRing are 73.92%±13.26% and 86.69%±6.97%. Across all sessions, TypeRing achieves significantly higher top-1 accuracy than TypeRingIMU. The top-12 accuracies of the two systems are both high (around 96-97%), with no significant difference. These results show that incorporating touch positions in TypeRing substantially improves text decoding accuracy, consistent with the offline evaluation in Section 5.4.2.

To understand why TypeRing improves decoding accuracy, we analyze the relationship between decoding accuracy and users' typing fingerings. We use the conditional probability of a finger identity f given a letter c, denoted as $P(f|c)$, to represent the typing fingering. We define a new metric, **fingering coefficient** ϕ, to measure the similarity between a user's fingering and the standard QWERTY fingering. It is computed as the average Bhattacharyya coefficient [51] between the user-specific and standard finger distributions over all letters:

$$\phi = \frac{1}{|C|} \sum_{c \in C} \sum_{f \in \mathcal{F}} \sqrt{P_{user}(f|c) \cdot P_{std}(f|c)} \in [0, 1] \quad (2)$$

where C is the English alphabet set with size 26, and $\mathcal{F}$ is the finger identity set with size 6. P_{std} and P_{user} are the standard fingering and the user-specific fingering, respectively.

Figure 10 shows the top-1 decoding accuracy of TypeRingIMU and TypeRing at Session 1 (since no personalization is applied) as a function of ϕ. For TypeRingIMU, decoding accuracy is strongly positively correlated with ϕ, indicating that the system performs better when users follow the standard fingering. This is expected because TypeRingIMU relies solely on finger identities for text decoding. In contrast, TypeRing shows little dependence on ϕ. Although participants exhibit diverse fingerings (ϕ ranging from 0.369 to 0.952, M=0.763, SD=0.177), their touch positions still generally conform to the QWERTY layout due to typing muscle memory (Figure 7). By incorporating touch positions, TypeRing decouples text decoding from finger identities and significantly improves decoding accuracy, particularly for users with non-standard fingerings.

6.2.4 Subjective Ratings. After the experiment, participants provided subjective feedback on our systems using a 7-point Likert scale across multiple aspects (Figure 11, modified from NASA-TLX [14] and SUS [4]). For consistency in comparison, higher scores always indicate a better experience across all aspects. Both systems received positive ratings, and TypeRing significantly outperformed TypeRingIMU in most aspects except latency, which showes no difference since both systems share the same touch event detection module. These results indicate an overall improvement in user experience with TypeRing.

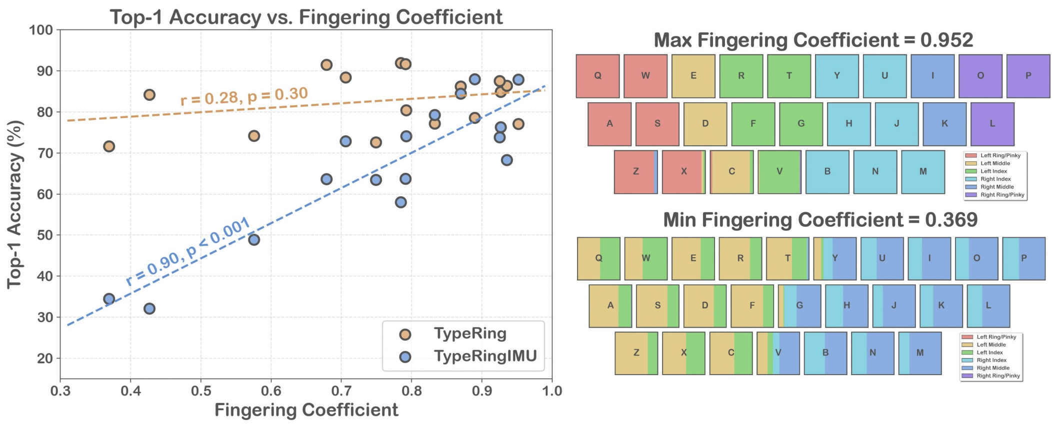


Figure 10: The top-1 decoding accuracy of our methods over the fingering coefficient. Each point represents one user at session 1. The trend lines are fitted using the least squares method, with the Pearson correlation coefficient and p-value annotated (left). The fingerings of the users with the highest and lowest coefficients are visualized (right).

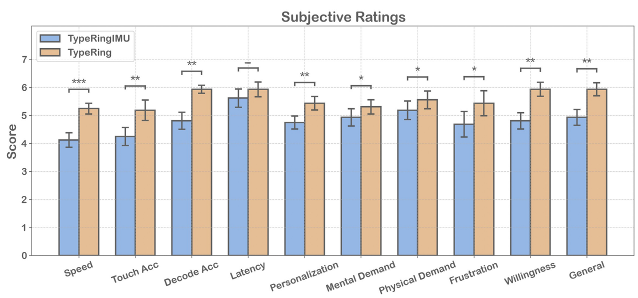


Figure 11: The subjective feedback: the 7-point Likert-scale ratings (higher is better) of our methods over 10 aspects. Error bars indicate standard deviation.

We further asked participants to compare our systems relative to smartphones, tablets, and physical keyboards, and provide a subjective score. It is defined as the general experience of our method divided by that of other devices. The scores of TypeRingIMU compared to smartphones, tablets, and physical keyboards are 0.81 ± 0.33, 1.14 ± 0.36, and 0.65 ± 0.23. The scores of TypeRing are 0.92 ± 0.35, 1.33 ± 0.39, and 0.75 ± 0.25. TypeRing consistently achieved higher subjective scores than TypeRingIMU ($p < 0.01$ for smartphone and tablet, $p < 0.001$ for physical keyboard). Both systems are rated higher than tablets and close to smartphones, suggesting strong potential as alternatives for mobile text input. Interestingly, subjective ratings are higher than the relative typing speeds reported earlier (e.g., the relative speed of TypeRing w.r.t. physical keyboards is 63.5%, while the relative experience score is 75.3%), indicating that users value factors beyond speed.

6.2.5 Robustness. To further evaluate the robustness of our systems under practical deployment variations, we conducted a follow-up user study. We re-invited all participants from the main study, among whom 10 were available to participate. To isolate learning effects, personalization was disabled and each condition was tested only for one session, while other procedures remained identical.

First, we evaluated TypeRingIMU on three surface materials: a table (hard), a mousepad (soft), and the user's thigh (soft and uneven). The typing speeds are 23.15 ± 6.93, 23.05 ± 5.86, and 17.34 ± 6.81 WPM. The UERs are $2.9\% \pm 3.3\%$, $2.0\% \pm 2.7\%$, and $2.2\% \pm 2.0\%$. Typing on the thigh is moderately slower due to the softer surface and less stable typing posture (model not finetuned). Nevertheless, the thigh condition still achieves about 75% of the table speed, demonstrating reasonable robustness across surfaces. To improve performance on soft surfaces, the touch detection model should be finetuned with more data, which is left for future work.

Second, we evaluated TypeRing under free typing on the table surface with three camera pitch angles (upper, middle, and lower views). The typing speeds are 22.58 ± 8.73, 23.59 ± 9.96, and 23.69 ± 9.01 WPM. The UERs are $2.1\% \pm 3.1\%$, $2.9\% \pm 3.7\%$, and $2.4\% \pm 2.6\%$. No significant differences are found between conditions, showing robustness across camera pitch views. The slightly lower speed compared to the main study is due to the absence of personalization and fewer test sessions. These results suggest that our method remains usable across diverse real-world conditions.

7 Discussion

7.1 Muscle Memory during Typing

Surface typing require users to type blindly without visual feedback, which inevitably relies on QWERTY muscle memory. We further analyze the nature of this muscle memory by decomposing it into two components: the mapping from letters to (1) finger identities and (2) touch positions. Our experiments show that while touch position distributions are relatively consistent across users, finger identity distributions vary greatly (Figure 10). This explains why TypeRingIMU, which relies on finger identities, performs better for users with standard fingerings, whereas TypeRing achieves consistently higher decoding accuracy by leveraging touch positions. Future interfaces may further assist users by providing lightweight visual cues of the keyboard layout during typing.

7.2 Practical Constraints of the Camera

Our design leverages an RGB camera to obtain touch positions, which plays a key role in removing fingering constraints. However, it also introduces several practical limitations. First, our current implementation models touch positions directly in the image space, requiring users to keep their hands near the center of the camera view. Although the follow-up study shows robustness to vertical angle variations, horizontal hand drift and rotation may still affect performance, and may requires additional calibration. Second, for mobile devices with front-facing cameras, capturing the hands typically requires placing the device at a relatively upright angle or using cameras with a wider FOV, which may not always be supported by current hardware. We emphasize that the current system mainly serves as a proof-of-concept demonstrating the benefits of optical-inertial fusion for surface typing, rather than a fully deployable solution. Future work could address these limitations by detecting overall hand positions and normalizing touch coordinates relative to the hand. Lastly, CV-based method may be affected by lighting conditions, which is not covered in our current study. Our algorithm only uses the CrCb channels of the YCrCb color space, which is naturally insensitive to brightness variations, but still need additional validation in real-world scenarios.

7.3 Input Capabilities

Our system recognizes ten-finger touches and decodes intended text. Currently, the input capability is limited to 10K English words and spaces, and does not yet support out-of-vocabulary (OOV) words, punctuation, or capitalization. We acknowledge these limitations and emphasize that our primary contribution lies in demonstrating the sensing capability: reliably detecting finger identities and touch positions and establishing their correlation with users' textual intent. This setting is also common in prior work on novel text-input techniques [13, 26, 32, 44], enabling fair comparison. We further position ring-based typing as a lightweight and portable input method better suited for short, opportunistic interactions (e.g., quick notes, short replies, or command execution) rather than replacing physical keyboards. In such scenarios, conveying semantic intent is often more important than producing fully formatted text. Punctuation and capitalization could be automatically inferred by LLMs, allowing users to focus on expressing their intended content.

7.4 Application Scenarios and Trade-offs

TypeRingIMU and TypeRing provide complementary benefits for different usage scenarios. TypeRingIMU emphasizes portability, requiring only two rings, making it suitable for space-constrained, low-power, or privacy-sensitive settings. Its minimal hardware requirements also facilitate integration with wearable devices such as AR glasses or earbuds for quick text entry with audio or visual feedback. In contrast, TypeRing achieves higher accuracy and speed by incorporating touch-position sensing, making it better suited for heavier typing tasks than TypeRingIMU. It effectively expands the input space of smartphones and tablets without requiring a physical keyboard and is particularly beneficial for users with non-standard fingering habits. Overall, the two systems represent a trade-off between portability and sensing capabilities. Future systems could dynamically switch between IMU-only and optical-inertial fusion modes based on device capabilities, user preferences, privacy constraints, and task demands.

7.5 Limitations and Future Work

Despite promising results, several limitations remain and provide directions for future work. Beyond the aspects discussed earlier, we highlight three additional areas for improvement. First, the robustness boundaries of the system in real-world environments remain to be systematically evaluated. Factors such as lighting variations, vibrations in mobile contexts (e.g., transportation), and deployment on existing devices may influence performance. These conditions involve complex interactions that are beyond the scope of this work and cannot be exhaustively examined here. Second, the sensing pipeline can still be improved. The current system uses a 100 ms IMU window for touch detection, which may fail to distinguish successive touches from the same hand within that interval. Improving temporal resolution or adopting adaptive window techniques could enable more precise detection. Third, the current sensing pipeline has an end-to-end latency of approximately 100 ms (from touch to character rendering, measured by an 120 FPS camera). Specifically, BLE transmission takes ~10 ms; touch peaks appearing near the center of a 20-frame IMU window takes ~40 ms (5 ms per frame); window aggregation (4 predictions, 8 frames) takes ~40 ms; touch detection model inference takes ~5 ms, and word decoding ~2 ms. Although participants did not report noticeable delays (Section 6.2.4), future work could further optimize the data pipeline, for example by running the touch detection model directly on the ring. Lastly, the top-1 accuracy of TypeRing (88.5%) still has room for improvement. Future work can leverage large language models to further improve decoding accuracy.

8 Conclusion

We present TypeRing, a dual-IMU-ring system that enables ten-finger typing on any flat surface. The system includes two variants. TypeRingIMU relies solely on two IMU rings to detect ten-finger touches and identities, enabling portable surface typing with minimal sensing cost. TypeRing further enhances it through optical-inertial fusion, leveraging a front-facing RGB camera to detect touch positions and remove fingering constraints, thereby improving input naturalness and decoding accuracy. Our systems achieve efficient surface typing performance, with TypeRing reaching 28.6 WPM (63.5% of physical keyboards) and consistently outperforming IMU-only solutions. Overall, our work demonstrates that combining lightweight inertial sensing with event-triggered vision provides an effective approach for enabling practical ten-finger surface typing. The two variants offer complementary trade-offs between portability and sensing richness, allowing flexible deployment across different mobile scenarios.

Acknowledgments

This work is supported by National Key R&D Program of China under Grant No. 2024YFB2808800 & 2024YFB2808803.

References

[1] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In *2009 IEEE Toronto international conference science and technology for humanity (TIC-STH)*. IEEE, 100–105.

[2] Jutamanee Auysakul, He Xu, and Vishwanath Pooneeth. 2018. A hybrid motion estimation for video stabilization based on an IMU sensor. *Sensors* 18, 8 (2018), 2708.

[3] Michael Bloesch, Sammy Omari, Péter Fankhauser, Hannes Sommer, Christian Gehring, Jemin Hwangbo, Mark A Hoepflinger, Marco Hutter, and Roland Siegwart. 2014. Fusion of optical flow and inertial measurements for robust egomotion estimation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3102–3107.

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

[5] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. 2019. Selective sensor fusion for neural visual-inertial odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10542–10551.

[6] Fabrizio Cutolo, Virginia Mamone, Nicola Carbonaro, Vincenzo Ferrari, and Alessandro Tognetti. 2020. Ambiguity-free optical–inertial tracking for augmented reality headsets. *Sensors* 20, 5 (2020), 1444.

[7] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text entry in virtual reality: A comprehensive review of the literature. In *International Conference on Human-Computer Interaction*. Springer, 419–437.

[8] Naser El-Sheimy and Ahmed Youssef. 2020. Inertial sensors technologies for navigation applications: State of the art and future trends. *Satellite navigation* 1, 1 (2020), 2.

[9] Jacqui Fashimpaur, Kenrick Kin, and Matt Longest. 2020. Pinchtype: Text entry for virtual and augmented reality using comfortable thumb to fingertip pinches. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*. 1–7.

[10] Leah Findlater, Jacob O Wobbrock, and Daniel Wigdor. 2011. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2453–2462.

[11] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces*. 194–195.

[12] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and low-latency sensing of touch contact on any surface with finger-worn IMU sensor. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 1059–1070.

[13] 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.

[14] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[15] Abdirahman Osman Hashi, Siti Zaiton Mohd Hashim, and Azurah Bte Asamah. 2024. A systematic review of hand gesture recognition: An update from 2018 to 2024. *IEEE Access* 12 (2024), 143599–143626.

[16] Changyu He, Peter Kazanzides, Hasan Tutkun Sen, Sungmin Kim, and Yue Liu. 2015. An inertial and optical sensor fusion approach for six degree-of-freedom pose estimation. *Sensors* 15, 7 (2015), 16448–16465.

[17] Hailey N Hicks, Howard Chen, and Sara A Harper. 2025. Sensor Fusion for enhancing motion capture: Integrating optical and inertial motion capture systems. *Sensors* 25, 15 (2025), 4680.

[18] Olga Iarygina and Aske Mottelson. 2025. A Systematic Review and Meta-analysis of Text Entry Studies. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–14.

[19] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery* 34, 6 (2020), 1936–1962.

[20] Utkarsh Kunwar, Sheetal Borar, Moritz Berghofer, Julia Kylmälä, Ilhan Aslan, Luis A Leiva, and Antti Oulasvirta. 2022. Robust and deployable gesture recognition for smartwatches. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*. 277–291.

[21] Yongseok Lee, Wonkyung Do, Hanbyeol Yoon, Jinuk Heo, WonHa Lee, and Dongjun Lee. 2021. Visual-inertial hand motion tracking with robustness against occlusion, interference, and contact. *Science Robotics* 6, 58 (2021), eabe1315.

[22] Chen Li, Li Song, Shuai Chen, Rong Xie, and Wenjun Zhang. 2022. Deep online video stabilization using IMU sensors. *IEEE Transactions on Multimedia* 25 (2022), 2047–2060.

[23] Juan Li, Maarten Slembrouck, Francis Deboeverie, Ana M Bernardos, Juan A Besada, Peter Veelaert, Hamid Aghajan, José R Casar, and Wilfried Philips. 2016. Handheld pose tracking using vision-inertial sensors with occlusion handling. *Journal of Electronic Imaging* 25, 4 (2016), 041012–041012.

[24] Zisu Li, Chen Liang, Yuntao Wang, Yue Qin, Chun Yu, Yukang Yan, Mingming Fan, and Yuanchun Shi. 2023. Enabling voice-accompanying hand-to-face gesture recognition with cross-device sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.

[25] Zhuojun Li, Lubin Wang, Chun Yu, Chang Liu, Mingyuan Du, Weinan Shi, and Yuanchun Shi. 2026. 3DRing: Enabling Low-Cost 3D Hand Position Tracking by Fusing Inertial and Low-Framerate Optical Sensing. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*. 1–18.

[26] Zhuojun Li, Chun Yu, Yizheng Gu, and Yuanchun Shi. 2023. ResType: Invisible and Adaptive Tablet Keyboard Leveraging Resting Fingers. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[27] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. 2021. DualRing: Enabling subtle and expressive hand interaction with dual IMU rings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–27.

[28] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. 2017. Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–24.

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

[30] I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*. 754–755.

[31] Oleg Makaussov, Mikhail Krassavin, Maxim Zhabinets, and Siamac Fazli. 2020. A low-cost, IMU-based real-time on device gesture recognition glove. In *2020 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 3346–3351.

[32] Ben Maman and Amit Bermano. 2022. TypeNet: Towards Camera Enabled Touch Typing on Flat Surfaces through Self-Refinement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1140–1149.

[33] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid touch interaction in virtual reality using wearable sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 519–528.

[34] Chaithanya Kumar Mummadi, Frederic Philips Peter Leo, Keshav Deep Verma, Shivaji Kasireddy, Philipp M Scholl, Jochen Kempfle, and Kristof Van Laerhoven. 2018. Real-time and embedded detection of hand gestures with an IMU-based glove. In *Informatics*, Vol. 5. MDPI, 28.

[35] Daniel R Rashid and Noah A Smith. 2008. Relative keyboard input system. In *Proceedings of the 13th international conference on Intelligent user interfaces*. 397–400.

[36] Mark Richardson, Fadi Botros, Yangyang Shi, Pinhao Guo, Bradford J Snow, Linguang Zhang, Jingming Dong, Keith Vertanen, Shugao Ma, and Robert Wang. 2024. Stegotype: Surface typing from egocentric cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[37] Mark Richardson, Matt Durasoff, and Robert Wang. 2020. Decoding surface touch typing from hand-tracking. In *Proceedings of the 33rd annual ACM symposium on user interface software and technology*. 686–696.

[38] Fendy Santoso, Matthew A Garratt, and Sreenatha G Anavatti. 2016. Visual–inertial navigation systems for aerial robotics: Sensor fusion and technology. *IEEE Transactions on Automation Science and Engineering* 14, 1 (2016), 260–275.

[39] Xiyuan Shen, Chun Yu, Xutong Wang, Chen Liang, Haozhan Chen, and Yuanchun Shi. 2024. MouseRing: Always-available Touchpad Interaction with IMU Rings. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–19.

[40] Weinan Shi, Chun Yu, Xin Yi, Zhen Li, and Yuanchun Shi. 2018. TOAST: Ten-finger eyes-free typing on touchable surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–23.

[41] Michael J Smith, Alexander Boxerbaum, Gilbert L Peterson, and Roger D Quinn. 2010. Electronic image stabilization using optical flow with inertial fusion. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1146–1153.

[42] Paul Streli. 2025. *Probabilistic Inference for Touch Input.* Ph. D. Dissertation. ETH Zurich.

[43] Paul Streli, Rayan Armani, Yi Fei Cheng, and Christian Holz. 2023. HOOV: Hand out-of-view tracking for proprioceptive interaction using inertial sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16.

[44] 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.

[45] Paul Streli, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. 2024. TouchInsight: Uncertainty-aware Rapid Touch and Text Input for Mixed Reality from Egocentric Vision. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–16.

[46] Jiankai Tang, Zhe He, Mingyu Zhang, Wei Geng, Chengchi Zhou, Weinan Shi, Yuanchun Shi, and Yuntao Wang. 2025. {\tau}-Ring: A Smart Ring Platform for Multimodal Physiological and Behavioral Sensing. *arXiv preprint arXiv:2508.00778* (2025).

[47] Inc. Tap Systems. 2025. Tap Strap 2. https://www.tapwithus.com/product/tap-strap-2/
[48] Rayane Tchantchane, Hao Zhou, Shen Zhang, and Gursel Alici. 2023. A review of hand gesture recognition systems based on noninvasive wearable sensors. *Advanced intelligent systems* 5, 10 (2023), 2300207.
[49] Tomasz Ursel and Michał Olinski. 2021. Displacement estimation based on optical and inertial sensor fusion. *Sensors* 21, 4 (2021), 1390.
[50] Zeyu Wang, Ruotong Yu, Xiangyang Wang, Jiexin Ding, Jiankai Tang, Jun Fang, Zhe He, Zhuojun Li, Tobias Röddiger, Weiye Xu, et al. 2025. Computing with Smart Rings: A Systematic Literature Review. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 9, 3 (2025), 1–54.
[51] Wikipedia. 2025. Bhattacharyya distance. https://en.wikipedia.org/wiki/Bhattacharyya_distance
[52] Vasco Xu, Brian Chen, Eric J Gonzalez, Andrea Colaço, Henry Hoffmann, Mar Gonzalez-Franco, and Karan Ahuja. 2026. SurfaceXR: Fusing Smartwatch IMUs and Egocentric Hand Pose for Seamless Surface Interactions. *IEEE Transactions on Visualization and Computer Graphics* (2026).
[53] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, et al. 2022. Enabling hand gesture customization on wrist-worn devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19.
[54] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Vladislav Golyanik, Shaohua Pan, Christian Theobalt, and Feng Xu. 2023. Egolocate: Real-time motion capture, localization, and mapping with sparse body-mounted sensors. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–17.
[55] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–19.
[56] Dian Zhang, Zexiong Liao, Wen Xie, Xiaofeng Wu, Haoran Xie, Jiang Xiao, and Landu Jiang. 2021. Fine-grained and real-time gesture recognition by using IMU sensors. *IEEE Transactions on Mobile Computing* 22, 4 (2021), 2177–2189.
[57] Mingrui Ray Zhang, Shumin Zhai, and Jacob O Wobbrock. 2022. TypeAnywhere: A QWERTY-based text entry solution for ubiquitous computing. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.
[58] Shiwen Zhao, Brandt Westing, Shawn Scully, Heri Nieto, Roman Holenstein, Minwoo Jeong, Krishna Sridhar, Brandon Newendorp, Mike Bastian, Sethu Raman, et al. 2019. Raise to speak: an accurate, low-power detector for activating voice assistants on smartwatches. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2736–2744.

A Implementation Details

A.1 Touch Detection Model

The model structure is illustrated in Figure 12. To capture rich motion features in finger touches from various dimensions, we adopt multi-scale convolutional kernels concurrently in each layer, inspired by the IncaptionTime model [19]. Our model is built on the customized Inception block (Figure 12b), which consists of four branches with different convolutional kernels (1, 3, 5) and a max-pooling layer. The full model (Figure 12a) uses 6 Inception blocks with 3 hidden sizes, 2 max-pooling layers, one conv layer to extract motion features, and outputs finger identity probabilities through an MLP head. Our model contains only 375K parameters, suitable for deployment on mobile devices.

A.2 Interaction Loop

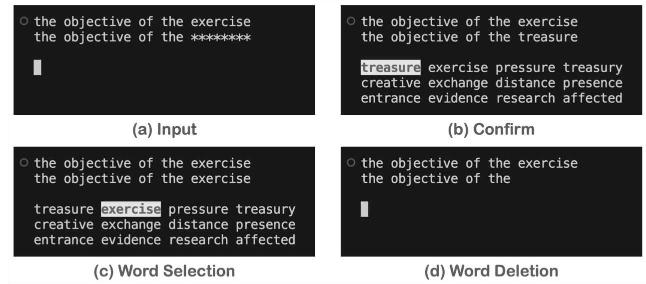


Figure 13: The interaction UI of TypeRing typing process. (a) Input letters, shown in asterisks. (b) Confirm a word by input a space. (c) Word selection with right ring touchpad. (d) Word deletion with left ring touchpad.

TypeRing leverages ten-finger touches and four ring touchpad gestures (left tap, left long-tap, right tap, right long-tap) to support the complete typing process, including letter input, word confirmation, word selection, and word deletion. As shown in Figure 13, the UI shows a target phrase, an input area, and a candidate matrix from top to bottom. Users can input letters with non-thumb touches, and an asterisk will be displayed for each touch. During input, users can use left tap to delete one asterisk, and left long-tap to delete all asterisks in the input area. After a word is input, users can confirm it by inputting a space, which will trigger word prediction and display the candidate matrix. In the mean time, the asterisks will be replaced by the top-1 candidate, and users can switch candidates with the right ring touchpad. Right tap will switch a column, and right long-tap will switch a row. If input errors occur, users can also delete the word by either left tap or left long-tap. Deleting a single letter in a predicted word is not supported.

Description	Count	Percentage(%)	Description	Count	Percentage(%)
Pinch with index finger.	5104	5.55	Snap fingers.	797	0.87
Pinch with middle finger.	4704	5.11	Push hand forward.	699	0.76
Pinch with ring finger.	3741	4.06	Clench hand.	880	0.96
Pinch with pinky finger.	3741	4.06	Flip hand.	401	0.44
Pinch down with index finger.	1998	2.17	Rotate wrist.	801	0.87
Pinch up with index finger.	2005	2.18	Draw circle with index finger.	801	0.87
Tap ring touchpad.	8663	9.41	Thumb up.	400	0.44
Touch ring touchpad down.	832	0.90	Thumb tapping index finger.	147	0.16
Touch ring touchpad up.	730	0.79	Bend index finger.	137	0.15
Slide on the ring.	2388	2.59	Slide on the plane.	447	0.49
Tap in air.	7588	8.24	Spread hand.	299	0.33
Flick index finger.	4700	5.11	Move finger up from plane.	300	0.33
Shake hands.	7188	7.81	Throw something.	300	0.33
Wave hands.	1601	1.74	Rotate ring.	40	0.04
Clap hands.	848	0.92	Other negative gestures.	22573	32.34

Table 4: The non-surface-typing negative gestures collected in the study, which contains 92041 samples in total.

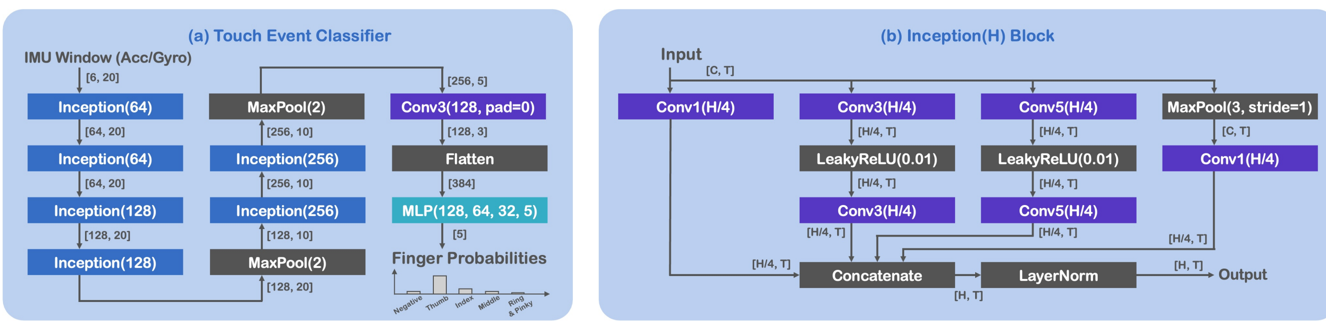


Figure 12: Touch event detection network structure. (a) the touch event detection and finger identification model, which is built on (b) the customized Inception block with four concurrent convolutional branches. The colored blocks indicate trainable layers, whose key hyperparameters are shown in the parentheses (e.g., hidden sizes for the Inception, Conv, and MLP layers).