TypeBoard: Identifying Unintentional Touch on Pressure-Sensitive Touchscreen Keyboard

ANONYMITY, address, Country

用户在触屏键盘上打字时，触摸屏幕即触发点击事件。因此，触屏用户不能像在物理键盘上一样通过触摸来align fingers positions[xx]，也不能将手指休息在键盘上，影响了触屏打字的效率和舒适度。在这篇论文中，我们提出了TypeBoard，一款带压力屏键盘上的防误触算法，我们也研究了用户使用TypeBoard时的打字行为。用户的打字行为和防误触能力之间是相互影响的，比如，在防误触的触屏键盘上，用户会更倾向于将手指休息在触碰上，造成更多的、更多样的非有意触摸点；而更多的、更多样的非有意触碰点会对防误触提出更高的要求。为此，我们通过迭代的数据采集和机器学习方法来设计了TypeBoard防误触算法。在一个使用TypeBoard写日记的评测实验中，用户的非有意触点个数占触点总数的xx.x%，我们的算法能在点击事件发生100 ms的时间内以xx.x%的准确率判断出其输入意图，而相关工作中基于压力和时间阈值的方法[xx]的识别准确率只有xx.x%，且必须在release以后做出判断。这份工作说明，第一，压敏触屏键盘有能力准确地、低延迟地防止用户休息、轻触等行为造成的误触。第二，用户在有效防误触的键盘上打字时，其用户行为会发生改变，比如手指休息的行为会显著增加。第三，防误触键盘和传统触碰键盘相比，能够防止用户疲劳，提高用户体验，且显著降低了输入任务的完成时间。

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Smart watch, text entry, touch input.

ACM Reference Format:
anonymity. 2018. TypeBoard: Identifying Unintentional Touch on Pressure-Sensitive Touchscreen Keyboard. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 20 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Increasingly more people are using tablets for text input tasks such as searching the Internet, writing email messages, and office work [41]. They use software keyboards on the touchscreens of the tablets. However, there is a gap between the software keyboard and the physical keyboard in aspects of the fatigue problem [19], switching of visual attention [7, 28, 38], and typing speed [14, 36] (e.g., 38 WPM on iPad vs. 69 WPM on MacBook [9]). Users can rest their fingers on the physical keyboard but cannot rest on the software keyboard, because touching the screen causes misrecognition. The resting behavior plays an important role in the usability of physical keyboards. First, it reduces fatigue [21]. Second, users align their fingers by touching the keys and achieve touch typing [10, 15, 26, 29]. Touch typists type quickly because they do not use the sense of sight to find the keys. To bridge the gap between software and physical keyboards, we proposed TypeBoard, a software keyboard that prevents unintentional touch. The TypeBoard allows users to rest their fingers on the touchscreen. Furthermore, if we add tactile landmarks on the TypeBoard through deformable screens [2], and changeable surface texture [3, 6, 25], the TypeBoard is expected to support touch typing.

Author's address: anonymity, anonymity@anonymity.com, address, P.O. Box 1212, Dublin, Ohio, Country, 43017-6221.

2476-1249/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

We are not the first to explore the detection of unintentional touch on the touchscreen keyboard. In 2013, Kim et al. proposed TapBoard [21], a touchscreen keyboard that regards "tapping actions" as keystrokes and the others as unintentional touches. Tapping actions were those touches that neither exceed a timeout threshold of 450 ms nor exceed a displacement threshold of 15 mm. Users needed to adapt to the thresholds, which led to unnatural typing behavior. TapBoard cannot judge a touch until the touch is released. Even so, the accuracy is not high (97%). To overcome TapBoard's limitations of naturalness, recognition rate, and delay, we have two strategies. First, we defined keystrokes as those touches that intend to enter words, while unintentional touches were those touches that do not intend to enter words. This is also the moral of our title, "TypeBoard". Second, we conducted user studies to understand user behavior, based on which we designed the algorithm of TypeBoard.

Techniques change user behavior. For example, users will not rest their fingers on the software keyboard unless the touchscreen prevents unintentional touches. Thus, to design TypeBoard, it is valuable to understand the users' typing behavior on the TypeBoard itself. However, how can we observe the user behavior on the TypeBoard before we have developed the TypeBoard? To solve this "chicken and egg" problem, we followed an iterative process, i.e., we developed a semi-finished TypeBoard, then analyze the user behavior on it, and finally improved the technique by using the latest dataset. In details, we conducted three user studies, each contributing to answering one of the three research questions as follows:

(1) **RQ1):** *What is the user behavior on an imaginary TypeBoard?* We conducted study one to collect users' typing data on a touchpad that has no feedback. Participants could not enter words, instead, they pretended to enter words, and imagined that the touchpad prevents unintentional touch. We found eleven categories of unintentional touches. The three most common ones were multiple finger resting, hypothenar eminence touches, and repeated touch reporting. We developed a semi-finished TypeBoard based on the analysis of user behavior.
(2) **RQ2):** *What is the user behavior on the TypeBoard?* We conducted study two to collect users' typing data on the (semi-finished) TypeBoard. We did not found new unintentional touch behavior. However, the user behavior in study two was different from study one in many aspects. For example, the average force of intentional touch was 33.78% lighter in study two because users learned that they could type without much effort, gradually making the touch lighter. We used the data in study two to improve the TypeBoard, achieving an accuracy of 98.88%, with a delay of 100 ms.
(3) **RQ3):** *What is the performance of the TypeBoard? What if there are tactile landmarks on the TypeBoard?* We conducted study three to evaluate users' typing performance on three settings, including ordinary touchscreen keyboard, TypeBoard, and TypeBoard plus (the TypeBoard with tactile landmarks). Users were expected to perform touch typing on the TypeBoard plus. 【结果未知】Results show that there is no significant difference in typing speed between regular keyboard and TypeBoard. Users tend to rest their fingers on the TypeBoard and feel more relaxed than the regular keyboard. TypeBoard plus is significantly the best in aspects of both typing speed and subjective feedback. TypeBoard plus improves the typing speed on touchscreen keyboard by xx.x%.

The contribution of this work is three-fold. First, we proposed TypeBoard, which identifies unintentional touch in text input tasks with high accuracy (98.88%) and low latency (100 ms). Second, we explored the user behavior on the touchscreen keyboard with unintentional touch prevention. We published the dataset. Third, 【结果未知】we empirically demonstrated that TypeBoard could reduce fatigue, while TypeBoard plus tactile landmarks significantly improve touchscreen keyboard in both typing speed and user experience.

2 RELATED WORK

2.1 Unintentional Touch Prevention on Touchscreen

Touch is the main input channel on touchscreen devices such as smartphones, tablets, and tabletops. However, not all contacts on the touchscreen are intended to trigger a digital response. Those touches that do not contribute to any interaction goal are known as unintentional touches [40], or namely, accidental/unwanted touches [27, 31]. If the system fails to filter out unintentional touches, it will affect the interaction's efficiency and naturalness [4, 40]. Unintentional touch will trigger unwanted responses and disrupt the user's workflow. Then, the user takes extra time to cancel the accidentally triggered operation. As time passes, the user tends to behave carefully and prudently on the device to avoid triggering unwanted touch events. For example, users of ordinary touchscreen keyboards hover their fingers over the screen to avoid accidental palm touches, which leads to the fatigue problem [39].

Though unintentional touch is not conducive to touchscreen interaction, it is inevitable. For example, the thenar eminence on the human hand will constantly contact the touchscreen during the daily use of smartphones [23]. Fortunately, we can filter out most unintentional touches by software techniques. In the literature, methods of preventing unintentional touches have been extensively studied. We introduce related work through two taxonomies: (1) use scenarios and (2) sensors.

2.1.1 Unintentional Touch Prevention over use scenarios. The definition of unintentional touch varies in different use scenarios. When the application is not limited, unintentional touches refer to those touches that do not contribute to any interaction goal [40]. It is more clear to determine the intentions of touches in specific tasks. For example, unintentional touches in the text entry task are those that do not intend to enter words.

A few techniques [31, 40] and a large amount of patents [8, 12, 18, 33, 34] identify unintentional touch over applications. Metero et al. presented guidelines to reduce the number of unintentional touches on smartphones [31]. Their filtering criteria rejected 79.6% of unintentional touches. Xu et al. identify and filter out unintentional touches on the interactive tabletop using gaze direction, head orientation, and screen contact data [40]. The accuracy was 91.3%. These approaches suffered from a low recognition rate.

In the literature, more studies investigated unintentional touch in specific scenarios. TapBoard and TapBoard2 discussed this issue under the text input task. TapBoard recognized "tapping actions" as typing behaviors, while others are unintentional touches. Tapping actions were those touches that the duration is shorter than 450 ms and the movement is within 15 mm. Users needed to adapt to the thresholds, which is unnatural. Even so, the recognition rate of TapBoard was not high (97%). TapBoard2 was an improved version that distinguished between typing and pointing actions with an accuracy of 95%. In pen and tablet interaction, unintentional touch is one of the most prominent features identified as problematic by users [4]. For example, users had to write in an uncomfortable position to avoid "palm touch". Several studies investigated the prevention of unintentional touch while inking on tablets [5, 17, 35]. Schwarz et al. used spatiotemporal touch features to train the classifier [35] that achieved the best self-reported performance, reducing accidental palm inputs to 0.016 per pen stroke, while correctly passing 98% of stylus inputs. In general, methods of unintentional touch prevention perform better when the scenario is limited.

2.1.2 Unintentional Touch Prevention over sensors. A mass of studies have been conducted to recognize unintentional input on touchscreen devices, including smartphones [23, 24, 27, 31], tablets [5, 18, 20, 21, 35] and tabletops [40]. Most techniques only leverage built-in touchscreen signals, including touchpoint information [18, 20, 21, 31, 35] and capacitive images [5, 23]. Metero et al. introduced an unintentional touch prevention method on smartphones using touchpoint patterns [31]. They proposed filtering criteria such as touch duration, position, and trajectory pattern that rejected 79.6% of unintentional touches while rejecting 0.8% of intentional touches. chwarz et al. presented a filter that distinguishes between legitimate stylus and palm touches on tablet computers [35]. They extracted features from touchpoints and used the decision forest to train a machine learning model, which reduced accidental palm inputs to 0.016 per pen stroke, and correctly passing 97.9% of stylus inputs. PalmTouch [23] is a smartphone technique that distinguishes between palm touch and finger touch. PalmTouch leveraged the touchscreen's capacitive image as input and used Convolutional Neural Network (CNN) as the method, resulting in an accuracy of 99.53% in realistic scenarios.

Other techniques enhanced the sensing ability to improve the performance [16, 24, 27, 40]. GestureOn [27] distinguishes intended gesture input from unintentional touches in the standby mode of smartphones. The user can trigger gesture shortcuts before turning on the screen. GestureOn used most smartphone sensors, including proximity sensors, light sensors, IR sensors, and Inertial Measurement Unit (IMU). The system also leveraged the pressure signal on the touchscreen, which is not popularized on smartphones yet. Based on sensor fusion, GestureOn acquired 98.2% precision and 97.6% recall. Xu et al. leveraged gaze direction, head orientation, and screen contact data to identify unintentional touches on interactive tabletop [40]. The result showed that the patterns of gaze direction and head orientation improved the accuracy by 4.3%, reaching 91.3%. In general, sensor fusion unsurprisingly improved the recognition rate of unintentional touch.

2.1.3 Summary. We suggest two viewpoints. First, in general, methods of preventing unintentional touch perform better when the use scenario is limited. In specific scenarios, we have the opportunity to prevent unintentional touches with high accuracy so that users are willing to change their behavior to acquire benefits from the techniques [4, 20, 21]. Second, sensor fusion can usually improve the recognition rate of unintentional touch. For example, several studies have proved that additional input channels such as pressure signals [27], gaze direction, and head orientation [40] provide help in recognition of unintentional touch. In this paper, we investigate the identification of unintentional touch on the pressure-sensitive touchscreen keyboard. Because we used rich sensor signals to detect unintentional touches in a specific task (text input), we expected a high recognition rate. In this situation, we were interested in a more profound question: can our proposal change the user behavior, be more natural and relaxing?

2.2 Benefits from Unintentional Touch Prevention

A direct benefit from unintentional touch prevention is to reduce the harm caused by the misrecognition, which improves efficiency and interaction naturalness. Besides this, we summarize other advantages of unintentional touch prevention as follows.

2.2.1 Bridge the Gap Between Software Keyboards and Physical Keyboards. More and more people use software keyboards on the touchscreens of the tablets [41]. However, software keyboards cannot compare to physical keyboards in usability [7, 19, 28, 38] and efficiency [9, 14, 36]. On physical keyboards, users can rest their fingers on the buttons, which is a crucial usability factor of physical keyboards. First, it reduces fatigue [21]. Second, users can input quicker through touch typing, using the sense of touch (instead of sight) to find the keys [10, 15, 26, 29]. Since the prevention of unintentional touch allows tablet users to rest their fingers on touchscreens, it reduces typing fatigue. Furthermore, on the premise that the user can touch the screen without triggering responses, we can provide tactile landmarks through deformable screens [2] or changeable surface texture [3, 6, 25], so that the users can perform touch typing. In this way, unintentional touch prevention bridges the gap between software keyboards and physical keyboards.

2.2.2 Leveraging Unintentional Touch as Input Channels. The goal of our work is to filter out unintentional touches on the touchscreen. However, some literature tried to use unintentional touches for situational awareness [43] and explicit interactions [20, 23, 32]. Zhang et al. leveraged unintentional touches such as the palm touch to augment pen and touch interactions, e.g., providing "Palm Tools" that teleports to a hand-centric location when the user plants their palm on the screen while drawing [43]. Koura et al. [22] proposed to use the forearm, which often creates problems such as incorrect recognition and occlusions, as a new input channel to manipulate menu and data storage. Matulic et al. enriched tabletop interaction by considering the whole hand as input [32]. The system detects seven different contact shapes with 91% accuracy and can be used to trigger, parameterize, and dynamically control menu and tool widgets. PalmTouch is an additional input modality that distinguishes between finger touches and palm touches [23]. PalmTouch can be used as a shortcut to improve reachability. TapBoard2 [20] distinguishes between typing and pointing actions, thereby unifying the keyboard and mouse control spaces. It depresses the burden of frequently switching between devices [13].

3 STUDY 1: USER BEHAVIOR ON AN IMAGINARY TYPEBOARD

In this study, we collected data from participants' typing actions on a pressure-sensitive touchpad. The motivation was to investigate users' typing behavior on an imaginary touchscreen keyboard that prevents unintentional touch, based on which we could design the algorithm of unintentional touch prevention. Because (incorrect) feedback would affect users' behaviors, participants typed on a touchpad without any feedback in this study. Participants could not enter words by typing on the touchpad. Instead, they imagined that the desired words are inputted. Besides, participants needed to adapt their behaviors according to the imagination that the keyboard can prevent unintentional touch.

3.1 Participants

We recruited 16 participants from the campus (aged from 19 to 26, M = 22.13, SD = 2.13, eight females). All the participants were right-handed and native Chinese speakers. They have used software keyboards on smartphones for not less than two years (M=7.50, SD = 2.25). All the participants were familiar with physical keyboards. Eight participants have ever used software keyboards on tablets.

3.2 Design and Procedure

Figure xx illustrates the experimental setting. There was a Sensel Morph [1] pressure-sensitive touchpad on the desk. We placed a Windows Surface tablet computer on the touchpad, covering half of the touchpad. We drew a QWERTY layout on the touchpad using highlighter pens. The devices were a substitute for the pressure-sensitive touchscreen, which is not available in markets yet. Participants filled in a Microsoft Word document to complete the experimental tasks. They touched on the tablet to select table cell and typed on the touchpad to pretend to enter words. The system recorded every touch and the screencast during the experiment.

【图：实验一的实验设置，包含子图：平板电脑+压力触控板=压敏平板电脑】

The experiment included four sessions of text input tasks: (1) filling in personal information, (2) short questions, (3) open-book examination, and (4) picture writing. The tasks were in Chinese. We counterbalanced the order of tasks using a balanced latin square. We did not include transcription as a task as many other text entry studies do [30, 37, 42], because our pilot study showed that participants seldom rested their fingers on the touchpad in a transcription task, resulting in low efficiency of obtaining unintentional touches. The detail of tasks is as follows:

(1) **Filling in personal information:** There was a table of ten blanks about personal information, such as name and gender. Figure 1(a) shows the examples in Chinese and the corresponding translation. This assignment represented those tasks that require frequent switching between text input and cursor control. To protect privacy, participants felt free to fill in fake information. However, participants should remember what they intended to enter, which is crucial for the subsequent process of labeling data.
(2) **Short questions:** There was a table of ten short questions, such as "the favorite color" and "the best friend". Participants were allowed to fill in a fake answer.
(3) **Open-book examination:** The exam consisted of five hard questions, such as "what is the 50th element on the periodic table?". Participants could hardly know the answers, so they needed to search the Internet.

This assignment represented the common task of browsing websites. Because the participants could not input words in the search engine, they said as they wrote, so the experimenter could replace them to enter the words.

(4) **Picture writing:** As figure 1(b) shows, participants described the picture in five sentences. This assignment represented the tasks of writing articles. Participants said as they wrote, so the experimenter could replace them to enter the words.

(a)

个人信息			
姓　　名：		性　　别：	
出生日期：		现居城市：	

Personal information			
Name：		Gender：	
Birthday：		Address：	

(b)

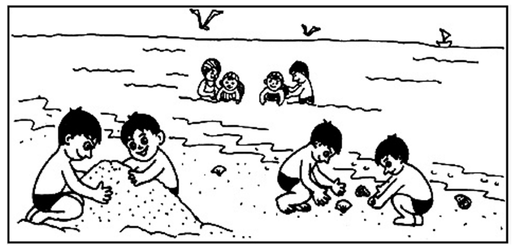


Fig. 1. The illustration of experimental tasks. The left side shows the examples of task one (filling in personal information) in both Chinese and translation. The right side is the figure we used in task four (picture writing).

Before the experiment, the participant had five minutes to get familiar with the experimental requirement. In the warm-up phase, participants typed on the touchpad freely. We reminded the participants of two points. First, the keyboard did not provide any feedback. Participants could not enter words but imagined to enter words. As the Chinese text entry method involved word selection, participants assumed that the desired word is always the first in the candidate list. Second, users needed to imagine that the keyboard prevents unintentional touches and adjust their behavior according to this assumption. For example, they could rest their fingers on the keyboard while thinking.

After each session, the participant labeled the data through an interactive program, which showed the touchpad capacitive images and the tablet screencast at the same time. As figure xx shows, there were red points on capacitive images that showed the touchpoints. Participants labeled the intended touches as green points. Participants were able to judge most touches because they could get context from the screencast. If participants were not sure, they labeled the touchpoints as blue points to remove the data. On average, participants spent five minutes finishing the text input tasks and spent 45 minutes labeling the data. Participants rested for five minutes between two sessions to avoid fatigue. The study lasted for 70 minutes.

【图：标注程序】

3.3 Apparatus

As figure xx shows, we placed a Windows Surface tablet computer and a Sensel Morph pressure-sensitive touchpad [1] together as a substitute for the pressure-sensitive touchscreen. The Sensel Morph senses the position and the pressure level of touches. The Sensel Morph contains 185 x 105 sensor elements ("sensels") at a 1.25mm pitch. Each contact can sense approximately 30000 levels, ranging from 5g to 5kg. The maximal frequency of the Sensel Morph was 125Hz (8ms latency). We slowed the frequency down to 50Hz to fetch stable data. The Sensel Morph provides capacitive images and touchpoint information, including position, timestamp, touch area, pressure level, and shape. The device is so sensitive that almost every touch is recognized. Thus, in this paper, we identified unintentional touches among reported touchpoints while ignoring the Sensel Morph's missing touches.

The size of the sensing area on the Sensel Morph was 240 mm x 138 mm. We used highlighter pens to draw a Qwerty layout on the touchpad, as figure xx shows. The Sensel Morph width (240mm) was shorter than the keyboard on a 15 inches MacBook (270mm). We removed some keys that are less frequently used, such as square brackets and the semicolon, so that the Qwerty layout could be placed in the Sensel Morph, while each key's size remained the same as Macbook. Because the Qwerty layout is changeable in the software keyboard, we did not leverage the layout as prior knowledge to recognize unintentional touch. The tablet computer was Windows Surface Pro6, with i7 Intel Core Processor. The program ran on the tablet at 50 FPS.

3.4 Result

The dataset contains 12659 touches, excluding the ambiguous touches (0.18%) in the labeling process. 67.5% of the data were positive samples (intentional touches), while 32.5% were negative samples (unintentional touches). Based on the data, we developed the TypeBoard in three steps, each contributing to solving one of the critical problems as follows:

(1) **Model V1):** *data processing*. There were some mislabeled data points because some participants misunderstood the concept of unintentional touches. We developed a naive Support Vector Machine (SVM) model to identify unintentional touch. If there was a vast difference between the predicting results and the labels, we asked the participants to relabel the suspicious data points through e-mails.
(2) **Model V2):** *filtering multiple fingers resting*. Observation suggested that multiple fingers resting was the most frequent (> xx%) unintentional touch, where users rested more than two fingers on the screen. We added targeted criteria into the SVM model to filter out multiple fingers resting.
(3) **Model V3):** *understanding the user behavior.* We analyzed the fail cases of Model **V2** to understand those user behaviors that challenged the algorithm, based on which we further improved the SVM model.

Table xx shows the feature vector we used to train the SVM model in each step and the recognition results. In the remainder of this subsection, we introduced the unintentional touch identification model in detail.

3.4.1 Model V1: naive model for data processing. In the first step, we trained a naive SVM model to identify unintentional touch using straightforward features. We sampled the first five frames (100 ms) of each touch. If the touch duration is shorter than five frames, we sampled the whole touch. As table xx (**V1**) shows, we extracted features from the samples as follows: for the contact area, ellipticity, displacement, force, and intensity over frames, we calculated the temporal features, including maximum, minimum, mean, skewness, and kurtosis. Then, we concatenated these values to obtain a feature of 25 dimensions and trained an SVM binary classifier, namely Model **V1**. We balanced positive and negative samples in weight.

We used the model to simulate the dataset. We found that some participants misunderstood the concept of unintentional touches, for example, regarding incorrect character inputs as unintentional touches. We asked the participants to relabel the suspicious data points through e-mail. After the label correction, we had xx data points, including xx.x% positive samples and xx.x% negative samples. Leave one out cross-validation shows that the accuracy of Model **V1** was 96.86% (SD=4.17%).

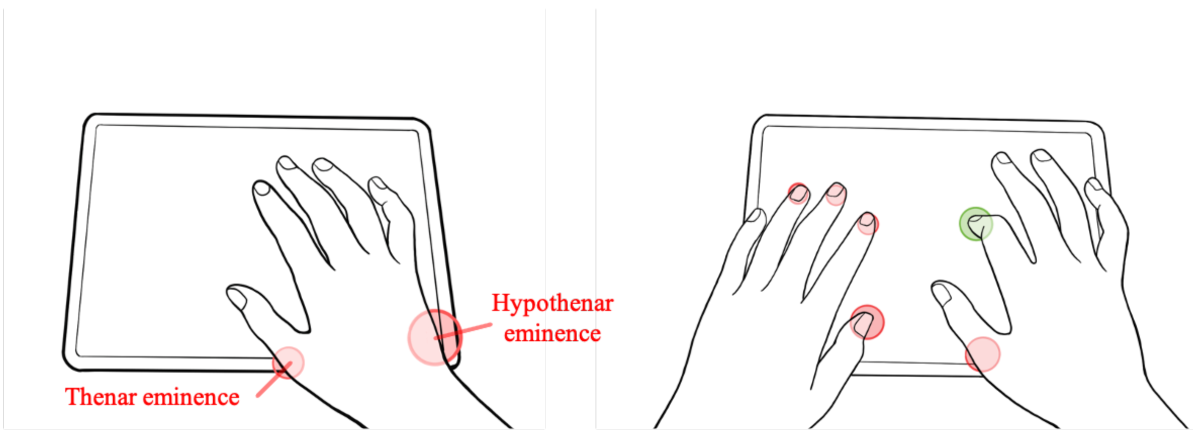
3.4.2 Model V2: filtering multiple fingers resting. Observation showed that most unintentional touches (xx.x%) were caused by multiple fingers resting, where users rested no less than three fingers on the screen simultaneously. The resting fingers contact the screen one after another. After the first touch, the following touches come within 100 ms in most instances. In xx.x% cases, the second finger touches within 100 ms, while in xx.x% cases, the third finger also touches within 100 ms. Because our model has a latency of 100 ms, the model has a big chance to realize that multiple fingers are resting. Thus, we could design targeted features to filter out the unintentional touches caused by multiple fingers resting. We added a series of features in Model **V2** to deal with this problem. The criterion was the proportion of the touch's pressure, intensity, and contact area among all touches. Table xx (**V2**) shows the details. Leave one out cross-validation shows that Model **V2** increased the recognition accuracy to 98.59%.

Table 1. The features we fed into the SVM model and the accuracy among model versions. For the features except "the relationship to recent/nearby touches", we extracted the temporal features over frames, including maximum, minimum, mean, skewness, and kurtosis.

Version	Criterion	Feature	Introduction	Accuracy
V1	The information of current touch	Contact area	The contact area in pixels.	96.86% (SD=4.17%)
		Ellipticity	The contact region is fitted as an ellipse. The ellipticity is the ratio of the minor axis to the major axis.	
		Displacement	The distance to the starting location.	
		Pressure	The contact force in grams.	
		Intensity	The ratio of the pressure to the contact area.	
V2	The proportion	Pressure proportion	The ratio of the pressure to the total pressure of all touches.	98.59% (SD=1.46%)
		Intensity proportion	The ratio of the intensity to the total intensity of all touches.	
		Area proportion	The ratio of the contact area to the total contact area of all touches.	
V3	The location of current touch	Distance to edges	The minimum distance to one of the three edges, including the bottom and the two flanks.	**99.07% (SD=0.71%)**
		Distance to corners	The minimum distance to one of the two corners, including the bottom left corner and the bottom right corner.	
	The relationship to recent/nearby touches	Recent touches	The relationships between the current touch and the last five touches in five seconds. The relationships include (1) the interval betwen their start times, (2) the average distance, and (3-5) the ratios in terms of contact area, force and intensity. If there are less than five touches, complete the feature vector with zeros.	
		Nearby touches	The relationships between the current touch and the five nearest touches within the liftcycle of the current touch. If there are less than five touches, complete the featue vector with zeros.	

3.4.3 Model V3: understanding the user behavior to improve the model. We analyzed the fail cases of Model **V2** to understand those user behaviors that challenged the model. The error rate of Model **V2** was 1.41%, including xx.x% false positives and xx.x% false negatives. The false positives referred to the misrecognized unintentional touches, while the false negatives were the missing intentional touches. As table xx shows, we classified the fail cases into 16 categories. We counted each kind of fail case, discussed the reasons, and gave possible solutions. For the fail cases with a white background in the table, humans (the researchers) could judge their intentions without extracting contextual information from the experiment screencast. That is, the machine should have correctly predicted if it is as smart as a human. For these cases, we proposed features to improve the model. Here are some examples.

(1) **EG1):** *Hypothenar eminence.* As figure 2(a) shows, the hypothenar eminence refers to a group of muscles of the palm that control the motion of the little finger, while the thenar eminence is the group of muscles on the palm at the base of the thumb. Both the two eminences may contact the touchscreen when a user is typing. In particular, the touches caused by the hypothenar eminence are usually heavy and intensive, which is easy to confuse with intentional touches. Fortunately, these touches are in the bottom left and the bottom right corners. Thus, the distance to the closest corner could be a powerful feature to reject these unintentional touches.
(2) **EG2):** *Continuous touches.* When a user continuously typed on the same key (e.g., the delete key), the following touches were lighter than the first touch (F,p). Among the continuous touches, the average pressure of the first touch was xx.x g, while the average pressure of the following touches was xx.x g. Because the following touches are light, they are likely to be recognized as unintentional touches. Information of recent touches help to correct this fail case.
(3) **EG3):** *One-hand typing.* As figure 2(b) shows, sometimes the participant typed with one hand while resting the other hand on the touchscreen. In this situation, the Model **V2** mistakenly believed that all the touches were unintentional touches caused by the multiple fingers resting behavior. We added information of nearby touches to correct this fail case because the typing finger is usually far away from the resting fingers.



(a) Thenar/Hypothenar eminence touching

(b) One-hand typing

Fig. 2. The examples of fail cases. The fail cases reveal those user behaviors that challenged the algorithm.

However, for the fail cases with gray background in table 2, humans (the researchers) could judge their intentions without watching the experiment screencast. We deemed that these fail cases are inevitable because the machine can not know what the user will enter in advance. Here are some examples. First, sometimes the participant rested one finger on the touchscreen heavily, indistinguishable from an intentional touch. Second, the participant performed a very light touch during entering a word, which is indistinguishable from an unintentional touch. Thus, the model's accuracy has a certain upper limit (roughly xx.x% in this dataset), while our goal is to approach it.

According to the user behaviors summarized in table 2, we added two criteria in the model training. The first one is the location of the touch, including the minimum distances to the edges and the corners. We did

Table 2. The fail cases of Model **V2**. The "N" column refers to the counting of each case.

Cases	**Introduction**	**N**	**Helpful features?**
False Positives			
Hypothenar eminence (figure 2a)	The hypothenar eminence usually contacts the screen while typing.	23	Touchpoint location, e.g., nearing the corners?
Thenar eminence (figure 2a)	The thenar eminence usually contacts the screen while typing.	2	Touchpoint location, e.g., nearing the bottom edge?
Repeated reporting (spatial)	A touch is misrecognized as two touches when the contact area is large.	12	Info. of nearby touchpoints.
Repeated reporing (temporal)	A touch is misrecognized as two touches if the touch is nearly released midway.	3	Info. of recent touchpoints.
Edge touch	Users trigger unintentional edge touch when adjusting the placement of devices.	7	Touchpoint location, e.g., nearing the flanks?
Two fingers resting	The user rests two fingers on the screen.	3	Info. of nearby touchpoints.
Extra touchpoint (light)	When inputting, users trigger extra touchpoints, which are lighter than the recent intentional touches.	9	Is this touch lighter than the recent touches?
Slide	A slide is less likely to be an intentional keystroke.	7	Touchpoint displacement.
One finger resting	Users rest one finger on the touchsreen, which is indistinguishable from an intentional touch.	9	No solution.
Extra touchpoint (heavy)	When inputting, users trigger extra touches heavily, which seems like an intentional touch.	15	No solution.
False Negatives			
Continuous touches	When a user continuously type on the same key (e.g., the delete key), the following touches will be lighter.	13	Info. of recent touchpoints.
Rollover-typing	The next key is pressed before the previous is released [11].	12	info. of recent touchpoints.
One-hand typing	The user types with one hand, while the other hand is resting on the screen. This case is similar to multiple fingers resting.	6	info. of nearby touchpoints.
Palm touch	The use types a key when the palm is resting on the screen. If the palm touch is detected as multiple fingers, this case is similar to multiple fingers resting.	5	info. of nearby touchpoins, e.g., is this touch near other touches?
Light touch	The very light but intended touch, which is indistinguishable from an unintentional touch.	46	No solution.
Small contact area	The intended touch with a very small contact area. This seems like an unintentional touch.	6	No solution.

not leverage the prior knowledge of the keyboard layout because the software keyboard layout is changeable, while we wanted a universal model. The second criterion is the relationships between the current touch and the recent/nearby touches. The features in detail are in table xx (**V3**). The Model **V3** used all the features in table xx. We concentrated these features to form a vector of 100 dimensions and trained an SVM binary classifier. Leave one out cross-validation showed that the accuracy was 99.07%. The error rate was 0.93%, including xx.x% false positives and xx.x% false negatives. So far, we have obtained a model with high recognition accuracy. However, as we said in the introduction, the data collected in study one did not represent the user behavior on the software keyboard with unintentional touch prevention, so we named Model **V3** as semi-finished TypeBoard. In study two, we collected the user behavior on the semi-finished TypeBoard and then used the new data to improve the model.

4 STUDY 2: USER BEHAVIOR ON THE TYPEBOARD

In this study, we obtained users' typing data on the TypeBoard, a touchscreen keyboard with unintentional touch rejection. Participants used the TypeBoard developed in study one. The motivation was to investigate users' typing behavior, based on which we can further improve the identification of unintentional touch.

4.1 Participants

We recruited 16 participants from the local campus (aged from xx to xx, M = xx.xx, SD = xx.xx, xx females). All the participants were righted handed and did not took part in the first study. They have used software keyboards on smartphones for not less than xx years. xx of the 20 participants were familiar with software keyboards on tablets.

4.2 Design and Procedure

As figure xx shows, the experimental devices were the same as those in study one, including a Windows surface tablet and a Sensel Morph force-sensitive touchpad. The working frequency was 50 FPS. There were four sessions of text entry tasks, which also reminded the same as study one. The differences in study two are as follows. First, participants could actually enter words by typing on the touchpad. Second, the system supported unintentional touch rejection. Intentional touches would trigger keystrokes and audio feedback, while unintentional touches were ignored.

【图：实验二的设置图，突出实验二中】

Before the experiment, participants warmed up by a ten-minute daily writing. Because no participant had experience with typing on a keyboard with unintentional touch rejection, we reminded the participants that they could rest their fingers on the keyboard while thinking. The resting behavior was not mandatory. Participants could decide to rest their fingers or not according to the task and their preference.

After finishing each session, participants labeled the data through the interactive program introduced in study one. During the labeling phrase, we compared participants' labels with the model predictions. When the two results were different, we immediately recorded and analyzed the data point. If the experimenter could not give reasons for the difference, he discussed with the participant. In average, participants spent 10 minutes to finish the text entry tasks and spent one hour to label the data. It took more time than study one, because participants needed to enter correct words in study two. Participants rested for five minute between two sessions to avoid fatigue. The study was generally completed within 90 minutes.

4.3 Result

The dataset contained 13789 touches, excluding the ambiguous touches (0.22%) in the labeling. After the double-check of labels, the dataset consisted of 71.01% positive samples (intentional touches) and 28.99% negative samples (unintentional touches). Using the machine learning model developed in study one, the TypeBoard predicted with an accuracy of 98.05% (SD=1.51%) in study two. There were 0.39% (SD=0.37%) false positive and 1.56% (SD=1.37) false negative predictions. In study two, participants encountered 2.20 unrecognized touchpoints and 0.55 false triggering touchpoints every 100 keystrokes in study two.

Compared with study one, this study did not find new cases of unintentional touches. However, the user behavior in study two was different from the last study. The differences included but are not limit to four examples as table xx shown. First, the average force of intentional touches was significantly lighter in this study. This may be because in real typing with feedback, users found that they could type letters without much effort, gradually making the touch lighter. 显著性分析是否支持该结论？ Second, the multiple fingers resting was more frequent in this study. This result indicated that the TypeBoard gained users' trust. Third, there were more continuous touches, where a touch is close to the last touch in both time interval (< 500 ms) and distance (< 0.5 key width). This is because participants need to remove incorrect words by continuously pressing the delete key in real typing task. Besides, some participants used the right key to select the desired work in the candidate list. Forth, there were fewer rollover-typing, where the next key is pressed before the previous is released. Participants typed slower in study two because they needed to enter correct words. This slower typing speed correlated with the fewer number of keystrokes typed with rollover [11].

Table 3. The differences of user behavior between the two studies. We use t test to evaluate the significance of the difference. If Levene's test rejects the homoscedasticity of data, we use unequal variances t test instead.

Measure	Introduction	study one	study two	Levene's test	T test
Average touch pressure	The average touch pressure of intentional touches in grams.	188.39g (SD=64.72g)	124.75g (SD=60.71g)	stat=0.54, p=0.47	stat=2.78, p=0.0094
Multiple finger resting	The number of touches that caused by multiple finger resting as a percentage of all unintentional touches.				
Continous touches	The number of continous touches as a percentage of all intentional touches.	4.02\% (SD=1.96\%)	11.89\% (SD=4.41\%)	stat=4.49, p=0.042	stat=3.57, p=0.0012
Rollover-Typing	The number of rollover-typing as a percentage of all intentional touches.	17.60\% (SD=9.34\%)	7.73\% (SD=5.22\%)	stat=6.53, p=0.016	stat=-6.32, p=0.0000

【图：用户的误触行为分类及出现频次】

It was valuable to explore the user behavior by counting the unintentional touches. Figure xx shows the frequencies of each kind of unintentional touch. The frequency was counted by the number of touches, e.g., a five finger resting behavior was counted five times. Here are some basic discoveries. First, the three most frequent unintentional touches were multiple finger resting (xx.x%, SD=xx.x%), hypothenar eminence touching (xx.x%, SD=xx.x%) and extra touchpoint (xx.x%, SD=xx.x%). Second, 0.31% of the touches were unintentional touches that humans (the researchers) could hardly identify. These cases were extremely hard to predicted by the machine learning model.

新增：个体用户之间的差异问题。

新增：用户在哪些任务中更愿意将手指休息在键盘上。

新增：与baseline（阈值方法）的比较。

We replaced the training set of the machine model with the dataset in study two and retrained the model. Leave one out cross-validation shows that the accuracy increased to 98.88%(SD=0.73%), significantly surpassed the model in this last study (F, p). There were 0.66% (SD=0.58%) false positive and 0.46% (SD=0.45%) false negative. In average, TypeBoard users will encounter 0.65 unrecognized touchpoints and 0.93 false triggering touchpoints every 100 keystrokes. The unrecognized touchpoints were significantly fewer after the training set was replaced by the new dataset (F, p). This is mainly because users pressed significantly lighter.

值得注意的是，在我们这个能代表自然用户行为和全面的文本输入任务的数据集上，先前工作TapBoard[xx]的识别准确率会下降到xx.x%，因此我们这份工作对这个技术问题的提高是非常巨大的。

4.4 Discussion

4.4.1 The iterative method. A lightspot of this paper is the iterative process to solve the problem, i.e., we developed a semi-finished TypeBoard, then conducted user experiment on it, and finally improved the technique by using the latest dataset. Because the relationship between a technique and the user behavior on it is a "chicken and egg" problem, most previous studies explored user behaviors through experiments on devices with no feedback (like our study one) [xx] or simulative feedbacks [xx]. We argue that the iterative method deserves more attention. In our work, the user behaviors we observed in study two (with feedback) are different from those in study one (without feedback). The model trained by the latest dataset also performed better. That is, the iterative process improved our technique and helped to gain a more practical model of user behavior.

4.4.2 The variety of tasks. Results show that the task had a significant effect on users' amount of unintentional touches. The percentages of unintentional touches among the four tasks were 29.73% (SD=23.71%), 29.18%(SD=22.06%), 18.02%(18.88%) and 17.62%(SD=13.76%). Bonferroni-corrected post-hoc tests showed significant differences between the following task pairs: 1-3 (p<), 1-4 (p<), 2-3 (p<), 2-4 (p<). This result indicates that user performed more unintentional touches on the task with more frequent switching between pointing and typing. We argue that the variety of tasks is important but usually neglected in studies of unintentional touch [xx]. In particular, it was improper to evaluate an unintentional touch rejection method on a transcription task [xx], because users seldom rest on the touchscreen in such a task, which will be proved in our study three.

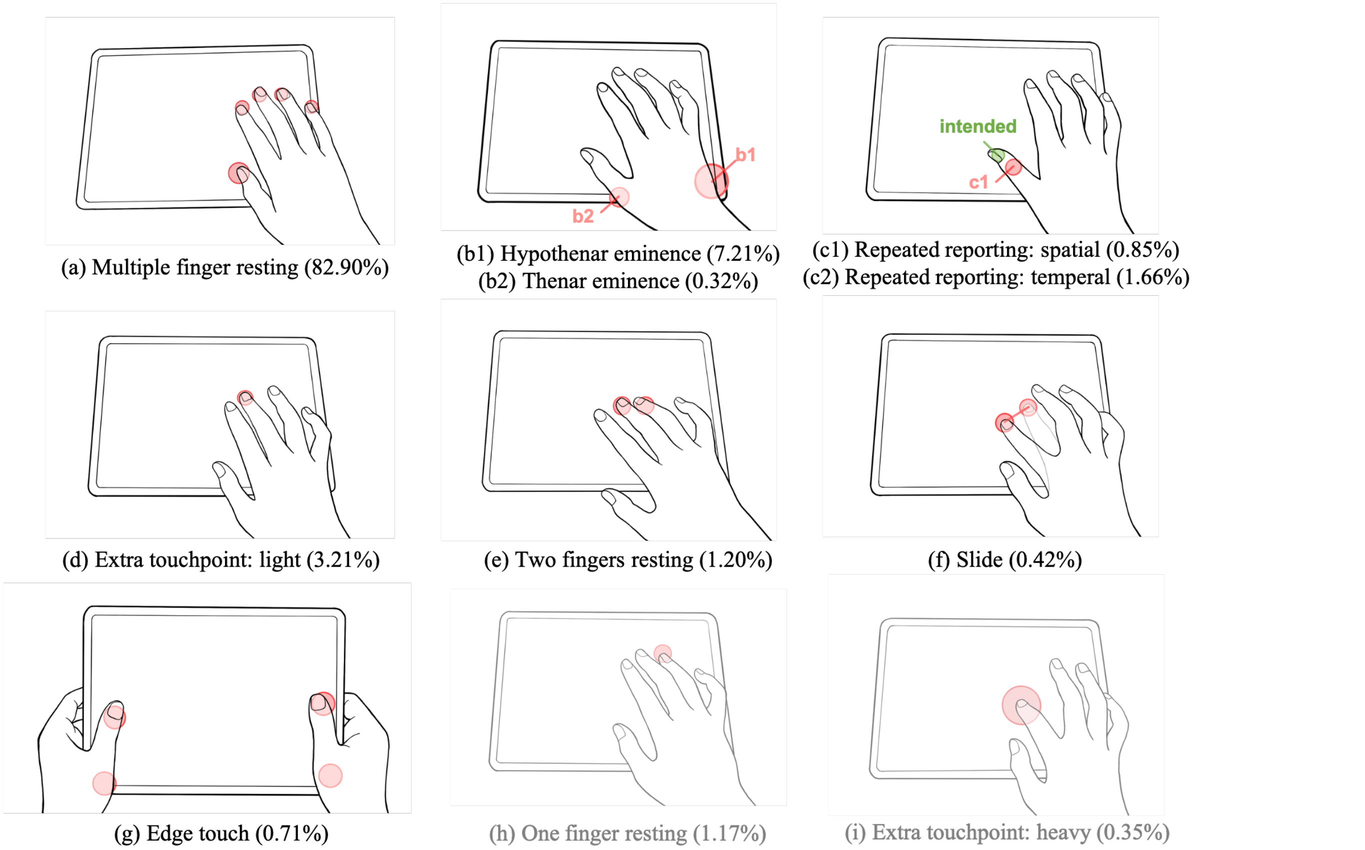
4.4.3 Can our model work with fewer sensors? The pressure signal on touchscreen is important to identify unintentional touch. However, most touchscreen devices have no pressure sensor yet, while a few device have four pressure sensors in the corners (e.g., the force touch trackpad on MacBook). To explored the feasibility of using our method on existing devices, we evaluated TypeBoard in four hardware settings.

(1) **Capactive-only:** The commonly used touchscreen devices have capactive signals, but do not have pressure signals. To evaluate our method on these devices, we removed all the features refer to pressure signals and retrained the model.
(2) **Total pressure:** The MacBook trackpad has four pressure sensors in the corners and provides developers the total pressure of all touches. To used our model, we estimated the pressure of each touch as the product of total pressure and the contact area proportion of the touch.
(3) **Pressure-enabled:** In the future, touchscreen devices may provide high resolution pressure signals. This is the experimental setting in our paper.

【图：三种硬件设置下算法的识别准确率，再用圆饼图画出误触和未识别的比例】

Figure xx shows that performances of our method among the three hardware settings. ANOVA shows a significant effect of hardware on the recognition accuracy (F, p). Bonferroni-corrected post-hoc tests showed significant differences between all hardware pairs: 1-2 (p<), 1-3 (p<), 2-3 (p<). Results show that the touchscreen devices with total pressure signal reach a balance between recognition accuracy and hardware cost.

Error rate (SD) / False Positive (SD) / False Negative (SD) (1) 0.02028 0.01537 0.01262 0.01212 0.00766 0.00619 (2) 0.01392 0.01158 0.00876 0.01016 0.00516 0.00542 (4) 0.01115 0.00726 0.00659 0.00578 0.00456 0.00451



(a) Multiple finger resting (82.90%)
(b1) Hypothenar eminence (7.21%)
(b2) Thenar eminence (0.32%)
(c1) Repeated reporting: spatial (0.85%)
(c2) Repeated reporting: temperal (1.66%)
(d) Extra touchpoint: light (3.21%)
(e) Two fingers resting (1.20%)
(f) Slide (0.42%)
(g) Edge touch (0.71%)
(h) One finger resting (1.17%)
(i) Extra touchpoint: heavy (0.35%)

Fig. 3. 各种可能的误触类型，和他们的频次，括号中的是该误触类型占总误触数量的比例。注意：1. 灰色(h,i)是理论不可分的；2. (d)Extra touchpoint和(h)单指resting的区别。3. c2没有被图示出来，具体是xxx的意思。4. 虽然双指/多指休息中，两个手指不可能是同时点上去的，但手指依次点下去的时间间隔大概率在100ms以内。

5 STUDY 3: EVALUATION ON TYPEBOARD

The motivations of study three were two-fold. First, we compared the performance and user experience of the TypeBoard and the baseline. The baseline was the regular software keyboard without unintentional touch rejection. Second, as we introduced in related work, tactile landmarks on keyboards improve users' typing speed by enabling touch type, so we investigated the feasibility of TypeBoard plus tactile landmarks in this study. In summary, we evaluated users' typing performance on three settings: (1) regular software keyboard, (2) TypeBoard, and (3) TypeBoard with tactile landmarks.

5.1 Participants

We recruited 15 participants from the campus (aged from xx to xx, M = xx, SD = xx, xx females). All the participants were right handed. They have used software keyboards on smartphones for not less than xx years. xx of the 15 participants were familiar with tablet keyboards. Participants did not take part in the last two experiments.

5.2 Design and Procedure

The study followed a within-subject design to compare users' typing speed in three keyboard configurations. As figure xx shows, the participant sat on an office chair. He was able to adjust the chair to a comfortable position. The participant typed on the pressure-sensitive touchpad to enter words, and received visual feedback from the tablet. There were three settings of the pressure-sensitive touchpad in the experiment as follows:

(1) **Config. 1):** *Regular Software Keyboard.* All contacts on the touchscreen are recognized as keystrokes. Users need to hang their wrists in the air to avoid unintentional touches.
(2) **Config. 2):** *TypeBoard.* TypeBoard is a software keyboard with unintentional touches rejection. The system only recognized intentional touches as keystrokes. Users can rest their hand on the keyboard.
(3) **Config. 3):** *TypeBoard plus.* "TypeBoard plus" refers to the TypeBoard plus tactile landmarks. To afford tactile landmarks on TypeBoard, we attached 0.05mm thick stickers on the touchpad to simulate physical keys. There were small bumps on the F and J keys, which is the same as the physical keyboard. Users could align their fingers without having to look at the keyboard.

【图：实验三要对比的三种实验设置，大图是用户实验的整体环境，大图中包含三个小图显示三种键盘配置】

There were five sessions for each of the three keyboard configurations. In each session, participants transcribed a Chinese paragraph on a typing speed measurement website [xx]. We randomly selected the task paragraphs in the website. Participants were asked to input as fast and accurately as possible. The transcription task is widely used in text entry researches [xx,xx,xx] to evaluate the ceiling typing speed. We counterbalanced the order of keyboard configuration using a balanced latin square.

Participants had five minutes to warm up before they used each keyboard. They transcribed two paragraphs to get familiar with the keyboard. The task phrases in the training step would not appear in the formal experiment. Participants rested for five minutes between sessions to avoid fatigue. In average, participant spent xx minutes to complete the experiment.

5.3 Reslut

A Repeated Measures (RM) ANOVA was conducted for text entry speed, Uncorrected Error Rate (UER) and Corrected Error Rate (CER). The within factor was the keyboard configuration. As UER and CER violated the normalcy, we used the Aligned Rank Transform [xx] for correction. If any independent variable had significant effects ($p < 0.05$), we used Bonferroni-corrected post-hoc tests for pairwise comparisons.

5.3.1 Speed. We measured text entry speed in Chinese characters per minutes (CPM). Participants used Pinyin [xx-wiki], a phonetic spelling system in Roman characters to input Chinese characters. To enter a Chinese character, users type the Pinyin of the desired character (2 - 6 letters) and then select the target from a candidate list. Users can also type the Pinyin of a Chinese word, which consists of two to four characters, and then select the word at one time. In short, the process of entering a Chinese character is similar to inputting an English word with word prediction/correction. We measured typing speed in CPM with this formula:

【中文文本输入速度公式】

where $|S|$ is the length of the transcribed paragraph in characters (including punctuation), and T is the complete time, i.e., the elapsed time in seconds from the first to the last touch in the task. All time consumption, including the time of selecting candidates, was taken into account.

Figure xx shows the speeds over sessions. The performance on regular keyboard starts with a speed of xx.xx CPM (SD = x.xx) and ends with a speed of xx.xx WPM (SD = xx.xx). xxx. xxx. 1. 显著性2. 组间显著性3. 对显著性的讨论。

【图：实验三中各种实验设置下用户输入速率随着session增长而增加的图】

5.3.2 Error rate. Two metrics were used to measure text entry accuracy: (1) Uncorrected Error Rate (UER) - text entry errors which remain in the transcribed string. UER is the number of uncorrected erroneous Chinese characters divided by the number of correct and erroneous characters. (2) Corrected Error Rate (CER) - text entry errors which are fixed (e.g., backspaced) during entry. CER is calculated by the number of corrected erroneous Chinese characters divided by the number of correct and erroneous characters. The corrections of Pinyin during inputting a word were not taken into account of CER. As UER and VER violated the normalcy, we used the Aligned Rank Transform for nonparametric factorial analysis [xx].

Figure xx shows the UER and the CER over sessions.

significance.

discussion (explanation of significance).

【图：各实验设置下UER和CER随着session增长而增加的图】

5.3.3 Unintentional touches. For the three configurations regular keyboard, TypeBoard and TypeBoard plus landmarks, the proportions of unintentional touches were xx.xx%, xx.xx% and xx.xx% respectively.

English.

为了方便计算，我们在上述统计中假设算法100%正确预测了点击的有意性。

统计发现，显著性影响。

讨论造成显著性的原因。

在实验三中，TypeBoard上的误触与实验二相比显著降低，我们认为，这是任务不同造成的。在誊写这种快速输入的任务下，用户在普通触屏键盘和TypeBoard上都没有必要将手指休息在键盘上。但在TypeBoard+tactile设置下，用户在誊写任务下仍然引发了大量的“误触”，这暗示，用户可能并不只是将手指休息在触屏上，而是尝试利用触屏上的纹理来对齐他们的手指，从而达到部分的盲打。

5.3.4 Touch position. Figure xx illustrates the multiple finger resting behavior through point clouds. The distribution seems optional on the ordinary TypeBoard, and seems regular on the TypeBoard plus tactile landmarks. xx.x% of the resting touchpoints laid on the second row of keys on the TypeBoard with landmarks, which has an significant difference (F, p) from the ordinary TypeBoard (xx.x%). This indicates that users leveraged the tactile landmarks to align their fingers.

【图：用户在multiple finger resting时的手指点云，TypeBoard with/without landmarks】

Figure xx shows the distribution of intentional touchpoints over keyboard configurations. We used xxx to cluster the point cloud of each key. xxx shows that the point cloud obeyed the 2D Gaussian distribution (F, p). The average standard deviation of the distributions were xx.x mm (SD=), xx.x mm (SD=) and xx.x mm (SD=). RM-ANOVA shows that the keyboard has a significant effect on users' touching accuracy (F, p). Users typed more accurately on the TypeBoard plus landmarks compared with the regular keyboard (p) and the TypeBoard (p). The analysis of touch position shows that users aligned their fingers on the TypeBoard with tactile landmarks, which improved their typing accuracy.

【图：各实验设置下用户有意点击点云】

5.3.5 Subjective Rating and Feedback. English.

物理负担（疲劳程度），心理负担，主观输入速度，主观输入准确率（误触准确率，而非选词准确率）。

5.4 Discussion

5.4.1 TypeBoard vs. Regular Keyboard. Compared with regular touchscreen keyboards, the TypeBoard has the advantages of avoiding fatigue and improving subjective user experience.

（1）避免疲劳。TypeBoard用户在誊写任务的每100次打字行为中，系统就会阻止xx.x次多指休息行为和xx.x次小鱼际点击。实验二表面，TypeBoard在其它需要更多键鼠切换或者思考的文本输入任务下，所

阻止的多指休息次数会更多（xx.x 每100次点击）。结果表明，用户在TypeBoard上会主动地利用可以将手指和手腕休息在touchscreen上的特性。而且在用户的主观反馈当中，用户也表示TypeBoard显著地降低了他们的疲劳程度。（2）主观用户体验。TypeBoard在主观输入速度、主观输入准确率等方面都显著优于普通键盘。这说明TypeBoard可以提高用户体验。

5.4.2 *TypeBoard vs. TypeBoard plus.* English

TypeBoard plus 是我们设想中的一种软键盘形态，即利用可变形触屏[xx]或者添加layout[xx]的方法来达到提供触觉landmarks的目的。我们发现，TypeBoard plus不仅能够降低疲劳、提高用户体验，还能够显著地提高软键盘上的文本输入速度，将输入速度提高xx.x%。输入效率提高的主要原因是用户可以在TypeBoard plus中align fingers，从而实现盲打。

6 DISCUSSION

6.1 Why sample five frames in each touch?

There is a trade-off between the amount of sampling frames and the accuracy. The more data we sample in each touch, the more accuracy the prediction is. However, a long sampling window means a large delay, which affects the user experience. We needed to strike a balance. Five frames of sampling results in an acceptable prediction accuracy (xx.xx%), meanwhile the delay of 100ms is not perceivable in the touching task [xx].

【图：不同延迟的准确率比较】

6.2 Why not deep learning?

In this paper, we used classical machine learning methods (SVM) to solve the problem. We did not used deep learning for two reasons. First, the accuracy of our model was high, nearing the ability of human. A more sophisticated method can hardly surpass our proposal. Second, the prevention of unintentional touch is a basic and underlying function on touchscreen devices, requiring fast and low-power solutions. Deep learning, as a computationally intensive tool, does not meet the requirement. For these reasons, we argue that classical machine learning methods are more practical for this problem.

6.3 How to improve the detection?

First, we can leverage the keyboard layout as a basis for unintentional touche detection, e.g., when a touch does not fall on any button, it has a greater probability of being an unintentional touch. Second, we can use the language model as priori knowledge. Bayesian decoder is widely used to predict users' desired words from the vocabulary [QwertyRing - 16, 18, 24, 40]. The decoder can also calculate the probability distribution of the next touch. When a touch falls on the button of low probability, it is more likely to be an unintentional touch. In this paper, our proposal leveraged neither the keyboard layout not the language model, because we explored the general method to solve the unintentional touch problem.

6.4 Language dependence of TypeBoard.

Though our studies were conducted in Chinese, we argue that TypeBoard can perform well in different languages. However, to achieve the best performance in other languages, we need to reproduce our study two in the target language, and use (1) the existing feature vector and (2) the new dataset to train the model.

(1) *Why the existing feature vector is adequate?* The Chinese input method was comprehensive. We observed a variety of unintentional touch situations in the study, which help us to design the feature vector thoughtfully. There are a large number of keyboard layouts (e.g., English, German, France and Russian), on which users type straightforwardly to enter characters. The orthography used for Chinese and other East Asian

languages (e.g., Japanese, and Korean) require special input methods. Users narrowed down the range of possibilities by entering the desired character's pronunciation, and then selected the desired ideogram.

(2) *Why we suggest a reproduction of study in the target language?* In study two, we found that the details of user behavior (e.g., the touch pressure and the frequency of a certain behavior) vary in different experimental settings, and these changes had a significant impact on the training results. We believe that this phenomenon exists when we reproduce the study in a different language. So adapting to the target language should slightly improve the performance of TypeBoard.

事实上，键盘布局发生变化的时候，也应该重新做实验二来保证最高准确率。

==========

我们这种凸起的方法相比于物理夹层来说，有哪些优势?

(1) 有可能做到built-in，使得它成为平板电脑的一部分。built-in的方法有可变形的键盘、静电和超声波。(2) 我们的凸起只有0.5mm，用户能感受到该纹理，同时也能把整个屏幕当成触摸板来使用。这样一来，打字和移动光标的操作就能够在同一个空间中进行[TapBoard2]降低了任务切换成本，提高使用效率。相比之下，物理夹层的厚度就大得多，整个屏幕会变成一个巧克力键盘，不能当触摸板来使用。

7 LIMITATION AND FUTURE WORK

这份工作存在着一些limitation，可以指导我们的未来工作。

实验二中，每个用户使用TypeBoard的时长只有20分钟，也就是说，我们采集了用户在初次使用防误触键盘时的用户行为，但没有采集到长期影响下的用户行为。在未来，一个长期的实验是有必要和有价值的。

在真实使用情况下，由于键盘的布局会在触屏的下方，所以小鱼际的误触可能会显著更少。好在我们的算法能处理的问题是包含小鱼际，小鱼际误触少的情况当然也是能够解决的。

有可能的follow up works:

键盘、触摸板一体for个人笔记本电脑?

利用手掌位置来处理触屏输入法中的漂移问题。

加入打字时的触觉反馈，在用户敲下了有意点击的时候，触摸屏可以用震动来模拟物理按键中下压的感觉（类似MacBook Trackpad）。触觉反馈和声音反馈都能提升打字准确率和速度[29]。

用户表示，需要触摸在touchscreen上，加大力度按下去的那种方式，能提高速度和体验。

8 CONCLUSION

一些结论。

REFERENCES

[1] [n.d.]. The Sensel Morph. https://morph.sensel.com/

[2] [n.d.]. Technology. http://www.tactustechnology.com/

[3] Michel Amberg, Frédéric Giraud, Betty Semail, Paolo Olivo, Géry Casiez, and Nicolas Roussel. 2011. STIMTAC: a tactile input device with programmable friction. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*. 7–8.

[4] Michelle Annett, Fraser Anderson, Walter F Bischof, and Anoop Gupta. 2014. The pen is mightier: understanding stylus behaviour while inking on tablets. In *Proceedings of Graphics Interface 2014*. 193–200.

[5] Michelle Annett, Anoop Gupta, and Walter F Bischof. 2014. Exploring and understanding unintended touch during direct pen interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 5 (2014), 1–39.

[6] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. TeslaTouch: electrovibration for touch surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 283–292.

[7] Matthew N Bonner, Jeremy T Brudvik, Gregory D Abowd, and W Keith Edwards. 2010. No-look notes: Accessible eyes-free multi-touch text entry. In *International Conference on Pervasive Computing*. Springer, 409–426.

[8] Jörg Brakensiek and Raja Bose. 2013. Method and apparatus for precluding operations associated with accidental touch inputs. US Patent App. 13/221,344.

[9] B Chaparro, B Nguyen, M Phan, A Smith, and J Teves. 2010. Keyboard performance: iPad versus Netbook. *Usability News* 12, 2 (2010), 1–9.

[10] Matthew JC Crump and Gordon D Logan. 2010. Warning: This keyboard will deconstruct—The role of the keyboard in skilled typewriting. *Psychonomic bulletin & review* 17, 3 (2010), 394–399.

[11] Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. 2018. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[12] Olumuyiwa M Durojaiye and David Abzarian. 2016. Classification of touch input as being unintended or intended. US Patent 9,430,085.

[13] Wolfgang Fallot-Burghardt, Morten Fjeld, C Speirs, S Ziegenspeck, Helmut Krueger, and Thomas Läubli. 2006. Touch&Type: a novel pointing device for notebook computers. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*. 465–468.

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

[15] Andrew M Gordon and John F Soechting. 1995. Use of tactile afferent information in sequential finger movements. *Experimental brain research* 107, 2 (1995), 281–292.

[16] Gregg S Goyins and Mark F Resman. 2001. Palm pressure rejection method and apparatus for touchscreens. US Patent 6,246,395.

[17] Jason Tyler Griffin. 2013. Touch screen palm input rejection. US Patent App. 13/469,354.

[18] Terho Kaikuranta. 2006. Method for preventing unintended touch pad input due to accidental touching. US Patent 6,985,137.

[19] Jeong Ho Kim, Lovenoor Aulck, Michael C Bartha, Christy A Harper, and Peter W Johnson. 2014. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied ergonomics* 45, 6 (2014), 1406–1413.

[20] Sunjun Kim and Geehyuk Lee. 2016. Tapboard 2: Simple and effective touchpad-like interaction on a multi-touch surface keyboard. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5163–5168.

[21] Sunjun Kim, Jeongmin Son, Geehyuk Lee, Hwan Kim, and Woohun Lee. 2013. TapBoard: making a touch screen keyboard more touchable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 553–562.

[22] Seiya Koura, Shunsuke Suo, Asako Kimura, Fumihisa Shibata, and Hideyuki Tamura. 2012. Amazing forearm as an innovative interaction device and data storage on tabletop display. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*. 383–386.

[23] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm as an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[24] Huy Viet Le, Sven Mayer, Benedict Steuerlein, and Niels Henze. 2019. Investigating Unintended Inputs for One-Handed Touch Interaction Beyond the Touchscreen. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–14.

[25] Vincent Levesque, Louise Oram, Karon MacLean, Andy Cockburn, Nicholas D Marchuk, Dan Johnson, J Edward Colgate, and Michael A Peshkin. 2011. Enhancing physicality in touch interaction with programmable friction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2481–2490.

[26] Gordon D Logan and Matthew JC Crump. 2011. Hierarchical control of cognitive processes: The case for skilled typewriting. In *Psychology of learning and motivation*. Vol. 54. Elsevier, 1–27.

[27] Hao Lu and Yang Li. 2015. Gesture on: Enabling always-on touch gestures for fast mobile access from the device standby mode. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3355–3364.

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

[29] Zhaoyuan Ma, Darren Edge, Leah Findlater, and Hong Z Tan. 2015. Haptic keyclick feedback improves typing speed and reduces typing errors on a flat keyboard. In *2015 IEEE World Haptics Conference (WHC)*. IEEE, 220–227.

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

[31] Juha Matero and Ashley Colley. 2012. Identifying unintentional touches on handheld touch screen devices. In *Proceedings of the Designing Interactive Systems Conference*. 506–509.

[32] Fabrice Matulic, Daniel Vogel, and Raimund Dachselt. 2017. Hand contact shape recognition for posture-based tabletop widgets and interaction. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 3–11.

[33] Valentin Popescu. 2015. Touch Screen with Unintended Input Prevention. US Patent App. 14/764,742.

[34] Luca Rigazio, David Kryze, Philippe Morin, and YUN Tiffany. 2013. System and method for differentiating between intended and unintended user input on a touchpad. US Patent 8,502,787.

[35] Julia Schwarz, Robert Xiao, Jennifer Mankoff, Scott E Hudson, and Chris Harrison. 2014. Probabilistic palm rejection using spatiotemporal touch features and iterative classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009–2012.

[36] Andrew Sears. 1991. Improving touchscreen keyboards: design issues and a comparison with other devices. *Interacting with computers* 3, 3 (1991), 253–269.

[37] R William Soukoreff and I Scott MacKenzie. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 113–120.

[38] Hussain Tinwala and I Scott MacKenzie. 2010. Eyes-free text entry with error correction on touchscreen mobile devices. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. 511–520.

[39] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. 2014. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 14–27.

[40] Xuhai Xu, Chun Yu, Yuntao Wang, and Yuanchun Shi. 2020. Recognizing Unintentional Touch on Interactive Tabletop. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–24.

[41] Teppei Yajima and Hiroshi Hosobe. 2018. A Japanese Software Keyboard for Tablets that Reduces User Fatigue. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, 339–346.

[42] Xin Yi, Chun Yu, Weinan Shi, Xiaojun Bi, and Yuanchun Shi. 2017. Word clarity as a metric in sampling keyboard test sets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4216–4228.

[43] Yang Zhang, Michel Pahud, Christian Holz, Haijun Xia, Gierad Laput, Michael McGuffin, Xiao Tu, Andrew Mittereder, Fei Su, William Buxton, et al. 2019. Sensing posture-aware pen+ touch interaction on tablets. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.