Exploring Low-Occlusion Qwerty Soft Keyboard Using Spatial Landmarks

KE SUN, CHUN YU, and YUANCHUN SHI, Tsinghua University, China

The Qwerty soft keyboard is widely used on mobile devices. However, keyboards often consume a large portion of the touchscreen space, occluding the application view on the smartphone and requiring a separate input interface on the smartwatch. Such space consumption can affect the user experience of accessing information and the overall performance of text input. In order to free up the screen real estate, this article explores the concept of Sparse Keyboard and proposes two new ways of presenting the Qwerty soft keyboard. The idea is to use users' spatial memory and the reference effect of spatial landmarks on the graphical interface. Our final design *K3-SGK* displays only three keys while *L5-EYOCN* displays only five line segments instead of the entire Qwerty layout. To achieve this, we employ a user-centered computational design method: first study the reference effect of a single landmark key (line segment) from empirical data, then make assumptions to generalize the effect to multiple landmarks, and finally optimize the best designs. To make the text entry function more complete, we also design and implement gestural interactions for editing operations and non-alphabetical characters' input. User evaluation shows that participants can quickly learn how to type with *K3-SGK* and *L5-EYOCN*. After five 15-phrase typing sessions, participants achieve 88.1%–92.8% of the full Qwerty keyboard in terms of words per minute on the smartphone and 98.4%–99.1% on the smartwatch. The differences on character and word error rate between our keyboard designs and the full Qwerty keyboard are not significant. The results of out-of-vocabulary words input are also promising. In addition, participants can quickly recall the typing skills and maintain the input performance even after a few days. User feedbacks in real application contexts show that with the low occlusion keyboard, users can acquire more information and perform less scrolling on the smartphone and achieve a higher input efficiency on the smartwatch with a more fluent input experience.

CCS Concepts: • **Human-centered computing** → *Keyboards*; *User studies*; *User centered design*;

Additional Key Words and Phrases: Keyboard design, text entry, low-occlusion interface, spatial landmark, spatial reference, spatial memory

ACM Reference format:
Ke Sun, Chun Yu, and Yuanchun Shi. 2019. Exploring Low-Occlusion Qwerty Soft Keyboard Using Spatial Landmarks. *ACM Trans. Comput.-Hum. Interact.* 26, 4, Article 20 (June 2019), 33 pages.
https://doi.org/10.1145/3318141

This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001200, the Natural Science Foundation of China under Grant No. 61672314 and No. 61572276, Tsinghua University Research Funding No. 20151080408, and also by Beijing Key Lab of Networked Multimedia.
Authors' addresses: K. Sun, C. Yu, and Y. Shi, Tsinghua University, 30 Shuangqing Rd, Haidian Qu, Beijing Shi, China, 100084; emails: k-sun14@mails.tsinghua.edu.cn, chunyu@tsinghua.edu.cn, shiyc@tsinghua.edu.c.

1073-0516/2019/06-ART20 $15.00
https://doi.org/10.1145/3318141

1 INTRODUCTION

Current smart mobile devices come standard with a mini Qwerty soft keyboard. Thanks to the direct input nature of touch and probabilistic decoding algorithms, users type both fast and accurately on the touchscreen. However, the keyboard itself consumes a significant amount of the screen space (32%–36% on a smartphone and 45%–56% on a smartwatch in our survey of 10 commercial products) and separates the input space from the output space. On a smartphone, the keyboard squeezes the application view upwards or directly blocks part of the content, affecting the user experience of accessing information. For example, when chatting or composing emails, the user may need to switch frequently between typing and scrolling up/down on the application view (Figure 1(a)). Such issue is aggravated on the smartwatch because of the ultra-small screen. Once typing with the built-in IME of Android Wear and Tizen, a separate input interface with only an edit field plus the keyboard will be launched (Figure 1(b)). In this case, users lose all contexts and the input efficiency is likely to decrease. To tackle these problems, researchers have investigated typing on an totally invisible keyboard [53]. With a spatially adapted model, users achieved a promising input performance at 37.9 WPM (words per minute) and <3% WER (word error rate). However, it would be difficult for users to perceive the exact boundary of the keyboard and difficult to input a single letter or out-of-vocabulary (OOV) words.

In order to free up the screen real estate and at the same time guarantee the input precision, the article proposes and explores two novel visualizations of the standard Qwerty layout, which only show a few reference keys and line segments instead of the whole keyboard. Our designing idea is mainly motivated by the following two aspects.

First, keep the standard Qwerty layout than change it. Researchers have modeled the learnability of a keyboard layout exposing significant learning costs to changes [21]. The visual search time requires hours to decrease and level on a new keyboard that has little similarity with the Qwerty layout. Though several previous research on keyboard designs can alleviate the occlusion issue, they mostly adopt a different layout against Qwerty (e.g., condensing to a smaller size [26], rearranging to a circular form [36, 51], using background image as reference [4], and gesturing [7, 49]). This forces users to change their input habits resulting in a long training process and low typing speed. In our work, to keep users' input habits and minimize the performance degradation, we preserve the Qwerty layout and "directly" reduce its visualization.

Second, leverage the effect of spatial landmarks. Soft keyboard is a typical memory-based interface with a stable layout [8, 14, 41]. Through regular practice, users can anticipate the location of each key and type with little visual search or guidance. Previous research on spatial memory-based interfaces [40, 44] has demonstrated that the "landmarks," which are the location features over the graphical interface, can serve as a strong reference that helps users build up spatial memory and revisit an intended location quickly. We guess that this property can also be applied to the Qwerty soft keyboard. For example, with only the key *I* visible, we expect the user can easily know the location of its adjacent key *U* and key *O*. Therefore, studying the reference effect of spatial landmarks on the soft keyboard forms the basis of our keyboard designs.

Based on the structure of the Qwerty layout, we introduce two new ways of presenting the soft keyboard: the *K-Board* and the *L-Board* which use two different forms of spatial landmarks and reference strategies. The *K-Board* displays only a few (three) individual keys as the "landmark keys" instead of all 26 alphabetical characters. In this design, we see each key as the basic input unit and try to reduce the keyboard on this level. The *L-Board* on the other hand, displays only a few (five) line segments between adjacent keys. For example, given the dividing line between key *D* and key *F*, we expect users can accurately access key *D* and key *F*, and perhaps can also easily know the location of key *X* and key *C* below and key *R* right above.

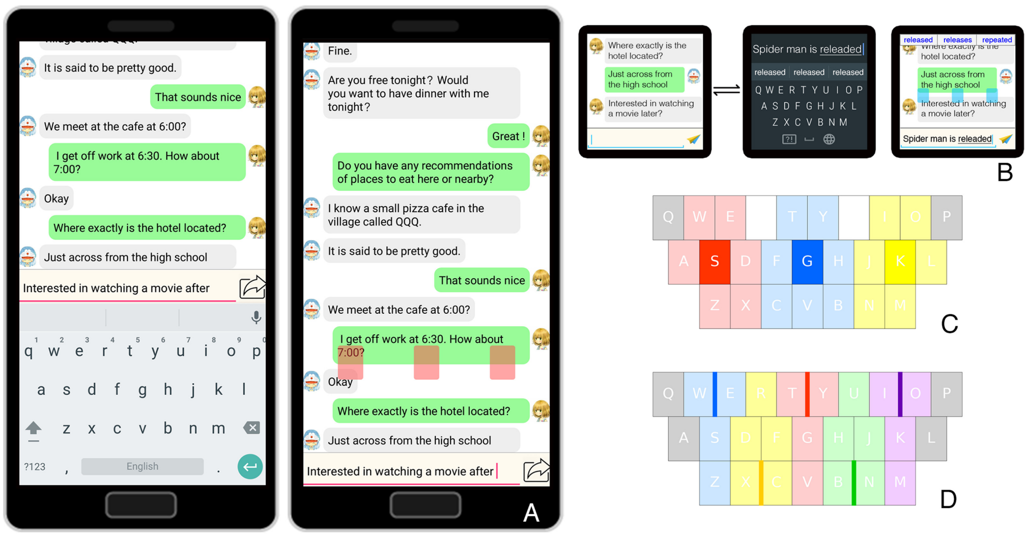

Fig. 1. Current soft keyboards on a mobile device separate the typing space and the application view, resulting in a substantial reduction in the screen real estate. This may affect the user experience of browsing information on a (A) smartphone, and slow down the input efficiency on a (B) smartwatch because of frequent interface switch. In this work, We use the reference effect of spatial landmarks to optimize low-occlusion Qwerty soft keyboard while maintaining a close typing performance to a full Qwerty keyboard. (C) and (D) show the two final designs: *K3-SGK* and *L5-EYOCN* as well as their expected reference ranges of each spatial landmark.

The key challenge is to figure out how many and which landmark keys/line segments should be displayed so that the user can orient themselves on the soft keyboard both globally (perceiving the entire typing area) and locally (knowing the exact location of each key). We adopt a computational design approach [33] to solve this problem. Given the number of visible landmarks as n, the best keyboard design maximizes the weighted input accuracy of the 26 letters, where the weight is the letter frequency. We start from the case when $n = 1$. Through user experiments, we collect substantial empirical data and study how a single visible landmark key/line segment affects the input accuracy of each letter. The results show that spatial and motor memory remain on a nearly invisible keyboard while the landmark provides a strong reference in the local area. However, the user's reference strategy and the effect region of a landmark key and of a landmark segment are significantly different. We then make two assumptions to generalize the reference effect to keyboards that have multiple landmarks when $n \geqslant 2$. This allows us to optimize the keyboard visualization in cases when any number of landmarks are visible.

After balancing the weighted input accuracy, the number of landmarks and user preferences, the final designs are: *K3-SGK* for *K-Board* and *L5-EYOCN* for *L-Board* (Figure 1). Consequently, *K3-SGK* takes up only 11.5% of the Qwerty area and *L5-EYOCN* occupies almost no space with only five line segments. To make the text entry function more complete, we also design and implement gestural interactions for basic editing operations and common non-alphabetical characters' input. We then carry out two user studies to examine the feasibility of the two proposed keyboard designs.

The first user study focuses on the typing performance and the learning process. Results demonstrate that with limited training, the input speed and accuracy of using *K3-SGK* and *L5-EYOCN* quickly raised and approached the performance of using a full Qwerty keyboard. After five 15-phrase typing sessions, participants were able to achieve 88.1%–92.8% of the full Qwerty keyboard in terms of WPM on the smartphone and 98.4%–99.1% on the smartwatch on dictionary-word phrase typing. We found no significant differences on character error rate (CER) and WER between our keyboard designs and the full Qwerty. On OOV-word phrase typing, the results were also promising. In addition, participants could quickly recall the typing skills and maintain the input performance even after a few days. The second user study examines the benefits and limitations of the low-occlusion keyboard in real application contexts through a comparison of *K3-SGK* with the full Qwerty keyboard. To the best of our knowledge, no research work has been able to reduce the display area of a Qwerty soft keyboard to the same level as we do, and at the same time achieve a close performance with a full Qwerty keyboard. The idea of using the reference effect of spatial landmarks and the optimizing procedure can also be applied to reducing other memory-based graphical interfaces.

2 RELATED WORK

Our work mainly builds on three aspects of prior research: typing behaviors on the soft keyboard, spatial reference effect on graphical user interfaces (GUI), and reduced keyboards on mobile devices.

2.1 Learning and Typing on a Soft Keyboard

Users' typing behavior on a soft (virtual) keyboard is a complex and composite process, but can be briefly modeled in two stages: (1) locating the target key, and then (2) move the finger/stylus to click the key. There are plenty of models in HCI research to describe both processes, for example, using Hick–Hyman Law [19, 20] for choice reaction time in the former process and Fitts's Law [12] for motor performance in the latter process. Based on the modeling of the motor performance, researchers have proposed new keyboard layouts against Qwerty by optimizing the movement time [34, 52], which technically enables a higher typing speed. However, these new layouts are rarely translated into practical use by users on account of high perceived learning effort in locating keys [21]. Empirical investigations showed that for novice the input speed on randomized layouts was only about 5.4 WPM [32].

For regular Qwerty soft keyboard users, long-term memory has been established [21]. With repeated use of the keyboard, the user continuously constructs the representation of the spatial location of each key. Eventually, the user can locate the key by gazing at it directly with little visual search. In parallel with the spatial memory, motor memory is also being strengthened and have an effect. Previous research showed that users could still type on a Qwerty soft keyboard with no visuals to a certain degree relying on finger motor memory [28, 53]. These memories shorten the recall and movement time during typing, allowing users to type at 20–40 WPM on a mobile device [45]. As the spatial and motor memory are valuable resources to reduce the learning effort, in our work we retain the Qwerty layout.

2.2 Reference Effect of Spatial Landmarks

Spatial memory is particularly important for efficient interaction on GUI. When an interface is spatially stable, users can remember and make rapid decisions about the location of target items [8, 17, 24, 37, 39, 43]. During the process of building up the spatial memory, the "*landmarks*" that are the location features over the interface, can serve as a strong reference frame for relocation. Researchers have shown that the accuracy of memorizing object location is significantly affected by its nearby objects and anchor points [35, 44].

Landmarks can be natural. For example, using the edges/corners of mobile devices and users' hands/fingers to organize menus [17, 40, 43]. When the number of selectable targets are large so that natural landmarks are not sufficient, artificial landmarks [41, 44] can be created. Researchers have used methods with artificial landmarks to improve the performance of searching and navigation. For example, color and shape changes [22, 25] can provide a clear hint and facilitate visual memory in icon search. Landmarks have also been used in first person navigation in virtual environment [46] and huge web retrieval [9].

For typing on the physical keyboard, landmarks also exit. Manufactures conventionally add raised tangible dots on key *F* and key *J* so that users can locate the entire keyboard and type totally eyes-free. We guess that the reference property of spatial landmarks can also be applied to the Qwerty soft keyboard when most users have already built up spatial memory on it. For example, with only the key *I* visible, we expect the user can easily and precisely recall the location of its adjacent key *U* and key *O*. This forms our basic research question: can the Qwerty soft keyboard display only a few keys or line segments as the spatial landmarks but allowing users to know exactly the location of each key and type both accurately and fast on it?

2.3 Keyboards on Mobile Devices

In this section, we review several representative categories of keyboard designs on mobile devices. Although their original goal may not be to free up the screen real estate, they do appear to occupy less display area than a standard Qwerty soft keyboard.

2.3.1 Reducing the Number of Keys. Researchers have examined various soft keyboard designs to reduce the number of keys, which has the benefit of saving the limited screen space on portable devices and allowing users to tap the buttons more accurately with finger. The basic idea is to group several letters into one button and employ word disambiguation techniques. LetterWise [29] condenses the 26 letters into 12 keys and uses prefix probabilities to guess the intended input. Braille-Touch [13] is a six-key chording pad that encodes letters into combinations of taps. H4-Writer [31] is a four-key design that uses Huffman coding to assign minimized key sequences to letters. Escape-Keyboard [4] is a sight-free and one-handed text entry method. To type a letter, the user touches one of four regions, and then performs a flick gesture in one of eight directions.

However, the limitation of these techniques is that they either use a new layout design or an entirely different typing manner against Qwerty, which leads to a long training process and low typing speed because of the lack of prior knowledge as discussed above. For example, H4-Writer could reach only ~20 WPM after 10 sessions' practice (600 phrases). The 1Line keyboard [26] wisely condenses the three rows of Qwerty into a single line with eight keys. The technique is based on the Qwerty layout and thus leverages user's existing typing experiences. But evaluations show that users achieved only 57.0% of the typing speed of using a full Qwerty keyboard after a 100-minute training.

2.3.2 Rearranging Keys and Using Gestures. For devices with an ultra-small display (e.g., smartwatch), one intuitive approach is to arrange the keys around the border of the screen (see the work of Arif et al. [2] for a review of text entry on smartwatch). Typically, Quikwriting [36] is a shorthand for entering text where the user drags the stylus between the central zone and eight outer zones. Compass [51] is a bezel-based technique with multiple cursors on a circular keyboard. Other approaches attempt to use gestures instead of tap which may be affected by the "*fat finger*" problem. EdgeWrite [49] is a gestural input method using patterns of movement between the corners and diagonals of an input square. Swipeboard [7] groups letters into nine regions and uses two swipes to specify a letter. These techniques are difficult to be applied on a larger display (e.g., smartphone) and still suffer from a low input speed.

In this work, the *K-Board* and *L-Board* preserve the Qwerty layout and "directly" reduce the keyboard visualizations. Since most users are experienced in typing with the Qwerty soft keyboard, we expect this premise can minimize the learning time. In addition, prior research has demonstrated that users can type at a high performance with the Qwerty keyboard even on the small smartwatch [15, 50] with the help of statistical decoding algorithms with spatial models and language models [18, 45]. We expect users can reach a close performance with our designs.

2.3.3 Semi-Transparent and Invisible Qwerty Keyboard. To reduce the visualization of the soft keyboard, an intuitive idea is to make the keyboard area translucent so that the user can see both the application view and the keyboard [23]. Researchers have also used a translucent virtual keyboard to show the camera view behind the keys to improve the performance of text entry on the move [1]. With this design, users did not need to swap their focus frequently between the interface as feedback was provided directly on the keyboard. However, in practical use, it is not always easy to get the proper opacity for different changing backgrounds. The overlaying design may also cause the main view and the keyboard view to interfere with the visibility of each other [16].

More recently, researchers are experimenting with more radical designs that the user directly types on an entirely invisible soft keyboard. As with our work, the feasibility of these techniques is also based on the user's spatial memory and motor memory of the Qwerty layout. BlindType [28] showed that on a smart TV users could recall relative key positions on a touch pad with the keyboard displayed on the TV screen. The statistical decoding algorithm predicts words relying on the relative position (the vector) between successive endpoints. Invisible Keyboard [53] makes the Qwerty soft keyboard entirely invisible on a smartphone. The authors derived the adapted spatial model for the Invisible Keyboard from experiment data based on the bivariate Gaussian distribution. The input speed of the Invisible Keyboard was measured to be 37.9 WPM and the WER to be <3% after training. However, thoroughly hiding the keyboard has several drawbacks. First, users are not allowed to accurately input a single letter or OOV words. Given the touch endpoints distribution in their papers, it seems that it is difficult for the user to type a target letter precisely. By examining Experiment 1 in the paper of Invisible Keyboard [53], we estimate the mean input accuracy of each key to be only 38.62% based on the bivariate Gaussian distribution with the values of offsets and variances presented in the article. Psychological research [27, 42] on keyboard memory also indicates that even skilled typists may have inaccurate explicit knowledge of Qwerty key locations. Second, because the techniques are on only word-level, users cannot know whether the decoder predicts correctly until they complete the final touch. This may add additional mental workload during typing. Third, the decoding performance may highly rely on the training data. The invisible typing patterns may be very different across users. In contrast, the *K-Board* and *L-Board* offer only a few but necessary spatial references, allowing the user to know exactly the location of each key after limited training. Therefore, input of OOV words is workable and users can always see what they are typing in real time. In addition, our keyboard design does not require a specially designed spatial model because its distribution of the touch points is very similar to a normal full Qwerty keyboard (see Figures 17 and 18).

3 DESIGNING *K*-BOARD

We use the standard Qwerty layout of soft keyboard: the first row and the second row differ by half a key, while the second row and the third row differ by one key. In fact, once any one of the 26 keys or any one of the 25 line segments is fixed, the entire keyboard has also been determined. The goal of the *K-Board* is to display the keys as few as possible while allowing users to type both accurately and quickly. In this section, we conduct a study to examine the user's input ability with only one landmark key visible. The results help us to compute the optimal design of the *K-Board.*

3.1 Appearance Factors to Consider

The visualization of the soft keyboard can also be reduced by just changing the appearance property (opacity, text, border style, etc.). An intuitive idea is to simply change the opacity of the entire keyboard so that the user can see both the application view and the keyboard [1, 23]. However, it is not always easy to get the proper opacity for different changing backgrounds. Such an overlaying design may also cause the main view and the keyboard view to interfere with the visibility of each other [16]. Another similar idea is "partially-invisible keyboard" [53], which removes the letter's text label and displays only the boundary line of each key. This design also has the problem of visual interference.

To alleviate the issue, the *K-Board* and *L-Board* directly set most of the keyboard elements to invisible. Meanwhile, the remaining visible landmark keys or line segments can still be transparent without text label to further improve the visual experience. Note that the focus of our work is not on the appearance factor, but on the user's understanding of the spatial location of the keyboard.

3.2 The Optimization Problem of *K*-Board

The goal is to maximize the overall text input performance. Given the number of landmark keys as n ($n < 26$), choose n letters from the alphabet A as the visible landmark key subset LK so that users can achieve the highest weighted input accuracy of all 26 letters. Then, the objective function is

$$\max_{LK \subseteq A} Score = \max_{LK \subseteq A} \sum_{c \in A} ACC(c \mid LK) \cdot f(c), \tag{1}$$

where c is any letter, ACC is the input accuracy of the letter, calculated by the probability that the touch point falls into the target key's boundary, and f is letter frequency serving as the weight. Like many previous works [3, 18, 50], for each key, we modeled the distribution of the touch points as a bivariate normal distribution $P(c)$ and estimated ACC thereafter.

Therefore, to derive the best design of *K-Board*, we need to know $ACC(c \mid LK)$ of each letter c for each possible combination of landmark keys LK. When only one key is visible ($n = 1$), there are $\binom{26}{1}$ possible keyboard designs and thus $\binom{26}{1} \times 26 = 676$ *ACCs* (distributions) to know. This quantity is still acceptable to acquire them through user experiments. When $n \geqslant 2$, however, there are $\binom{26}{n} \times 26 \geqslant 8{,}450$ distributions to know, which is impractical to directly get through user experiments. We decided to start from the case when $n = 1$, analyze the user input results, and then simplify the derivation of *K-Boards* with multiple landmark keys. *Kn*, where n is a number, represents "*K-Boards* with n landmark keys," and *Kn-C*, where C is a set of alphabetical letters, represents "the *K-Board* which has these letters as the landmark keys," in the following sections. For example, *K3-ABC* indicates a Qwerty keyboard that only displays 3 keys: *A*, *B*, and *C*.

3.3 Study 1: Understanding the Reference Effect of One Landmark Key ($n = 1$)

We conduct a study to collect typing data from participants on the *K-Boards* with only one landmark key visible (*K1*).

3.3.1 Participants. In total, 20 participants were recruited from the campus (6 female) with an average age of 24.5 ($SD = 2.5$). All participants use the Qwerty soft keyboard regularly and had the experience of using touch screen devices for an average of 8.3 years ($SD = 2.1$).

3.3.2 Apparatus. We collected data on both a smartphone and a smartwatch. The smartphone is a Huawei P9 running Android 7.0 with a 5.2″, 1,920 × 1,080 px display. The smartwatch is Samsung Galaxy Gear Live running Android 5.1 with a 1.63″, 320 × 320 px display.

3.3.3 Procedure and Task. Participants were first informed that this study concerned about how one visible key affects each letter's input distribution on the soft keyboard. Then they completed blocks of typing trials on all 26 *K1*s. The *K1* (key height: key width = 4 : 3) was rendered on the

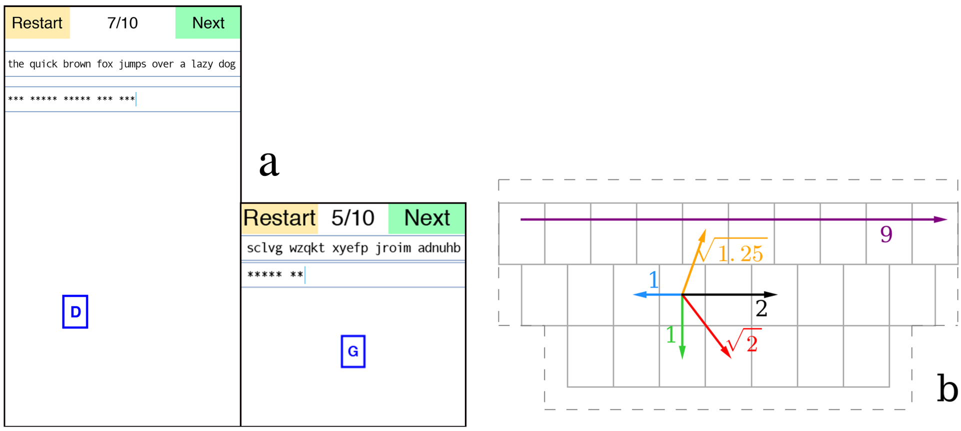

Fig. 2. (a) Interface of Study 1 on smartphone and smartwatch. (b) The effective ranges of the outer keys and examples of distance measurement.

touch screen with the lateral ends aligned with the screen edges. Participants could adjust the keyboard vertically to the comfortable position by a *long-press and drag* gesture.

Each letter-block corresponds to a *K1*. At the beginning of each block, the participant was given a 3-minute practice to get familiar with typing with the *K1* keyboard and the key locations. During the practice, the participant was free to type on it and switch whether to display the entire keyboard or only the landmark key. The participant was instructed to remember the location of each key as much as possible by leveraging both their spatial memory and the reference effect of the landmark key. Then, the participant was exposed to the typing trials. For each trial in the block, a target phrase was displayed on top of the screen. During typing, an asterisk would appear in the input filed whenever a tap event was detected to ensure a 1:1 mapping between user touches and characters in the phrase (Figure 2). The participant could perform a right-swipe gesture to leave a space and a left-swipe gesture to delete an input character when they realized they had committed an unintentional touch mistake. Participants were asked to type in their usual manner and input each key as accurately as possible. A 5-minute break was enforced between blocks to reduce the effect of previous typing experience on the location perception of the current keyboard. At the end of the experiment, the participant was interviewed about their typing and referencing strategy on the *K1*.

3.3.4 Experimental Design. A repeated measures within-subject factorial design was used. The independent variables were *K1 keyboard* (landmark key from *A* to *Z*) and *device* (smartphone and smartwatch). Ten participants performed the study on smartphone while the other ten on the smartwatch to avoid the interference. Each participant completed 26 blocks corresponding to all *K1s*. Each block consisted of 10 trials of phrase input. To cover all the letters, five of the trials were different pangrams and the other five were artificially generated phrases that consisted all 26 letters in a random order with four spaces (Figure 2). For each participant, the blocks and the trials within each block were presented in a random order. In summary, the experimental design was 2 devices × 10 participants × 26 blocks × 10 trials = 5,200 trials.

Before the formal user study, one of the authors and another recruited participant completed a pilot study with all the experiment tasks. It was found that the average completion time per block was 2.41 minutes on the smartphone and 2.66 minutes on the smartwatch, which means that the entire user study would take around 4.42 hours on the smartphone and 4.54 hours on the

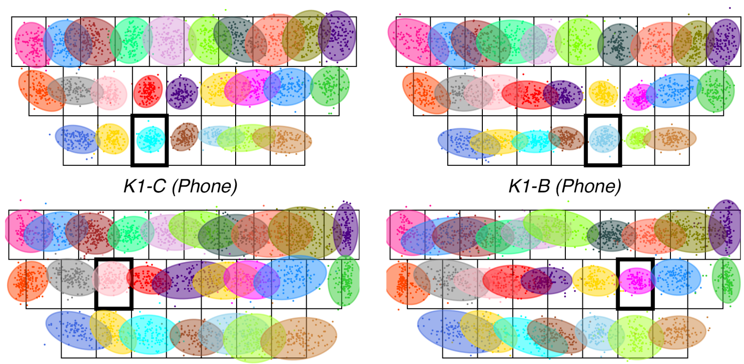

Fig. 3. Examples of touch point distribution using *K1* with a 95% confidence ellipse of each key.

smartwatch including the practice and break session. To reduce the mental and physical fatigue, in the formal user study, the experimental tasks for each participant was split up into consecutive days on the basis of duration per day. In each day, the participant performed the experiment for no more than 90 minutes. If the participant became fatigued and felt their input performance was thus affected, they could suspend later tasks early and postpone them to the next day. As a result, participants completed the entire experiment for three to five days. The completion time ranged from 4.28 to 4.83 hours (M = 4.60, Std = 0.18) on the smartphone and 4.45 to 4.89 hours (M = 4.69, Std = 0.14) on the smartwatch.

The research goal of this study was to investigate users' input ability when only one key is visible and then understand the reference effect of the landmark key. Although users can establish long-term spatial memory for the Qwerty layout [21, 53], the previous research [42] indicates that even skilled typists have poor explicit knowledge of the exact location per key. Therefore, a 3-minute practice was included before each letter-block to help users remember the key locations and more importantly, their locations relative to the landmark key. It was found in the pilot study that 3 minutes was a suitable time for this process. However, it is worth noting that as the letter-blocks progressed, the user's familiarity with the keyboard would increase cumulatively due to the practices, affecting the input performance. To balance and alleviate this effect, we made the presentation order of letter-blocks random and inserted a 5-minute break after each block.

3.3.5 Results. Across all participants, we collected a total of 148,200 labeled key taps, at least 100 touch points for each key in each *K1*. In the following analysis, the distance over the keyboard was measured in units of the key width horizontally and of the key height vertically. The effective ranges of the outer keys were extended by half the key width (Figure 2). Below, we summarize some of our key findings.

Participants Could Still Roughly Figure Out the Entire Keyboard Layout. In agreement with previous work [3, 18], the touch endpoints of each key roughly followed a bivariate normal distribution (Figure 3). Overall, the endpoints of different keys maintained the right relative position. We calculated the key input accuracy of all 676 distributions. With only one landmark key visible, the mean key input accuracy was 91.13% (min: 56.43%, max: 100.00%) on the smartphone and 79.74% (min: 30.55%, max: 98.87%) on the smartwatch. This confirms that participants have built up spatial correspondence and muscle memory of each key during the daily typing process, and can transfer such knowledge and motion to a nearly invisible keyboard [28, 53].

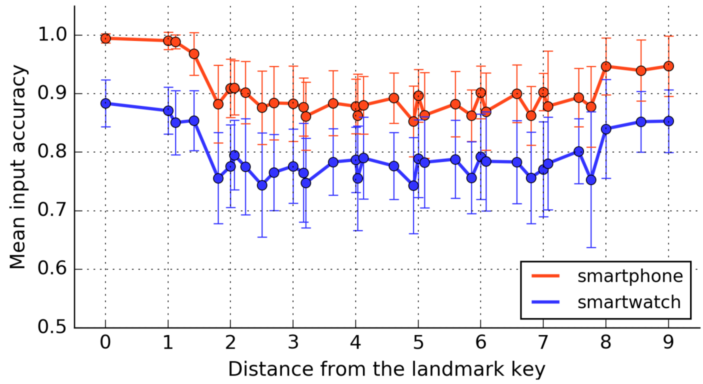

Fig. 4. Mean input accuracy of a target key at different distances from the landmark key.

The Landmark Key Provided a Strong Reference But Only in Limited Distances. We then analyzed how the single landmark key affected the typing pattern across the keyboard (Figure 3). Although overall participants could roughly perceive where the keys were, there existed many cases when they could not access the keys accurately. This is mostly reflected in the large offsets of the touch points from the actual key centers and the large standard deviations in the horizontal direction. For example, key *D* and *F*, key *E* and *R*, key *I* and *O* are easy to confuse with each other.

In such cases, the landmark key could offer a strong reference and largely reduce such confusions. For example, given *C* as the landmark key on the smartphone, the mean input accuracy of *C* and keys around it—*X*, *D*, *F* was significantly higher than given *B* as the landmark key (98.80% vs. 86.47% in Figure 3.*top*). This was because the mean offsets of the touch points of these keys were close ($offset_x$: 0.812 vs. 0.852), but their standard deviations on *K1-C* were much smaller than *K1-B* (std_x: 0.183 vs. 0.328). The reference effect of the landmark key also existed on the smartwatch on which the overlapping and confusion issue was more serious because of the ultra-small display and the "fat finger" problem. For example, in Figure 3.*bottom*, with landmark key *D*, key *E* and key *R* could be distinguished more clearly and be accessed more accurately than with landmark key *J*, where $P(E \mid D) = 86.84\% > P(E \mid J) = 69.16\%$, and $P(R \mid D) = 93.68\% > P(R \mid J) = 79.29\%$.

We then looked into the effect area of the landmark key. Over the keyboard, there are totally 34 distances between keys, ranging from 0, 1, $\sqrt{1.25}$, $\sqrt{2}$, ..., to 9, shown in Figure 2(b). Figure 4 shows the mean input accuracy of a target key at different distances from the landmark key. Distance $\sqrt{2}$ from the landmark key was a clear boundary of the target key's input accuracy. The key input accuracy at distances $\leqslant \sqrt{2}$ was much higher: 98.57% ($SD = 0.010$) on the smartphone and 86.49% ($SD = 0.013$) on the smartwatch, compared to 88.26% ($SD = 0.038$) on the smartphone and 78.50% ($SD = 0.035$) on the smartwatch at distances $> \sqrt{2}$ and < 8. A two-tailed Welch's *t*-test showed a significant difference between the two distance categories on both devices: $t = 15.38$ and 11.04, $p < .001$. Distance $\sqrt{2}$ represents the case where two keys are diagonally adjacent (e.g., key *D* and key *C*). This suggests that the landmark key provides a stronger reference only for its neighboring keys than others. When the distance from the landmark key is $> \sqrt{2}$, the input accuracy of a closer target key will not be significantly higher than a farther key.

Participants' subjective feedback on the reference strategy confirmed and explained the result. P3 commented that "With the single landmark, I could perceive very well the keyboard area, the key size, and the three rows' heights. The landmark key has a direct impact within and only within the local area. I had had some familiarity with the relative positions of the keys, and the warm-up process made this information more clear and accurate to me. For example, given landmark key *G*,

Table 1. *K*-Board Designs with the Top-5 Scores with Multiple Landmark Keys Based on Equation (1) on the Tested Smartphone

$n = 1$		$n = 2$		$n = 3$		$n = 4$		$n = 5$	
LKs[a]	Score	LKs	Score	LKs	Score	LKs	Score	LKs	Score
D	0.9227	*DJ*	0.9598	*EGK*	0.9794	*EGIN*	0.9853	*EYOCN*	0.9864
F	0.9218	*DK*	0.9585	*DGK*	0.9792	*EGJK*	0.9848	*EOFHN*	0.9863
K	0.9211	*RK*	0.9584	*SGK*	0.9787	*EGIJ*	0.9843	*EIGXN*	0.9863
O	0.9210	*EG*	0.9568	*DYK*	0.9785	*DGJK*	0.9843	*EOFGJ*	0.9863
G	0.9194	*RJ*	0.9543	*EGI*	0.9784	*EFUK*	0.9842	*EUAFK*	0.9863

[a]LKs stands for landmark keys.

I could easily know that key *F* is just on the left of it, key *C* is on its bottom left corner, and key *T* and *Y* are right above it. During typing, I could access these keys quickly and accurately, and was very confident. For farther keys, I knew their y positions. But for the x positions, I still relied on spatial memory and muscle feelings. The landmark key does not help much."

There Are Natural Landmarks for the Soft Keyboard. Previous research [17, 40] has shown that the screen edges can provide a natural external reference for spatial actions. We calculated the accuracy of each letter when it does not neighbor to the landmark key (distance $> \sqrt{2}$). It was found that key *Q*, *P*, *A*, and *L* that are located along the edge on average were more accurate than other internal keys (95.51% on the smartphone and 89.47% on the smartwatch). This also explains the high input accuracy when distances $\geqslant 8$ in Figure 4.

Based on Equation (1) and the experiment results above, the best *K1* design is *K1-D* (key *D* as the landmark key) with the highest score: 0.9227 on the smartphone and 0.8296 on the smartwatch. However, this accuracy does not meet the requirements of actual use. A good *K-Board* should have multiple landmark keys and the user should easily know the location of each key.

3.4 Designing *K*-Board: When the Number of Landmark Keys $n > 1$

Based on the landmark key's effective reference range, we make two assumptions to derive the final designs of *K-Board* with more than one landmark keys (Kn, $n \geqslant 2$). For any landmark key set LK and a target key α, we assume that

(1) If α is the neighboring key (distance $\leqslant \sqrt{2}$) of one of the landmark keys, say $\beta \in LK$, then $P(\alpha \mid LK) = P(\alpha \mid \beta)$;
(2) If α is not the neighboring key (distance $> \sqrt{2}$) of any landmark key, then $P(\alpha \mid LK) = \max_{\beta \in LK} P(\alpha \mid \beta)$;

where $P(\alpha \mid \beta)$ is the input accuracy calculated in the previous *K1* study.

The first assumption is that a landmark key will directly determine the input accuracy of its neighboring keys. The second assumption is that if the target key has no neighboring landmark key, then its input accuracy with all the landmark keys will not be lower than with only any one of them. We calculated the *K-Board* designs when $n = 2, \ldots, 5$ based on Equation (1) and these two approximations. The results with the highest scores on two devices are listed in Tables 1 and 2. In general, the landmark keys of the top results are scattered across the keyboard so that they can influence more area. Among, them, key *G*, *E*, *D*, *J*, and *K* played important roles because they either are or have neighboring keys with high letter frequency (e.g., *E*, *S*, *T*, *O*, *I*). Except key *E* which is the letter with the highest frequency, key *G*, *D*, *J*, and *K* are all located at the middle row on the keyboard. This has the benefit of covering and affecting more neighboring keys than being located at the top or bottom row, meanwhile, allowing users to identify the three rows more easily.

Table 2. *K*-Board Designs with the Top-5 Scores with Multiple Landmark Keys Based on Equation (1) on the Tested Smartwatch

$n = 1$		$n = 2$		$n = 3$		$n = 4$		$n = 5$	
LKs	Score	LKs	Score	LKs	Score	LKs	Score	LKs	Score
D	0.8196	*E J*	0.8464	*E G K*	0.8684	*EGIN*	0.8725	*ETIFN*	0.8763
K	0.8185	*D J*	0.8451	*D Y K*	0.8662	*EGIJ*	0.8718	*ETIDJ*	0.8763
F	0.8145	*R J*	0.8442	*E G I*	0.8650	*EGIM*	0.8712	*RISGN*	0.8762
E	0.8091	*D K*	0.8424	*D G K*	0.8647	*DGOJ*	0.8712	*EXIGN*	0.8753
R	0.8085	*E J*	0.8420	*S G K*	0.8643	*ETIN*	0.8710	*EIODJ*	0.8751

For *K2*, we chose *K2-DJ* as the final design which was ranked in the top two on both devices. Key *D* and key *J* are symmetrically located on the second row. With *K2-DJ*, the landmark keys and their neighboring keys cover 16 of the 26 keys. The natural landmarks—the screen edges can help users get another four keys *Q*, *P*, *A*, and *L*. However, the positions of key *T*, *Y*, *G*, *V*, *W*, and *O* are still unclear (Figure 6(b)).

For *K3*, we chose *K3-SGK* as the final design, which was ranked at third on the smartphone and fifth on the smartwatch. Note that *K3-EGK* and *K3-DGK* actually had slightly higher scores than *K3-SGK* in the calculated results because of the high frequency (weight) of character *E* and *R*. Before determining the final design, we asked participants in Study 1 to try out and type with the three candidate designs: *K3-EGK*, *K3-DGK*, and *K3-SGK*. Participants reported that *K3-SGK* was more visually comfortable and offered both a good local and global reference on the keyboard thanks to its symmetrical layout. With *K3-EGK*, participants had difficulty knowing the specific location of key *Z* and *X*, and the touch endpoints of keys in the lower left (*Z*, *X*, *C*) had a noticeable upward offset. With *K3-DGK*, the asymmetrical location of key *D* and key *K* forced users to receive different reference information and take different reference strategies when entering symmetrically located letters. For example, key *Z* is on the lower left of key *D* while key *M* is just below key *K*. Because participants already knew that *Z* and *M* are at the very ends on the bottom row, they preferred to get the same reference with the landmark keys. *K3-SGK* alleviated such issues and had an equally high score. Therefore, we balanced the scores and user preference after the formal optimization, and finalized the design of *K3* as *K3-SGK*. On *K3-SGK*, the landmark keys and their neighboring keys directly cover 22 of the 26 keys (Figure 6(c)). For the remaining four letters, key *Q* and *P* can be accessed by the screen edges. Key *R* (*U*) is at the center of key *S* (*K*) and key G in the horizontal direction. We expected users can also easily perceive their locations.

Following the similar designing process, we chose *K4-EIGN* as the final design for *K4* and chose *K5-EIGXN* for *K5*. *K4-EIGN* has the asymmetry issue. *K5-EIGXN* is symmetrically distributed and covers every key on the QWERTY layout (Figure 6(d)). However, compared with *K3-SGK*, *K5-EIGXN* adds two visible landmark keys, resulting in an increase of 67% in screen occlusion and overlapped neighboring areas between landmark keys (Figure 6(e)), while the score only increases by 0.78% on the smartphone and 1.27% on the smartwatch (Figure 5). Because our original goal is to reduce the keyboard area and free up the screen space, to keep the design compact, we choose *K3-SGK* as the final design for *K-Board*.

4 DESIGNING *L*-BOARD

In the above section, we have defined and designed *K-Board* that uses landmark keys as the basic visual reference element to reduce the Qwerty layout. In this section, we explore another possibility—the *L-Board*—that shows only a few dividing line segments between keys. Compared

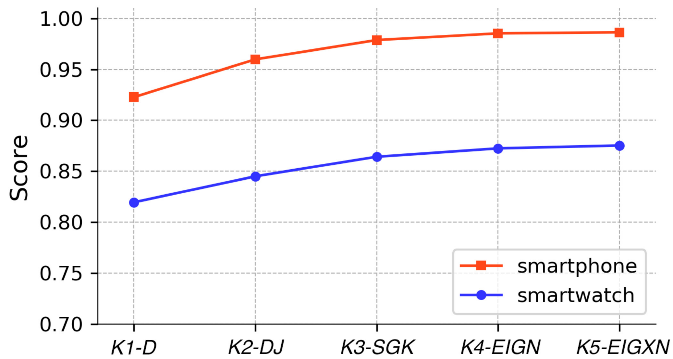

Fig. 5. Score (Equation (1)) of the best designs of *K*-Board with different number of landmark keys.

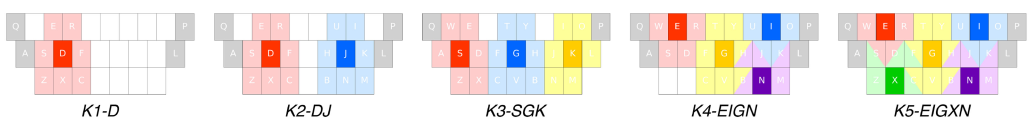

Fig. 6. Best designs of *K1*–*K5* and the reference ranges of the landmark keys on them.

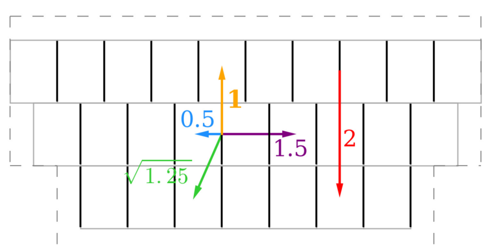

Fig. 7. Twenty-five valid line segments on the Qwerty layout, and examples of distance measurement from a line segment to a target key.

with displaying a complete key, the line segment further reduces the displaying area of the soft keyboard while still offering certain spatial reference. To get the best design of the *L-Board*, we take the similar user-centered, optimization-based design process as designing the *K-Board*. From Study 1, we know that users can easily distinguish the three rows of the keyboard, but the challenge is to identify adjacent keys within the same row. So in this design, we only consider the vertical line segments between keys. There are a total of 25 valid line segments on the Qwerty layout, shown in Figure 7.

4.1 The Optimization Problem of *L*-Board

Given the number of landmark line segments as n ($n < 25$), choose n segments from the segment's universal set U as the visible landmark segment subset LS, so that users can achieve the highest weighted input accuracy of all 26 letters. The objective function is

$$\max_{LS \subseteq U} Score = \max_{LS \subseteq U} \sum_{c \in A} ACC(c \mid LS) \cdot f(c), \tag{2}$$

where c is any letter, ACC is the input accuracy of the letter, and f is letter frequency as the weight.

To derive the best design of the *L-Board*, we need to know $P(c \mid LS)$ of each letter c for each possible combination of landmark segments LS. We still decided to start from the case when $n = 1$, and then generalize the derivation of *L-Boards* with multiple landmark segments. *Ln*, where *n* is a number, represents "*L-Board* with *n* landmark segments," and *Ln-C*, where *C* is a set of alphabetical letters, represents "the *L-Board* which has the *left* border lines of these letters as the landmarks," in the following sections.

4.2 Study 2: Understanding the Reference Effect of One Landmark Segment ($n = 1$)

We conduct a study to collect typing data from participants with the *L-Board* with only one landmark segment visible (*L1*). The experiment task and design were the same as Study 1 except for the keyboard visualization.

4.2.1 Participants and Apparatus. In total, 20 participants were recruited from the campus (7 female) with an average age of 22.25 ($SD = 1.5$). None of them had participated in Study 1. All participants use the Qwerty soft keyboard regularly and had the experience of using touch screen devices for an average of 7.5 years. The smartphone and the smartwatch were the same as in Study 1. In total, 10 participants performed the study on the smartphone while the other 10 on the smartwatch.

4.2.2 Task and Experimental Design. The experimental task was the same as Study 1 except that participants completed blocks of typing trials on all *L1s*. A repeated measures within-subject factorial design was used. The independent variables were *L1 keyboard* and *device* (smartphone and smartwatch). Each participant completed 25 blocks corresponding to all *L1s*. For each participant, the blocks and the trials within each block were presented in a random order. In summary, the experimental design was: 2 devices × 10 participants × 25 blocks × 10 trials = 5,000 trials.

4.2.3 Results. Across all participants, we collected a total of 142,500 labeled key taps, at least 100 touch points of each key in each *L1*. As on *K1*, participants could also roughly orient themselves on *L1*, where the touch endpoints of different keys maintained the correct relative positions. The mean key input accuracy was 90.54% on the smartphone and 80.43% on the smartwatch.

The landmark segment also provided a strong spatial reference and largely reduced local confusions. However, the effect area and the reference strategy of the landmark segment is significantly different from an entire landmark key. To make this clear, we demonstrate the results of the three situations in terms of in which row (top, middle, or bottom) the landmark segment lies on the keyboard. Figure 8 shows the mean input accuracy of the target key (except key *Q*, *P*, *A*, and *L* along the screen edges) at different distances from the landmark segment. The distance is defined as the length from the center of the target key to the center of the segment (Figure 7).

The Landmark Segment Provided Two Kinds of Spatial Reference: Separating and Pointing. The reference range of *separating* corresponds to the distance 0.5 and $\sqrt{1.25}$ in Figure 7 with a mean target key input accuracy of 98.72% ($SD = 0.009$) on the smartphone and 89.70% ($SD = 0.016$) on the smartwatch. An example is that with only the segment between key *F* and key *G* visible, participants could accurately access key *F*, key *G*, and the two keys below them: key *C* and key *V* (see Figure 8(d)). On the other hand, the reference range of *pointing* refers to the distance 1.0 and 2.0 in Figure 7 with a mean input accuracy of 99.09% ($SD = 0.010$) on the smartphone and 90.29% ($SD = 0.014$) on the smartwatch. For example, given the segment between key *U* and key *I* in the top row, participants could accurately access key *J* just below it in the middle row and even key *N* in the distant bottom row. Therefore, the effective reference range of one landmark segment is distance $\{0.5, 1.0, \sqrt{1.25}, 2.0\}$. The landmark segment provides a strong reference not only for its neighboring keys in the horizontal direction, but also for the keys that it is pointing to in the vertical direction. The mean input accuracy at other distances was only 85.99% ($SD = 0.044$) on

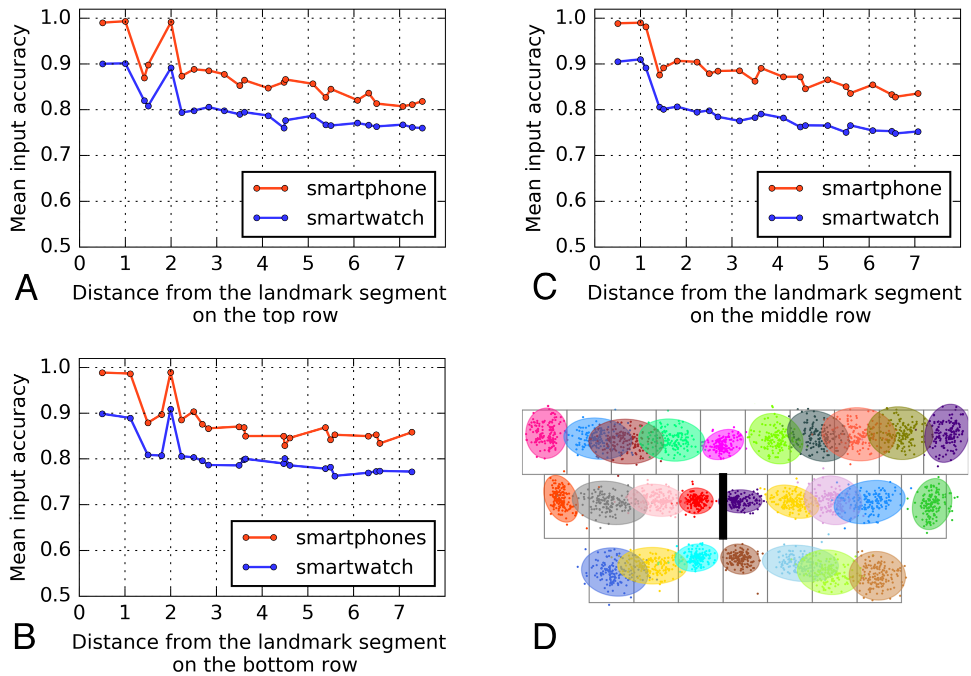

Fig. 8. Mean input accuracy of a target key at different distances from the landmark segment in three rows: (A) top, (B) middle, and (C) bottom. An example of the touch point distribution with a 95% confidence ellipse of each key when only the segment between key *F* and key *G* is shown.

the smartphone and 78.20% ($SD = 0.055$) on the smartwatch. In such cases, participants mainly relied on spatial or muscle memory rather than referencing the landmark segment.

4.3 Designing *L*-Board: When the Number of Landmark Segment $n > 1$

Based on Equation (1) and the experiment results above, the best *L1* design is *L1-G* (the left segment of key *G* as the landmark segment) with the highest score: 0.9189 on the smartphone and 0.8283 on the smartwatch. We then make two assumptions to derive the final designs of *L-Board* with more than one landmark segments (Ln, $n \geqslant 2$). For any landmark segment set LS and a target key α, we assume that

(1) If α is in the effective reference area (distance $\in \{0.5, 1.0, \sqrt{1.25}, 2.0\}$) of one of the landmark segments, say $\beta \in LS$, then $P(\alpha \mid LS) = P(\alpha \mid \beta)$;
(2) If α is not in the effective reference area (distance $\notin \{0.5, 1.0, \sqrt{1.25}, 2.0\}$) of any landmark segment, then $P(\alpha \mid LS) = \max_{\beta \in LS} P(\alpha \mid \beta)$;

where $P(\alpha \mid \beta)$ is the input accuracy calculated in the previous *L1* study.

These two assumptions are the same as the section of designing the *K-Board* except for the definition of the effective reference range. We calculated the *L-Board* designs when $n = 2, \ldots, 6$ based on Equation (1) and these two approximations. The results with the highest scores on two devices are listed in Tables 3 and 4. In general, the landmark segments of the top results are also scattered across the keyboard so that they can influence more area. Among them, the left segment of key *D*, *J*, *N*, *T*, *E*, and *O* played important roles because their reference areas can cover and affect more high-frequency characters.

Table 3. *L*-Board Designs with the Top-5 Scores with Multiple Landmark Segments Based on Equation (1) on the Tested Smartphone

$n = 2$		$n = 3$		$n = 4$		$n = 5$		$n = 6$	
LSs[a]	Score	LSs	Score	LSs	Score	LSs	Score	LSs	Score
DJ[b]	0.9465	TDJ	0.9663	*TODN*	0.9809	*EYOCN*	0.9883	*ERUIOV*	0.9891
XJ	0.9457	TXJ	0.9660	*TOXN*	0.9808	*EYOFN*	0.9883	*ERUIOG*	0.9890
RJ	0.9452	ETJ	0.9638	*TODJ*	0.9806	*EYOCJ*	0.9879	*EYUIOC*	0.9890
EJ	0.9440	TDN	0.9634	*TOXJ*	0.9803	*EYOFJ*	0.9878	*EYUIOF*	0.9890
DN	0.9431	TXN	0.9633	*ETON*	0.9783	*EOFVN*	0.9865	*ERTIOH*	0.9889

[a]LSs stands for landmark segments.
[b]DJ means only showing the *left* segments of key D and key J.

Table 4. *L*-Board Designs with the Top-5 Scores with Multiple Landmark Segments Based on Equation (1) on the Tested Smartwatch

$n = 2$		$n = 3$		$n = 4$		$n = 5$		$n = 6$	
LSs	Score	LSs	Score	LSs	Score	LSs	Score	LSs	Score
RJ	0.8574	TDJ	0.8781	*TODJ*	0.8931	*EYOFN*	0.8990	*EUFGLM*	0.9012
DJ	0.8572	TDN	0.8772	*TODN*	0.8927	*EYOCN*	0.8986	*EUDLVM*	0.9012
RN	0.8570	TXJ	0.8763	*TOXJ*	0.8909	*EYOCJ*	0.8983	*RUSGLM*	0.9011
DN	0.8569	TDM	0.8763	*TOXN*	0.8909	*TIDHL*	0.8983	*EOFHVN*	0.9010
XJ	0.8564	DGJ	0.8759	*ODGN*	0.8904	*EOFJV*	0.8983	*EUFHLV*	0.9010

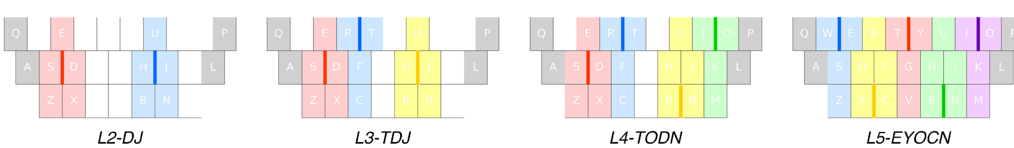

Fig. 9. Best designs of *L2*–*L5* and the reference ranges of the landmark segments on them.

For $n = 2$, 3, and 4, we chose *L2-DJ*, *L3-TDJ*, and *L4-TODN* as the final design, respectively, which were ranked in the top positions on both devices. However, their direct reference area adding key *Q*, *P*, *A*, and *L* can only cover 14, 18, and 22 of the 26 keys (Figure 9(a)–(c)). In addition, they all have an asymmetrical or shifted landmark layout, which is visually defective and probably affects user input performance like the *K3-DGK* in the previous section.

For *L5*, we chose *L5-EYOCN* as the final design that was ranked at 1st on the smartphone and 2nd on the smartwatch. On *L5-EYOCN*, the effective reference area of the landmark segments directly cover 22 of the 26 keys, while the rest four keys, *Q*, *P*, *A*, and *L* can just be referenced by the screen edges (Figure 9(d)). *L5-EYOCN* also have a perfect symmetrical layout, with three segments evenly distributed on the top row and the other two evenly distributed on the bottom row. It is visually comfortable and expected to provide both a good local and global reference on the keyboard. When the number of landmark segments increases further, the *L*-Board (e.g., *L6-ERUIOV*) will have an inferior layout and wasted, overlapped reference regions between landmark segments while the score only increases by 0.081% on the smartphone and 0.247% on the smartwatch (Figure 10). To keep the design compact, we choose *L5-EYOCN* as the final design of *L-Board*.

In the above, to save the screen space we have designed *K3-SGK* for *K-Board* and *L5-EYOCN* for *L-Board* as two new visualizations of the standard Qwerty soft keyboard through a user-centered,

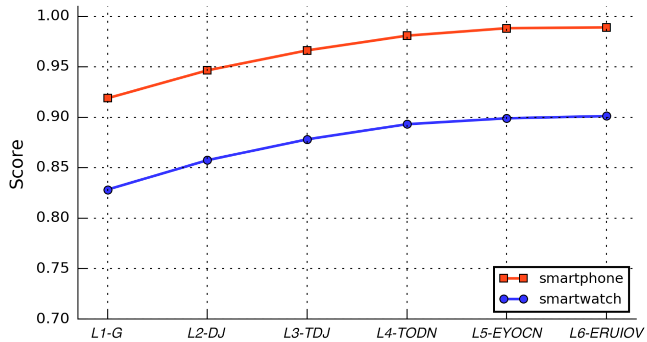

Fig. 10. Score (Equation (2)) of the best designs of *L*-Board with different number of landmark segments.

optimization-based approach. In the following sections, we first describe how our keyboard designs enable common editing operations and non-alphabetical characters' input, and how the keyboards are integrated into real applications. Then, we conduct two user studies to investigate the feasibility, the input performance, and the benefit of freeing up more screen space of the two keyboard designs.

5 SUPPLEMENTAL INTERACTIONS AND INTERFACE IMPLEMENTATION

To make the text entry experience more complete, we also design several supplemental functions for our two keyboard designs. The following gesture interactions, interface features and statistical decoding algorithms are integrated into the applications in the next user studies.

5.1 Interface Design and Implementation

As with a normal keyboard, our low-occlusion keyboard automatically appears as a floating window when the user touches on the editable contents. Users can interact with the main interface outside the keyboard area. Any touch event within the keyboard area is recognized as the user's input action.

The keyboard location depends on the user's editing position. It is by default placed at the lower half of the screen. If conflicted with the editing position, the keyboard is then automatically placed above the edit position with a proper margin. The user can also manually adjust the keyboard vertically by a *long press* and drag gesture.

Suggestion bar is only visible when the user is typing. On the smartphone, the suggestion bar is placed right above the keyboard area or the edit position presenting four to five top suggestions. On the smartwatch, the suggestion bar is always placed at the top of the screen with three suggestions. Taps on the suggestion bar will automatically insert a space after the selected word. Users can swipe right on the bar to look at more suggestions.

Prompt function is designed for cases when users happen to forget a key's position and want precise input such as OOV words (e.g., password, email, place names). On the phone, if the finger stays on the keyboard area for over 150ms, the hit key and its neighbors become visible until the finger leaves the screen. On the watch, the difference is that these keys are shown in a pop-up view [47] above the finger with a cursor indicating the finger position.

The keyboard will disappear when the system detects that users have completed the input (e.g., tapping the send button in a messaging app). Alternatively, users can manually minimize the keyboard by a two-finger pinch gesture.

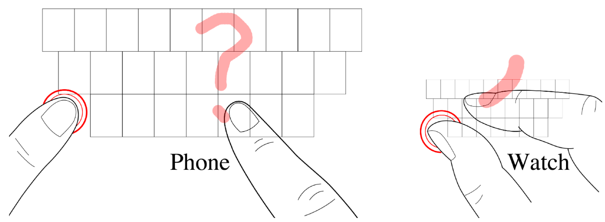

Fig. 11. Gestural input for basic non-alphabetical characters.

5.2 Gestural Input

Besides the 26 letters, we also implemented several basic editing operations and the input of common non-alphabetical characters. Since our goal is to reduce the occlusion, the central idea is using gestures. For common operations, inspired by [26], the entire keyboard area is the effective gesture area. The user can perform the following one-finger gestures:

- *Short left swipe*: delete the last input character.
- *Long left swipe*: delete the entire last input word.
- *Right swipe*: enter a blank space.
- *Down and right swipe*: enter the first word in the suggestion bar and then automatically insert a blank space.

For non-alphabetical characters (punctuation and symbols), we learned from the gesture technique in the previous work [11] and adapt it from large touch screens to smaller devices (Figure 11). Specifically, on the phone the user touches the bottom left corner of the keyboard area with the left thumb and holds to the activate the non-alphabetical mode. The user then draws the intended character gesture with the right-hand finger. When the left thumb leaves the screen, the gesture is recognized and the result is entered to the text field. On smartwatch, the difference is that the interaction is done with only one hand: the right thumb and the index finger. The system now supports number "0, 1, . . . , 10," common punctuation ". , : ; ! ? - ()," and symbols "@ # % $ &." The gesture recognition is done using template matching.

5.3 Probabilistic Decoding Algorithm

The decoding algorithm of both our keyboard designs employ a 100k word English vocabulary based on the Bayesian model: $P(W|I) \propto P(I|W)P(W)$, where $I = \{i_1, i_2, \ldots, i_n\}$ is the user's discrete input sequence with horizontal and vertical components of the touch points on the screen, and $W = \{w_1, w_2, \ldots, w_n\}$ is a possible word in the vocabulary.

For the touch model, we assume that the user's each current tap is independent of any previous tap. So, $P(W|I) = \prod_{k=0}^{n} P(i_k|w_k)$. We calculate $P(i_k|w_k)$ based on the two-dimensional Gaussian distribution. For the distribution of each key, the center and covariance matrix was first initialized using the touch points data in Study 1 and Study 2. For example, the distribution of key F on *L5-EYOCN* is set as the distribution of key F on *L1-C* in Study 2, because we have assumed that users would mainly reference the left segment of key C to access key F. The parameters of each key's distribution will be updated as the user studies progresses. For the language model, we use a word-level bigram model where $P(W) = \prod_{i=0}^{n} P(W_i|W_{i-1})$. The bigram word frequency is acquired from the Google Web 1T 5-gram database [6].

Fig. 12. The three tested keyboard design in Study 3. The experiment interface is similar to Study 1&2.

6 STUDY 3: EVALUATING INPUT PERFORMANCE

The goal of this study was to validate the feasibility of our two proposed keyboard designs: *K3-SGK* and *L5-EYOCN*. Specifically, we aimed to examine the following three questions:

(1) Can the user type accurately and quickly on the partially visible keyboards, and how is the learning process?
(2) Since one of the primary motivations is to allow users to know exactly the location of each key, how is the input performance for OOV words?
(3) Can the user keep the typing skill after a few days?

Comparing with a full Qwerty keyboard, we evaluated the input speed and input accuracy on both character and word level of the two keyboard designs. In the following studies, to facilitate writing and reading, we simply refer to full Qwerty keyboard as *FK*, *K3-SGK* as *K3*, and *L5-EYOCN* as *L5*.

6.1 Participants and Apparatus

In total, 40 subjects (referred to as $P1-P40$, 16 female) participated in this experiment, 20 for the smartphone ($P1-P20$) and the other 20 for the smartwatch ($P21-P40$). None of them had participated in Study 1 and Study 2.

To make the experiment results representative on a relatively general user population, the 40 participants (aged from 16 to 45) included 16 university students, 10 high school students, 6 teachers, and 8 office staff. All participants were using at least one touch screen device and regularly used the Qwerty soft keyboard for text entry. The self-reported typing skill on the Qwerty soft keyboard (1: extremely weak; 10: extremely strong) ranged from 4 to 10 (mean: 5.9, median: 6). The self-reported familiarity with the key locations on the Qwerty layout (1: extremely unfamiliar; 10: extremely familiar) ranged from 3 to 9 (mean: 5.95, median: 6).

The smartphone in this experiment was a Huawei P20 running Android 8.1 with a 5.8″, 2,244 × 1,080 px display. The smartwatch was Samsung Galaxy Gear Live running Android 5.1 with a 1.63″, 320 × 320 px display.

6.2 Experiment Design

The study was a typical phrase transcription task. We evaluated the input performance on both the smartphone and the smartwatch. For each device, a repeated measures within-subject factorial design was used. The independent variable was keyboard design (Figure 12), including *K3*, *L5*, and *FK*. The entire experiment consists of four stages.

— *Stage 1*: Using the *first* keyboard, the participant performed two sessions of phrase transcription in sequence, first on dictionary words and then OOV words. The dictionary-word session included 5 blocks, each with 15 phrases. The OOV-word session included 2 blocks, each with 15 phrases.
— *Stage 2*: The participant repeated the task in Stage 1 with the *second* keyboard.
— *Stage 3*: The participant repeated the task in Stage 1 with the *third* keyboard.

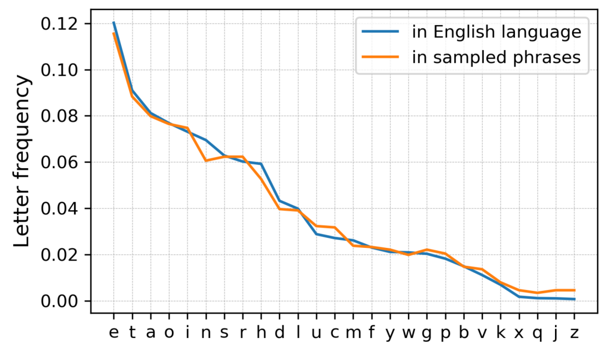

Fig. 13. The letter frequency distribution of the sampled phrases in our study and in English language.

— *Stage 4*: After a week, using each of the *K3* and *L5* keyboard, the participant performed two revisit sessions of phrase transcription: dictionary words and OOV words. Each session included 2 blocks, each with 15 phrases.

The goal of Stage 1–3 was to examine the user's learning process (5 blocks of the dictionary-word session) and the final input performance (on both dictionary words and OOV words) of the three keyboards. The order of the keyboards was counter-balanced across participants. Stage 4 is designed to examine whether the input performance would be maintained after a few days. In summary, the experiment included:

— *Stage 1–3*: 2 devices × 20 participants × 3 keyboards × (5 + 2) blocks × 15 phrases = 12,600 phrases (9,000 dictionary-word phrases and 3,600 OOV-word phrases).
— *Stage 4*: 2 devices × 20 participants × 2 keyboards × (2 + 2) blocks × 15 phrases = 4,800 phrases (2,400 dictionary-word phrases and 2,400 OOV-word phrases).

For the dictionary-word blocks, each participant typed the same set of (5 + 2) × 15 = 105 different phrases. The 105 phrases were randomly sampled from Mackenzie and Soukoreff's phrase set [30] by trying to follow the letter frequency distribution in English language as much as possible (see Figure 13). For the OOV-word blocks, each participant typed the same set of (2 + 2) × 15 = 60 different phrases. Each phrase was composed of four to five OOV words. In total, there were 258 different OOV words (e.g., "rbs," "ephross," "gymboree"), with a mean word length of 5.70 (Std = 1.93). These OOV words were randomly sampled from the Enron mobile email dataset [10]. No capitalization and punctuation appeared in our phrase set.

For each participant, the experiment on each keyboard design in Stage 1–3 was scheduled at least 24 hours apart to reduce the interference of the experience on the other keyboards and the effect of phrase/word repetitions. The experiment of Stage 4 was scheduled at least 7 days after Stage 3.

6.3 Procedure and Task

6.3.1 Stage 1–3. The participant was first informed that this study concerned about the input performance and the learning process of each keyboard design. Then, in each of the Stage 1–3, the participant completed five blocks of dictionary-word phrase input and another two blocks of OOV-word phrase input using the specified keyboard.

In each block, the participant was asked to type 15 phrases. For each phrase trial, a task phrase was displayed on the top of the screen. During typing, the actual input characters were shown in the input filed and the top suggestions by the probabilistic decoder were shown in the suggestion bar. The participant could decide whether to use the suggestion bar or not depending on their input. After entering a phrase, the user pressed the "Next" button to proceed to next phrase.

In the first five blocks of the dictionary-word phrase input, although the decoder could usually output the correct suggestion even if the user's input was not completely accurate, the participant was asked to type each "character" as accurately as possible because we hoped that the participant would become more familiar with the location of each key as the experiment progressed. On this basis, the participant was asked to type as quickly as possible. The participant could decide whether to correct the input error, but was encouraged to remember the errors and gradually improve the performance. However, if the input error was caused by the uncertainty of finger touch—the participant intended to tap a key but accidentally tap another because of the occlusion and fat finger problem, the participant could ignore it and continue. In the two blocks of the OOV-word phrase input, the participant was instructed to enter it as accurately and quickly as possible.

Each block took 3–7 minutes. Before each block, the participant was given a 1-minute warm-up to get familiar with typing on the keyboard. After each block, a 3-minute break was carried out in which the participant was interviewed to rate their subject feedback on *Mental Effort* on a 10-level Likert scale. *Mental Effort* measures the amount of effort that the user spent on recalling the key locations while typing (1: extremely easy as typing on the *FK*; 10: extremely hard). It reflects the user's familiarity with the location of each key on a partially visible keyboard. In total, the experiment took around 1 hour for each stage.

6.3.2 Stage 4. In Stage 4, using each of the *K3* and *L5* keyboard, the participant completed two blocks of dictionary-word phrase input and two blocks of OOV-word phrase input. Before the formal study of each keyboard, the participant performed a 2-minute warm-up block to quickly get re-familiar with the keyboard. During the formal study, no more practice was given. The participant was instructed to enter the phrase as accurately and quickly as possible.

6.4 Metrics

The metrics are CER, WER, and input speed, defined as follows:

$$Uncorrected\ Character\ Error\ Rate\ (UCER) = \frac{INF}{C + INF + IF} \times 100\%, \tag{3}$$

$$Corrected\ Character\ Error\ Rate\ (CCER) = \frac{IF}{C + INF + IF} \times 100\%, \tag{4}$$

where C is number of the input characters that are not errors, INF is the character errors that go unnoticed and appear in the user's actual transcribed text (before corrected with the word suggestions from the decoder), and IF is character errors in the input stream that are corrected. CCER can reflect the learning process of a keyboard and UCER reflects the final input results:

$$WER = \frac{IW}{\#\, of\ words\ in\ S} \times 100\%, \tag{5}$$

where IW is the number of words that are not correctly input in the final transcribed string S:

$$WPM = \frac{|S - 1|}{T} \times 60 \times \frac{1}{5}, \tag{6}$$

where T is the elapsed time.

Based on the experiment design, we then in sequence analyze the results of the input performance using different keyboard designs in (1) the learning session (five blocks of dictionary-word phrase typing) in Stage 1–3, (2) two blocks of OOV-word phrase typing in Stage 1–3, and (3) revisit-typing in Stage 4.

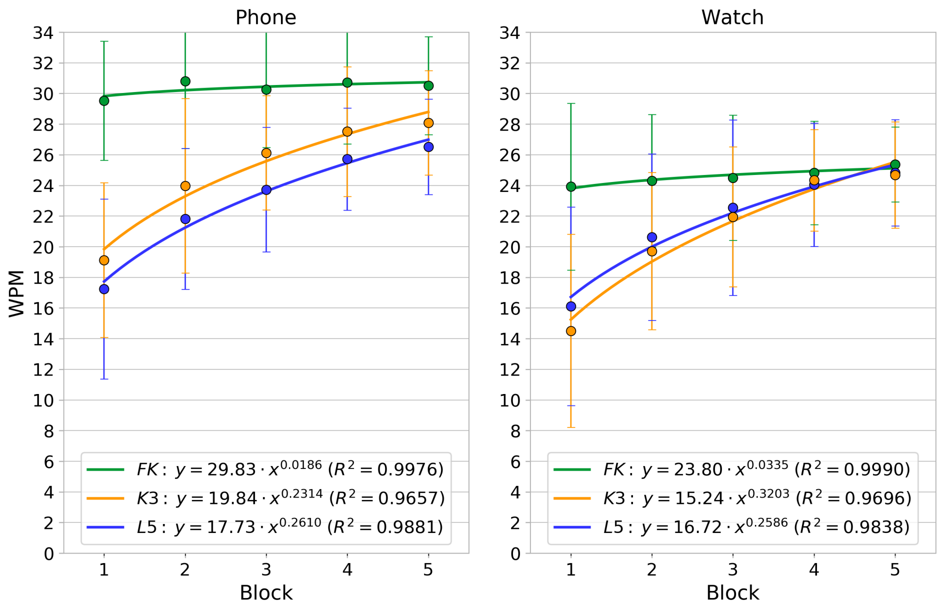

Fig. 14. Mean input speed (Std. Error) over five blocks with fitting a power function.

6.5 Results of Learning Session

6.5.1 Input Speed. Repeated-measures ANOVA (RM-ANOVA) was used to compare the results of input speed. Shapiro–Wilk tests showed that the WPM data of each keyboard design was normally distributed on both the smartphone (*FK*: $W = .990, p = .646$; *K3*: $W = .980, p = .126$; *L5*: $W = .979, p = .104$) and the smartwatch (*FK*: $W = .986, p = .375$; *K3*: $W = .982, p = .193$; *L5*: $W = .984, p = .292$). The Greenhouse–Geisser corrected results were reported in cases where Mauchly's test indicated a violation of sphericity.

Figure 14 shows the means of input speed over 5 blocks on two devices across all participants. Two-way RM-ANOVA revealed a significant interaction between keyboard design and block affecting WPM on both devices (*Phone*: $F_{3.54,67.16} = 11.95$, $p < .001$; *Watch*: $F_{8,152} = 4.84$, $p < .001$). There was a significant main effect of the keyboard design on input speed (*Phone*: $F_{2,38} = 63.72$, $p < .001$; *Watch*: $F_{2,38} = 33.14$, $p < .001$). As expected, participants achieved the highest input speed on *FK* at 30.13 WPM on the phone and 24.53 WPM on the watch because all keys were visible, requiring little reaction/recalling time when typing. They served as the baselines to examine the performance of the other two keyboards.

RM-ANOVA showed that the blocks had a significant main effect on the input speed of both *K3* (*Phone*: $F_{2.66,50.46} = 19.66$, $p < .001$; *Watch*: $F_{2.67,50.67} = 16.54$, $p < .001$) and *L5* (*Phone*: $F_{2.34,44.36} = 43.77$, $p < .001$; *Watch*: $F_{4,76} = 11.46$, $p < .001$):

On the smartphone, at the 1st block, the mean input speed of *K3* and *L5* was 19.12 WPM and 17.24 WPM, respectively, 57.2%–62.1% of the input speed of *FK*. Then, the input speed increased quickly in the first three blocks and trended to stabilize at the 4th and 5th block. At the 5th block, participants averaged 28.08 WPM with *K3* and 26.52 WPM with *L5*, only 7.15% and 11.96% slower than the *FK*. Comparing *K3* and *L5*, Figure 14 shows a similar learning curve with the two keyboards but the means of input speed of *K3* were higher than *L5* in each of the five blocks. Paired *t*-tests showed that the difference between them was significant in all blocks ($p < .05$) except the 2nd block ($p > .10$).

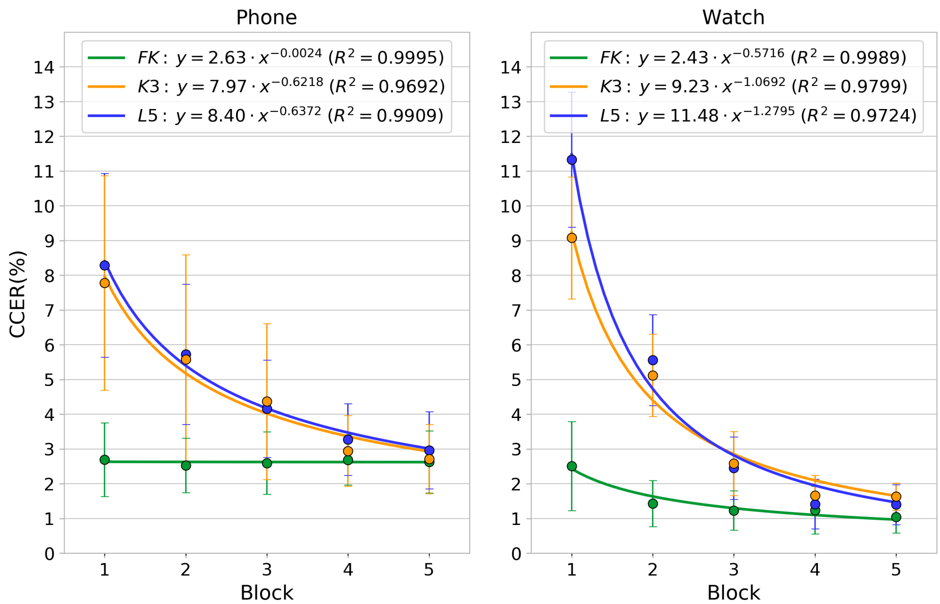

Fig. 15. Mean corrected character error rate (Std. Error) over five blocks with fitting a power function.

On the smartwatch, the input speed with *K3* (16.22 WPM) and with *L5* (14.51 WPM) was 67.39% and 60.66% of *FK* (23.92 WPM) at the 1st block. After 5 blocks' learning, participants reached 25.13 WPM with *K3* and 24.98 WPM with *L5*, which are only 0.96%–1.55% slower than *FK*. Paired *t*-tests showed that the difference between *K3* and *L5* was not significant in each of the five blocks ($p > .10$).

6.5.2 Input Accuracy. RM-ANOVA was used to compare the results of input accuracy. Shapiro–Wilk tests showed that the CCER and UCER were lognormal (base 10) for both *K3* and *L5* on both the smartphone and smartwatch ($p > .05$). However, the data of WER was non-normal (even after simple math transformation, e.g., log, square root) for both *K3* and *L5* on both devices ($p < .001$). We therefore applied the Aligned Rank Transform (ART) [38] on the original data with *ARTool* [48] before the ANOVA procedure. The Greenhouse–Geisser corrected results were reported in cases where Mauchly's test indicated a violation of sphericity.

Figure 15 shows the means of CCER over five blocks across all the participants. Still, *FK* had the best performance on CCER at 2.63% on the phone and 1.04% (in the last four blocks) on the watch. RM-ANOVA showed that there was a significant main effect of the number of blocks on the CCER of both *K3* (*Phone*: $F_{4,76} = 18.10$, $p < .001$; *Watch*: $F_{4,76} = 17.30$, $p < .001$) and *L5* (*Phone*: $F_{4,76} = 24.42$, $p < .001$; *Watch*: $F_{4,76} = 26.57$, $p < .0001$), confirming the significant learning effect of our two keyboards.

One the phone, the CCER at the 1st block was 7.79% using *K3* and 8.29% using *L5*. Then, the CCER on both keyboards quickly decreased, tended to be stable at the 4th block, and finally reached 2.71% and 2.96%, respectively, in the 5th block. Considering the last two blocks only, paired *t*-tests showed no significant differences between any two keyboards of *FK*, *K3*, and *L5* on CCER.

On the watch, the CCER had a similar curve of improvement. The CCER at the 1st block was 9.08% with *K3* and 11.33% with *L5*. At the 5th block, the CCER of *K3*, *L5*, and *FK* was 1.63%, 1.40%, and 1.04%, respectively, which were even smaller than those on the phone.

Table 5. Means (Std) of UCER and WER of Dictionary-Word Phrase Input with Three Keyboards

	On phone		On watch	
	UCER	WER	UCER	WER
FK	1.59 (1.98)	0.19 (0.43)	6.20 (1.75)	0.23 (0.43)
K3	2.17 (1.76)	0.21 (0.59)	6.08 (1.73)	0.21 (0.53)
L5	2.26 (2.26)	0.22 (0.41)	5.81 (2.04)	0.25 (0.48)

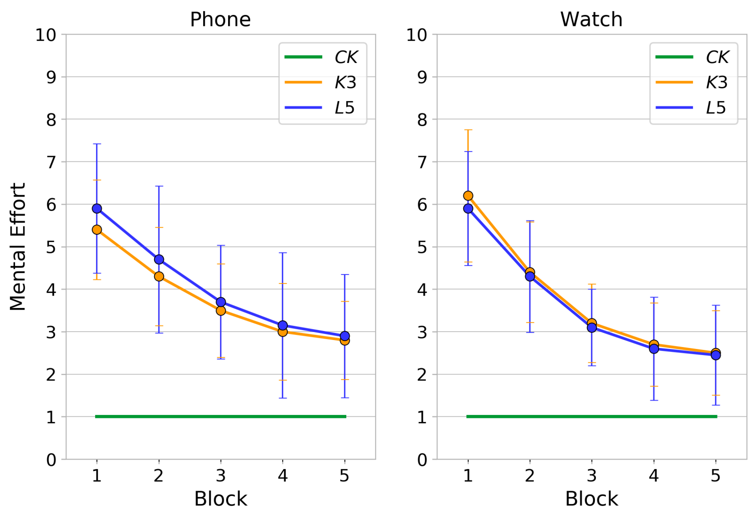

Fig. 16. Means (Std. Error) of participants' *Mental Effort* over five blocks.

For any of the keyboards, RM-ANOVA showed no significant main effect of the number of blocks on the UCER and WER in each *keyboard* × *device* condition ($p > .05$). The means of UCER and WER of the five blocks across the participants are shown in Table 5. RM-ANOVA did not show a significant main effect of keyboard on UCER and WER on both devices ($p > .05$), where the UCER using *FK*, *K3*, and *L5* was 1.59%, 2.17%, and 2.26%, respectively.

6.5.3 Input Mental Effort. Figure 16 displays the means of self-reported *Mental Effort* on a 10-level Likert scale over five blocks across all the participants (1: extremely easy as typing on *FK*; 10: extremely hard). Similar to the input speed and accuracy, the *Mental Effort* also showed a clear decline trend. Friedman tests showed that the number of blocks had a significant main effect on the *Mental Effort* with both *K3* (*Phone*: $\chi^2 = 73.38$, $p < .001$; *Watch*: $\chi^2 = 75.12$, $p < .001$) and *L5* (*Phone*: $\chi^2 = 74.05$, $p < .001$; *Watch*: $\chi^2 = 76.03$, $p < .001$). At the 5th block, the mean *Mental Effort* of using *K3* and *L5* was 2.85 on the phone and 2.475 on the watch. Wilcoxon Signed-Ranks tests indicated that the difference of *Mental Effort* between *K3* and *L5* was not significant in all blocks ($p > .05$).

6.6 Results of OOV Session

Table 6 shows the means of WPM, CCER, and UCER of the two OOV blocks across all participants using different keyboard designs. As in the learning session, Shapiro–Wilk tests indicated that the

Table 6. Means (Std) of Input Performance of OOV-Word Phrase Input with Three Keyboards

	On phone			On watch		
	WPM	CCER	UCER	WPM	CCER	UCER
FK	27.22 (5.0)	3.52 (1.6)	0.28 (0.3)	22.49 (4.1)	4.67 (1.8)	1.02 (0.7)
K3	23.48 (5.5)	4.87 (1.5)	0.58 (0.6)	22.59 (3.9)	5.35 (1.9)	1.01 (0.6)
L5	23.13 (4.7)	4.98 (1.5)	0.62 (0.7)	22.47 (3.9)	5.04 (1.8)	1.09 (0.6)

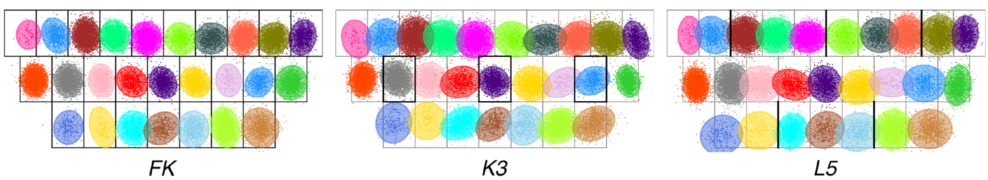

Fig. 17. Touch points on three keyboards on the smartphone in Study3.

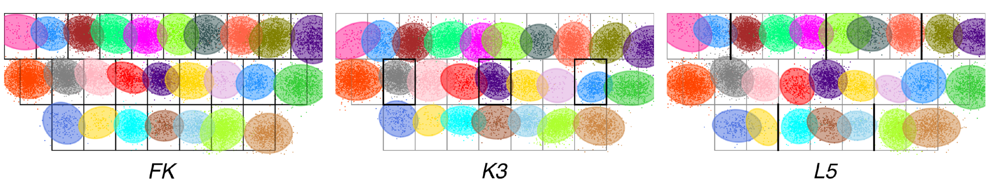

Fig. 18. Touch points on three keyboards on the smartwatch in Study3.

WPM data was normal while the CCER and UCER data were lognormal for each keyboard on both devices ($p > .05$).

On the phone, compared to the input performance of the last two dictionary-word blocks, WPM of all three keyboards declined, with a drop of 10.7%–14.7%. Meanwhile, CCER increased by 0.88–2.04 percentage points but UCER decreased by 1.31–1.64 percentage points. This indicates that when typing OOV-words, participants were more careful trying to enter each character correctly in the final transcribed text. Participants still performed the best with *FK* on each of WPM, CCER, and UCER. RM-ANOVA showed that the effect of the keyboard design on all the three measures were significant ($p < .05$).

On the watch, compared to the dictionary-word typing task, the input performance of OOV-word had a similar trend to that on the phone, where WPM decreased, CCER increased, and UCER decreased. But RM-ANOVA showed that there was a moderate significant effect of the keyboard design only on CCER ($F_{1.48,27.58} = 7.22, p < .042$).

6.7 Distribution of Touch Points

Figures 17 and 18 show all the touch points in the last two blocks of the dictionary-word session plus the two blocks of the OOV session merged from all participants, with the 95% confidence ellipses modeled by a bivariate normal distribution for each key. It provides an intuitive display of how each keyboard affects the user's typing pattern (e.g., with *L5*, the distribution of touch points of key *D*, *F*, *C*, *V* and *H*, *J*, *B*, *N* are obviously affected by the two landmark lines in the bottom row). In the horizontal direction, the mean absolute value of $offset_x$ of all keys was 0.052 for *FK*, 0.085 for *K3*, and 0.091 for *L5* on the phone, and 0.105 for *FK*, 0.093 for *K3*, and 0.119 for *L5* on the

Table 7. The Score (Equation (1)) of Three Keyboard Designs Calculated Using the Touch Points of the Last Two Sessions in Study 3

	FK	*K3*	*L5*
On phone	0.9849	0.9570	0.9548
On watch	0.9209	0.9230	0.9256

Table 8. Means of Input Performance of *K3* and *L5* in the Revisit Sessions of Stage 4

	Dictionary-word session				OOV-word session		
	WPM	CCER	UCER	WER	WPM	CCER	UCER
K3 phone	28.79 ↑	2.54 ↓	2.43 ↑	0.21 ·	24.89 ↑	5.10 ↑	0.60 ↑
L5 phone	24.51 ↓	3.34 ↑	2.45 ↑	0.20 ↓	22.54 ↓	5.23 ↑	0.64 ↑
K3 watch	25.15 ↑	1.59 ·	6.11 ↑	0.22 ↑	22.68 ↑	5.10 ↓	1.04 ↑
L5 watch	24.77 ↓	1.47 ↑	5.69 ↓	0.25 ·	22.49 ↑	4.88 ↓	1.05 ↓

The arrows indicate the changes compared to the last four sessions of Stage 1–3.

watch. The mean std_x 0.188 for *FK*, 0.225 for *K3*, and 0.233 for *L5* on the phone, and 0.275 for *FK*, 0.277 for *K3*, and 0.272 for *L5* on the watch.

In the vertical direction, compared to *FK* and *L5*, the touch points of the keys on the first and third row are closer to the second line because of the reference effect of the three landmark keys. The mean $offset_y$ of all keys on the first row was −0.083 for *FK*, −0.176 for *K3*, and −0.072 for *L5* on the phone, and +0.037 for *FK*, −0.191 for *K3*, and +0.099 for *L5* on the watch (+ indicates an upward offset and − indicates a downward offset on a keyboard). On the third row, the mean $offset_y$ of all keys was −0.104 for *FK*, +0.033 for *K3*, and −0.102 for *L5* on the phone, and +0.020 for *FK*, +0.167 for *K3*, and +0.019 for *L5* on the watch.

Table 7 shows the weighted scores in Equation (1) of the three keyboards using the above data points. It is worth noting that on the phone, the actual scores of *K3* and *L5* are lower than that of *FK* and also lower than the estimated scores in the designing section (*K3*: 0.9787 in Table 1 and *L5*: 0.9883 in Table 3); However, on the watch, the actual scores of *K3* and *L5* are even slightly higher than that of *FK* and much higher than the previous estimated scores (*K3*: 0.8643 in Table 2 and *L5*: 0.8990 in Table 4).

6.8 Results of Revisit Session

Table 8 shows the results of the input performance with *K3* and *L5* in Stage 4, in which the participant completed two blocks of dictionary-word phrase typing and two blocks of OOV-word phrase typing after a week since Stage 1–3. Overall, the values of each metric fluctuated slightly around the previous values in Stage 1–3, confirming that participants could quickly recall the typing skills and maintain the input performance even after a few days.

However, it is worth noting that participants performed and felt differently between using *K3* and *L5*. For example, WPM of *K3* increased in all four conditions but WPM of *L5* decreased except for typing OOV-word phrases on watch. Qualitatively, 34 of 40 participants expressed that it was easier to recover the typing skills with *K3* than *L5*. Typically, P8 commented that "*When typing on K3, I could quickly recall the key locations and their relationship with the visible keys. I could even feel that I was still improving at this stage and became more familiar compared to the last time. But* *it is much harder when using L5. I had to take a long time to re-establish the spatial correspondence with the 5 lines.*"

6.9 Summary and Discussion

6.9.1 Can Users Type Well When Only a Few Keyboard Landmarks Are Visible? The answer to this question is affirmative. In summary, with limited training, the input speed and input accuracy of using *K3* and *L5* quickly raised and approached the performance of using a fully displayed Qwerty keyboard. Meanwhile, *K3* takes up only 11.5% of the keyboard area and *L5* occupies almost no space. The results confirmed our initial hypothesis that for an experienced Qwerty soft keyboard user, the display of the Qwerty layout is to a certain degree redundant and can be substantially reduced. However, the input performance and mental effort of using the two designed keyboards still had a certain decline compared to typing with a full keyboard, although very small.

One of our main motivations and contributions is to enable users access "each" key accurately on an ultra-partially-visible keyboard, which can be useful in entering OOV words. The experiment results supported this claim (Figures 17 and 18). Researchers [53] have investigated typing on a *completely invisible* and on a *partially invisible* (key labels were hidden but the boundaries were visible) keyboard on the smartphone. With a decoder of spatially adapted model + language model, participants achieved a promising input performance at 37.9 WPM and <3% WER. The authors did not report the character-level error rates. But by examining the distributions of touch points in Experiment 1 in the paper [53], we estimate the mean input accuracy of each key to be only 38.62% with the *completely invisible* keyboard and 79.39% with the *partially invisible* keyboard, based on the bivariate normal distribution with the values of offsets and variances presented in the article. In contrast, *K3* and *L5* provide >95% key input accuracy. It is interesting that *K3* and *L5* seem to outperform the *partially invisible* keyboard on key accuracy when *K3* and *L5* are essentially two proper subsets of the *partially invisible* keyboard (only several boundaries vs. all boundaries are visible). We suspect that this benefits from the reference effect of the spatial landmarks. With the *partially invisible* keyboard, it may be difficult for a user to directly locate a target key in a grid, especially the ones inside the keyboard [27, 42], for example key *G*, *H*, *J*, *C*, *V*, *B* (see Figure 3 in the paper [53]). However, with *K3* or *L5*, if the user has mastered the relative position between each key and its nearby landmark, then it will be easy to locate the target key accurately. The drawback of *K3* or *L5* is that they require a learning process. Please note that our work and the paper [53] have a different sample of participants, different experiment tasks and requirements. So the comparison above is not completely rigorous and needs further studies.

6.9.2 Differences in Input Performance on the Smartphone and Smartwatch. The results also show that the participants performed better on the smartwatch than on the smartphone. The gap between the performance of *K3*/*L5* and the performance of *FK* was smaller on the smartwatch on WPM, CCER, and UCER (see Figure 14, Tables 5 and 6). In addition, the actual scores (1) of *K3* and *L5* are at the same level as *FK* on the smartwatch. We suspect there are two main reasons for this. First, our participants had little experience of one-finger typing on the watch so that motor memory is lacking. Therefore, even typing with *FK*, the input performance is limited. Second, precisely pointing the ultra-small keys on the smartwatch is difficult due to occlusion and fat finger problem. The low absolute precision of finger touch [5] may have a greater impact on the input accuracy compared to the user's familiarity with the key location.

6.9.3 Which Keyboard Is Better, K3 or L5? After the experiment, when being asked "which keyboard do you prefer, *K3* or *L5*?" 36 of 40 participants chose *K3*. For example, P5 commented that "*The experience of using two the keyboards is different. K3 retains the 'key' element and is visually more comfortable. Also, the typing experience is in line with FK. However, L5 requires a different*

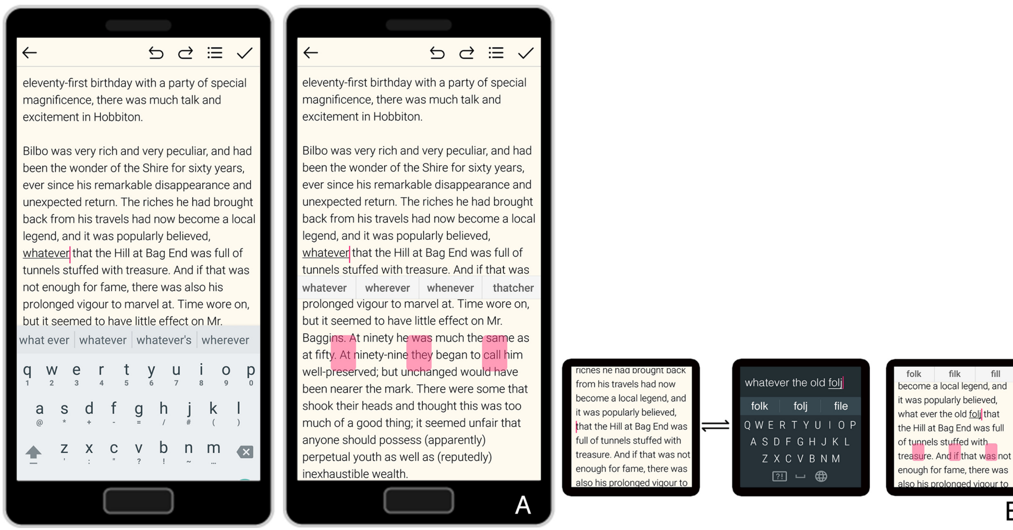

Fig. 19. The interface of the Notepad app in Study 4.

reference strategy. Although I could finally type accurately and fast, I felt like I was using another kind of keyboard, but not Qwerty." For participants who preferred *L5*, P31 commented that "*The lines in L5 provided a more strong reference effect than the key. I found myself typing more accurately with this keyboard.*" From the experiment results, the average performance using *K3* was not worse than *L5*, but even slightly better at the beginning blocks. In addition, results also showed that it was easier for participants to maintain the typing skills of *K3* than *L5*, suggesting *K3* was more suitable for long-term use. Therefore, although *L5* consumes less screen space, we follow the preferences of our participants and think of *K3* as a more intuitive, user-friendly design of low-occlusion keyboard.

6.9.4 Does the Language Model Have a Significant Impact? Our statistical decoder employs a word-level bigram language model. In our results, WER of *K3* and *L5* were at the same level as that of *FK*. This raises a question: the correction ability from the language model might be very strong so that the WER performance was artificially boosted than it should be. Our analysis shows this is not the case. After the experiment, we investigated if the correct words could be suggested by the decoder with only the spatial model but without the language model. A simulation was run based on the participants' typing events recorded in the log files. We found that in the user's input, 99.36% (*FK*), 99.26% (*K3*), and 99.42% (*L5*) of the words could be correctly recommended in the top three suggestions on the phone, and 99.21% (*FK*), 99.10% (*K3*), 99.15% (*L5*) on the watch.

7 STUDY 4: EVALUATING USER FEEDBACK ON APPLICATIONS

The goal of this study was to understand the benefits and limitations of typing with *K3* in real application contexts. We gathered user reactions and comments from participants on two applications: Messaging and Notepad.

7.1 Participants and Experiment Interface

This study was carried out right after the stage 4 of Study 3 with the same 40 participants (referred to as P1–P40). We developed two applications: Messaging and Notepad that integrated *K3* for input on both the smartphone and the smartwatch (see Figures 1 and 19). The three landmark keys of

K3 were set to rounded rectangles with an opacity of 0.3. The keyboard was implemented with all the features described in Section 5 including gesture interactions, non-alphabetical character input, automatic positioning, and statistical decoder. As a comparison, the design (e.g., location, symbol key) of the full keyboard (*FK*) was a copy of the built-in keyboard of Android OS.

7.2 Study Design and Procedure

At the beginning, the researcher briefed the participants on the experiment goals, followed by a demonstration of using *K3* and *FK* in in two applications. Then, the participants try out typing in each [*FK*, *K3*] × [Messaging, Notepad] condition.

In the Messaging app session, the 20 participants for the smartphone and the other 20 for the smartwatch were one-one paired. Each pair chatted with each other by simulating a 5-minute conversation of discussing and scheduling a one-day trip (Figure 1(a)–(b)). In the Notepad app session, participants were asked to read an article and give their own thoughts/conclusions after each paragraph (Figure 13). Participants could also edit the text freely. The Notepad app session took 10 minutes. During the experiment, participants were free to talk to the researcher, ask questions and comment on their experience while exploring. In the last 10 minutes, an interview about the benefits and limitations of *K3* was carried out. Note that in this study, when typing dictionary-words, participants could relax the requirement of input precision of each letter, relying more on the suggestions by the statistical decoder (different from Study 3, in which the goal was layout learning and participants were asked to input each key correctly at first).

7.3 Results

7.3.1 Benefits of K3. In general, participants gave positive feedback to *K3* and liked the experience of typing with *K3*. Below, we summarize some of our key observations.

First, participants liked that they were able to see more screen space on the device. This brings two benefits. (1) Because *K3* is in fact a floating window superimposed on the screen, "*the main application view is more visually stable*" (P3) rather than being squeezed by the *FK* on the phone (P3) or switched to another interface and losing the contexts on the watch (P25). (2) More screen space means "*more information can be acquired*" (P10), which leads to a more efficient and smooth input experience. For example, On the smartphone, P12 commented that "*With K3, I felt like the screen grew bigger. When I needed to find or view information in the context, I spent less effort going back and forth, especially in the Notepad app.*" On the smartwatch, the preference of *K3* was more significant because of the ultra-small display. All the 20 participants expressed that *K3* was a better choice for input than *FK*. "*The old way is that when I type, it starts a new interface only for text input. So I lose all the contexts. If I want to look at the conversation history again, I have to switch back with a lot of extra effort. K3 however offers a more concurrent and smooth input experience*" (P27).

Second, most participants liked the gesture interactions of *K3*. Participants found it intuitive and natural to use the swiping gestures to delete, leave spaces, and select words. For example, P1 commented that "*I just needed to concentrate on typing without paying attention to other function keys.*" P37 commented that "On the smartwatch, the space bar and delete button are too small. I need to click on them very carefully. The gesture interactions solve this problem very well." Participants also loved the flexible and customizable location of the keyboard, enabled by the *long-press and drag* gesture. On the smartphone, participants liked to place the keyboard near the editing location. "*This reduces the distance of my switching the line of sight*" (P3).

Third, participants did not report a perceived performance degeneration on typing with *K3*. "*Since in this study I did not have to input each letter of the dictionary words correctly, I found I could type as fast as usual because the keyboard could always suggest the intended word even when I made some mistakes. I felt more relaxed than in Study 3*" (P5).

7.3.2 Limitations of K3. On the other hand, participants also pointed out issues and concerns with *K3*.

Foremost, participants gave mixed reviews for the gestural input of punctuations and symbols. Most participants recognized this design which enables punctuation and symbol input without mode switch and visual search in a table. But there are two major limitations. (1) Using two fingers to make a gesture on the small screen of the watch is awkward and sometimes tiring (P22). (2) Using gestures cannot cover all the punctuations and symbols. For example, P12 pointed out that "*The keyboard probably cannot distinguish comma and apostrophe.*" In addition, learning to draw the symbols needs time and efforts (P40). P27 suggested using this method for only some of the most basic punctuations' input. For more complex symbols, choosing from a pop-up menu is still more intuitive and easy.

Second, although *K3* increases the display area of the screen content, it does not increase the interaction space synchronously. Users are not allowed to directly interact with the content right under the keyboard interface (e.g., placing the cursor or selecting text in the Notepad app). With our implementation, the user have the following three choices: (1) scroll the content so that the position of interest is out of the keyboard and then interact, (2) long-press and drag the keyboard away and then interact, and (3) pinch to minimize the keyboard and then interact (at the same time the keyboard appears in the proper place). We asked the participants to try out these three methods in the Notepad app. As a result, 16 of 40 participants preferred the first method and 24 preferred the third. For example, P3 who preferred the first method commented that "*Scrolling conforms to my usual habits and can keep the keyboard position.*" P15 who preferred the third method commented that "*The pinch gesture is the fastest among the three. In actual use, I may use different methods depending on the situation.*" More ideally, we can introduce multitasking interaction techniques to solve this problem. One solution is to use finger identification technique [16]. For example, use the middle finger or ring finger to interact with the contents under the keyboard.

8 CONCLUSION

The Qwerty soft keyboards on current mobile devices consume 1/3 to 1/2 of the screen space. The keyboard may squeeze the application view upwards or directly block part of the content, affecting the user experience of accessing information, and the performance of entering text. The goal of our work is to reduce the consumed area of the Qwerty layout.

Different from the previous research that deforms or rearranges the keyboard layout, we keep the original Qwerty layout and directly reduce its visualization. The idea behind is leveraging the reference effect of spatial landmarks. Because the keyboard is a stable and memory-based interface, we expect that with only a few landmarks, users can quickly re-build up spatial memory and accurately access each key. Two kinds of landmarks over the keyboard are explored: the landmark key and the landmark segment. Accordingly, we studied and optimized two low-occlusion keyboards: *K3-SGK* and *L5-EYOCN*. *K3-SGK* takes up only 11.5% of the keyboard area and *L5-EYOCN* occupies almost no space, allowing users to view more contents on the smartphone and input more fluently on the smartwatch. To our knowledge, no previous keyboard design can achieve a close result. With limited training, the input speed and accuracy of using both keyboards quickly raise and approach the performance of using a full Qwerty keyboard. In contrast, previous keyboard designs [4, 26] generally lead to a significant performance degeneration (43%–52%) on the typing speed. We further discuss the benefits and limitations of using *K3-SGK* in two real applications.

Although in this work we have considered the reduction issue of only the Qwerty soft keyboard, the idea of using the reference effect of spatial landmarks and the optimizing procedure can also be applied to other memory-based graphical interfaces. For example, hiding the application icons on the home screen of a mobile device to protect user privacy. We have also revealed that different forms of spatial landmarks (rectangle and line segment) may result in different reference strategies and effects. In future work, we plan to figure out a more rigorous model to describe the effect of landmarks on user's location perception and the model of its learning process. As mentioned above, one limitation of our optimizing procedure (Study 1 and Study 2) is the experimental design. To cover all the cases, the time of the whole user experiment was too long for each participant. In addition, the practices across the experiment had an effect on cumulatively increasing the familiarity of the key locations. These two issues might have largely affected the results of the studies. We think one solution is to use crowdsourcing, for example, assigning tasks to a large number of recruited users through the Internet where each participant completes only one letter-block. This method guarantees both the quantity of collected data and the rigor of results.

ACKNOWLEDGMENTS

The authors would like to thank Yiqin Lu and Weinan Shi for assistance with text entry techniques, and Zirui Huang for making the figures.

REFERENCES

[1] Ahmed Sabbir Arif, Benedikt Iltisberger, and Wolfgang Stuerzlinger. 2011. Extending mobile user ambient awareness for nomadic text entry. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference (OzCHI'11).* ACM, New York, NY, 21–30. DOI: https://doi.org/10.1145/2071536.2071539

[2] Ahmed Sabbir Arif and Ali Mazalek. 2016. A survey of text entry techniques for smartwatches. In *Proceedings, Part II, of the 18th International Conference on Human-Computer Interaction. Interaction Platforms and Techniques - Volume 9732.* Springer-Verlag, Berlin, 255–267. DOI: https://doi.org/10.1007/978-3-319-39516-6_24

[3] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'12).* ACM, New York, NY, 251–260. DOI: https://doi.org/10.1145/2371574.2371612

[4] Nikola Banovic, Koji Yatani, and Khai N. Truong. 2013. Escape-keyboard: A sight-free one-handed text entry method for mobile touch-screen devices. *International Journal of Human-Computer Studies* 5, 3 (2013), 42–61. DOI: https://doi.org/10.4018/jmhci.2013070103

[5] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts law: Modeling finger touch with Fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13).* ACM, New York, NY, 1363–1372. DOI: https://doi.org/10.1145/2470654.2466180

[6] Thorsten Brants and Alex Franz. 2006. *Web 1T 5-gram Version 1.* Linguistic Data Consortium.

[7] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST'14).* ACM, New York, NY, 615–620. DOI: https://doi.org/10.1145/2642918.2647354

[8] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting novice to expert transitions in user interfaces. *ACM Computing Surveys* 47, 2 (November 2014), Article 31, 36 pages. DOI: https://doi.org/10.1145/2659796

[9] Mary Czerwinski, Maarten Van Dantzich, George G. Robertson, and Hunter G. Hoffman. 1999. The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3D. In *Proceedings of the INTERACT.* 163–170.

[10] EnronData. 2008. Enron Email Datasets. Retrieved October 1, 2018 from https://enrondata.readthedocs.io/en/latest/data/enron-email-datasets/.

[11] Leah Findlater, Ben Lee, and Jacob Wobbrock. 2012. Beyond QWERTY: Augmenting touch screen keyboards with multi-touch gestures for non-alphanumeric input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12).* ACM, New York, NY, 2679–2682. DOI: https://doi.org/10.1145/2207676.2208660

[12] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (1954), 381.

[13] Brian Frey, Caleb Southern, and Mario Romero. 2011. BrailleTouch: Mobile texting for the visually impaired. In *Universal Access in Human-Computer Interaction. Context Diversity.* Springer, Berlin, 19–25.

[14] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. 2006. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'06).* ACM, New York, NY, 201–208. DOI: https://doi.org/10.1145/1133265.1133306

[15] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI'16)*. ACM, New York, NY, 3817–3821. DOI: https://doi.org/10.1145/2858036.2858242

[16] Aakar Gupta, Muhammed Anwar, and Ravin Balakrishnan. 2016. Porous interfaces for small screen multitasking using finger identification. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST'16)*. ACM, New York, NY, 145–156. DOI: https://doi.org/10.1145/2984511.2984557

[17] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster command selection on tablets with FastTap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, 2617–2626. DOI: https://doi.org/10.1145/2556288.2557136

[18] Niels Henze, Enrico Rukzio, and Susanne Boll. 2012. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*. ACM, New York, NY, 2659–2668. DOI: https://doi.org/10.1145/2207676.2208658

[19] William E. Hick. 1952. On the rate of gain of information. *Quarterly Journal of Experimental Psychology* 4, 1 (1952), 11–26.

[20] Ray Hyman. 1953. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology* 45, 3 (1953), 188.

[21] Jussi P. P. Jokinen, Sayan Sarcar, Antti Oulasvirta, Chaklam Silpasuwanchai, Zhenxin Wang, and Xiangshi Ren. 2017. Modelling learning of new keyboard layouts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, New York, NY, 4203–4215. DOI: https://doi.org/10.1145/3025453.3025580

[22] David E. Kieras and Anthony J. Hornof. 2014. Towards accurate and practical predictive models of active-vision-based visual search. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, 3875–3884. DOI: https://doi.org/10.1145/2556288.2557324

[23] KlearKeys. 2013. KlearKeys: A Transparent Keyboard on Phone. Retrieved on November 25, 2018 from https://play.google.com/store/apps/details?id=com.touchscapesoftware.klearkeys.phone&hl=en_US.

[24] Gordon P. Kurtenbach, Abigail J. Sellen, and William A. S. Buxton. 1993. An empirical evaluation of some articulatory and cognitive aspects of marking menus. *Human-Computer Interaction* 8, 1 (1993), 1–23. DOI: https://doi.org/10.1207/s15327051hci0801_1

[25] J. P. Lewis, Ruth Rosenholtz, Nickson Fong, and Ulrich Neumann. 2004. VisualIDs: Automatic distinctive icons for desktop interfaces. In *Proceedings of the ACM SIGGRAPH 2004 Papers (SIGGRAPH'04)*. ACM, New York, NY, 416–423. DOI: https://doi.org/10.1145/1186562.1015739

[26] Frank Chun Yat Li, Richard T. Guy, Koji Yatani, and Khai N. Truong. 2011. The 1Line keyboard: A QWERTY layout in a single line. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST'11)*. ACM, New York, NY, 461–470. DOI: https://doi.org/10.1145/2047196.2047257

[27] Xianyun Liu, Matthew J. C. Crump, and Gordon D. Logan. 2010. Do you know where your fingers have been? Explicit knowledge of the spatial layout of the keyboard in skilled typists. *Memory and Cognition* 38, 4 (2010), 474–484. DOI: https://doi.org/10.3758/MC.38.4.474

[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), 18:1–18:24. DOI: https://doi.org/10.1145/3090083

[29] I. Scott MacKenzie, Hedy Kober, Derek Smith, Terry Jones, and Eugene Skepner. 2001. LetterWise: Prefix-based disambiguation for mobile text input. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST'01)*. ACM, New York, NY, 111–120. DOI: https://doi.org/10.1145/502348.502365

[30] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Proceedings of the CHI'03 Extended Abstracts on Human Factors in Computing Systems (CHI EA'03)*. ACM, New York, NY, 754–755. DOI: https://doi.org/10.1145/765891.765971

[31] I. Scott MacKenzie, R. William Soukoreff, and Joanna Helga. 2011. 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of H4-writer. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST'11)*. ACM, New York, NY, 471–480. DOI: https://doi.org/10.1145/2047196.2047258

[32] I. Scott MacKenzie and Shawn X. Zhang. 2001. An empirical investigation of the novice experience with soft keyboards. *Behaviour & Information Technology* 20, 6 (2001), 411–418.

[33] Antti Oulasvirta. 2017. User interface design with combinatorial optimization. *Computer* 50, 1 (2017), 40–47.

[34] Antti Oulasvirta, Anna Reichel, Wenbin Li, Yan Zhang, Myroslav Bachynskyi, Keith Vertanen, and Per Ola Kristensson. 2013. Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*. ACM, New York, NY, 2765–2774. DOI: https://doi.org/10.1145/2470654.2481383

[35] David Peebles and Corinna Jones. 2014. A model of object location memory. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 36.

[36] Ken Perlin. 1998. Quikwriting: Continuous stylus-based text entry. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST'98).* ACM, New York, NY, 215–216. DOI: https://doi.org/10.1145/288392.288613

[37] George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. 1998. Data mountain: Using spatial memory for document management. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST'98).* ACM, New York, NY, 153–162. DOI: https://doi.org/10.1145/288392.288596

[38] K. C. Salter and R. F. Fawcett. 1985. A robust and powerful rank test of treatment effects in balanced incomplete block designs. *Communications in Statistics - Simulation and Computation* 14, 4 (1985), 807–828.

[39] Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. 2012. Improving command selection with CommandMaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12).* ACM, New York, NY, 257–266. DOI: https://doi.org/10.1145/2207676.2207713

[40] Katherine Schramm, Carl Gutwin, and Andy Cockburn. 2016. Supporting transitions to expertise in hidden toolbars. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI'16).* ACM, New York, NY, 4687–4698. DOI: https://doi.org/10.1145/2858036.2858412

[41] Wendy Ark, D. Christopher Dryer, Ted Selker, and Shumin Zhai. 1998. *Landmarks to Aid Navigation in a Graphical User Interface.* IBM Thomas J. Watson Research Division.

[42] Kristy M. Snyder, Yuki Ashitaka, Hiroyuki Shimada, Jana E. Ulrich, and Gordon D. Logan. 2014. What skilled typists don't know about the QWERTY keyboard. *Attention, Perception, & Psychophysics* 76, 1 (2014), 162–171. DOI: https://doi.org/10.3758/s13414-013-0548-4

[43] Md. Sami Uddin and Carl Gutwin. 2016. Rapid command selection on multi-touch tablets with single-handed HandMark menus. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS'16).* ACM, New York, NY, 205–214. DOI: https://doi.org/10.1145/2992154.2992172

[44] Md. Sami Uddin, Carl Gutwin, and Andy Cockburn. 2017. The effects of artificial landmarks on learning and performance in spatial-memory interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17).* ACM, New York, NY, 3843–3855. DOI: https://doi.org/10.1145/3025453.3025497

[45] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15).* ACM, New York, NY, 659–668. DOI: https://doi.org/10.1145/2702123.2702135

[46] Norman G. Vinson. 1999. Design guidelines for landmarks to support navigation in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'99).* ACM, New York, NY, 278–285. DOI: https://doi.org/10.1145/302979.303062

[47] Daniel Vogel and Patrick Baudisch. 2007. Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07).* ACM, New York, NY, 657–666. DOI: https://doi.org/10.1145/1240624.1240727

[48] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The aligned rank transform for non-parametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 143–146.

[49] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST'03).* ACM, New York, NY, 61–70. DOI: https://doi.org/10.1145/964696.964703

[50] Xin Yi, Chun Yu, Weinan Shi, and Yuanchun Shi. 2017. Is it too small?: Investigating the performances and preferences of users when typing on tiny QWERTY keyboards. *International Journal of Human-Computer Studies* 106 (2017), 44–62.

[51] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. 2017. COMPASS: Rotational keyboard on non-touch smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17).* ACM, New York, NY, 705–715. DOI: https://doi.org/10.1145/3025453.3025454

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

[53] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an invisible keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI'18).* ACM, New York, NY Article 439, 13 pages. DOI: https://doi.org/10.1145/3173574.3174013

Received April 2018; revised February 2019; accepted February 2019