Facilitating Text Entry on Smartphones with QWERTY Keyboard for Users with Parkinson's Disease

Yuntao Wang
Department of Computer Science and Technology, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua University, Beijing, China, 100084
yuntaowang@tsinghua.edu.cn

Ao Yu
Global Innovation Exchange (GIX), Tsinghua University and University of Washington, 98005, WA, USA
ya18@tsinghua.org.cn

Xin Yi*
Department of Computer Science and Technology, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua University, Beijing, China, 100084
hhdxs2@163.com

Yuanwei Zhang
Global Innovation Exchange (GIX), Tsinghua University and University of Washington, 98005, WA, USA
mekerzhang@gmail.com

Ishan Chatterjee
Paul G. Allen School of Computer Science and Engineering, University of Washington, 98195, Seattle, USA
ichat@cs.washington.edu

Shwetak Patel
Paul G. Allen School of Computer Science and Engineering, University of Washington, 98195, Seattle, USA
shwetak@cs.washington.edu

Yuanchun Shi
Department of Computer Science and Technology, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua University, Beijing, China, 100084
shiyc@tsinghua.edu.cn

ABSTRACT

QWERTY is the primary smartphone text input keyboard configuration. However, insertion and substitution errors caused by hand tremors, often experienced by users with Parkinson's disease, can severely affect typing efficiency and user experience. In this paper, we investigated Parkinson's users' typing behavior on smartphones. In particular, we identified and compared the typing characteristics generated by users with and without Parkinson's symptoms. We then proposed an elastic probabilistic model for input prediction. By incorporating both spatial and temporal features, this model generalized the classical statistical decoding algorithm to correct insertion, substitution and omission errors, while maintaining direct physical interpretation. User study results confirmed that the proposed algorithm outperformed baseline techniques: users reached 22.8 WPM typing speed with a significantly lower error rate and higher user-perceived performance and preference. We concluded that our method could effectively improve the text entry experience on smartphones for users with Parkinson's disease.

*This denotes the corresponding author.

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445352

CCS CONCEPTS

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

KEYWORDS

Parkinson's disease, text entry, QWERTY keyboard, touch model, statistical decoding

ACM Reference Format:
Yuntao Wang, Ao Yu, Xin Yi, Yuanwei Zhang, Ishan Chatterjee, Shwetak Patel, and Yuanchun Shi. 2021. Facilitating Text Entry on Smartphones with QWERTY Keyboard for Users with Parkinson's Disease. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411764.3445352

1 INTRODUCTION

As the world population's average age increases, Parkinson's disease has become a challenge for more and more people. In 2020, the number of Parkinson's patients reached 10 million [1]. Parkinson's disease is a long-term nervous system disorder that mainly affects the motor system. The most common symptoms are the "pill-rolling" hand tremor (between 4 – 6 hertz) and muscle rigidity/stiffness [42]. As a result, Parkinson's patients usually find fine motor movements (e.g., grabbing spoons and pressing buttons) difficult. Interacting with touchscreen devices is a major challenge for users with Parkinson's disease, specifically, inaccurate input and accidental touches significantly limit their interaction performance

[1]https://www.parkinson.org/understanding-parkinsons

and experience [13, 20–24]. For example, when typing on smartphone QWERTY keyboards, inexperienced users with the hand tremor could only type 4.7 words per minute (WPM) [23], about 11% of the speed of young adults [2].

To improve the text entry performance of users with these symptoms, researchers have proposed several techniques (e.g., keyboard layout optimization [11, 27–30], dynamic accessibility configuration [34], tutoring system on a 9-key keyboard [8, 9] and stroke-gesture text input methods [17, 36, 43]). Although these solutions have been proven to be useful for the motor impaired on various platforms (e.g., tablet, mobile phone), they either targeted beginners or required interface layout modification, causing users to face a potentially steep learning curve [47]. This paper instead focuses on the touch-based software QWERTY keyboard among experienced Parkinson's users, the most dominant text entry method on smartphones, according to our user survey.

Since Goodman et al. introduced the language model to software keyboards [7], the statistical decoding method has been widely deployed and proven effective in various text entry scenarios [31, 32, 44, 48]. The statistical decoding method maps touchpoints' 2D coordinates, also known as the touch model, to the word they most likely represent, known as the language model. However, as we will show in this paper, users with Parkinson's disease suffer from significantly higher insertion and omission errors that can not be handled by classical statistical decoding methods [4, 6]. To correct these types of errors, researchers have proposed pattern matching [12] or machine learning based approaches [37, 40] with their penalties tuned manually or trained by machine. These methods lack physical interpretability, which is difficult to generalize to other scenarios like ours. To the best of our knowledge, there is no existing work that explored an effective statistical decoding method for software QWERTY keyboard to support users with Parkinson's disease.

This paper presents and evaluates a smartphone QWERTY keyboard for users with Parkinson's disease using an elastic probabilistic model. We first conducted a user survey to explore how users input text in a daily scenario, including the most widely used keyboard layout and typing posture. Then we investigated and compared the typing behaviors generated by both Parkinson's and non-Parkinson's users. Finally, we proposed an elastic probabilistic model to correct all major types of errors while maintaining direct physical interpretation by incorporating spatial-temporal features. In a second user study, we evaluated the performance versus two baseline models: the basic language model (BLM) [7] and elastic pattern matching (EM) [12]. Results showed that our method achieves significantly higher typing speed (22.8 WPM), 26.8% and 14.6% faster than BLM and EM respectively; as well as lower word-level error rate (8.0%), 7.1% (5.5%) lower than BLM (EM) and keystrokes per character (1.06), 7.8% (5.5%) lower than BLM (EM). Finally, users ranked our method to be the best in terms of perceived accuracy, speed, error correction performance, confidence, and overall preference. This paper's contributions include:

1. We built and analyzed the touch model of experienced Parkinson's users on the software QWERTY keyboard, including both temporal and spatial features.

2. We propose a smartphone QWERTY keyboard for users with Parkinson's disease based on the elastic probabilistic model.

3. We proved the high text entry performance and usability of our method against baselines through an in-lab, controlled user evaluation study.

2 RELATED WORK

In this section, we describe related work including studies describing the effect of Parkinson's symptoms, text entry methods for users with Parkinson's disease, classical statistical decoding methods, and related error auto-correction methods.

2.1 Effect of the Parkinson's Symptoms

Parkinson's disease causes multiple motor impairments such as hand tremors, muscle rigidity/stiffness etc. [42] However, these symptoms can also occur in other elderly people [13, 21] resulting in difficulty interacting with touchscreens [21, 23]. To understand the effects of these symptoms, researchers investigated users' interaction behavior with touchscreen phones [13, 20–23]. They found that symptoms such as hand tremors and muscle stiffness have a significantly negative effect on touchscreen interaction tasks, including target selection (especially small targets) [13, 20, 21], text entry [21, 22], and object manipulation (e.g., zoom in/out) [22].

Recently, Nunes et al. [24] investigated Parkinson's users' tap, swipe, multiple-tap, and drag gesture performance through a user study composed of 39 Parkinson's participants. They observed relatively poor small keys (79.83% for 7.0*mm* key width) tap accuracy. Further, Nicolau and Jorge analyzed the text entry performance of elderly citizens with different severity hand tremors [23]. They found relatively high error rates compared with young adults due to the hand tremor. They suggested future work to use temporal and spatial features to increase text entry performance. However, Nicolau and Jorge studied users with no experience using smartphone touchscreens. As touchscreens become more ubiquitous, we have observed elder people interacting with them much more frequently [36], which leads to different usage habits and touch behavior. This motivates our work to enhance text entry performance for experienced users with Parkinson's disease.

2.2 Text Entry Methods for Parkinson's Users

Text entry is one of the most challenging tasks on smartphones among elderly people with Parkinson's symptoms [22, 23, 26], significantly harming their user experience. When typing on smartphone QWERTY keyboards, inexperienced Parkinson's users could only input 4.73 words per minute (WPM) [23], about 11.7% of the speed of non-Parkinson users [17]. Therefore, researchers have explored solutions to solve this problem [3, 9, 9, 27–30, 34]. Rodrigues et al. found that highlighting the next most likely letter for a predicted target word or providing candidate words did not improve the text entry performance [27, 28] for the elderly without touchscreen experience. Researchers also explored the text entry tutoring system [8, 9], which automatically detects input stumbles and provides instructions that help users resolve them independently. They showed that an assistive typing application increased typing speed by 17.2% and reduced input stumble incidence by 59.1%. Shari Trewin described a keyboard that dynamically self-adjusts the configuration features, including key repeat delay, key repeat rate, and debouncing time for accessibility [34]. Specifically targeted at hand tremors, Sarcar et al. found that users with finger tremor and dyslexia can achieve 4.68 WPM (14.20% error rate) on a standard QWERTY keyboard and 5.35 WPM (10.61%) on their optimized T9 keyboard [29, 30]. Jabeen et.al [11] proposed a new keyboard layout for Parkinson's users to input Chinese characters, achieving an average input speed of 3.88 WPM. Other researchers explored stroke-gesture text input on touchscreens [17, 36, 43], considering that sliding helps reduce finger oscillation [17, 38]. For example, EdgeWrite used an assistive piece of hardware to guide stroke input [43] and users achieved 6.6 WPM text input performance.

Although these solutions have been proven useful for motor impaired users on various platforms (e.g., tablet, mobile phone), they either targeted beginners or modified interface layouts with could result in a significant learning barrier [47]. Therefore, in contrast to prior work, we focused our study on the common tap-based software QWERTY keyboard, the dominant text entry method on touchscreen phones among experienced Parkinson's users.

2.3 Classical Statistical Decoding Algorithm

The classical statistical decoding algorithm is a probabilistic model that calculates the likelihood of each word in a pre-defined dictionary according to the user's input and recommends the word with the highest likelihood. This algorithm was first proposed by Goodman et al. [7], which yields:

$$P(W|I) \propto P(I|W) \times P(W) \tag{1}$$

Where I is a series of input points, W is a candidate word, $P(W)$ quantifies the probability of W, and $P(I|W)$ models the noise in users' typing behavior. So far, this statistical decoding algorithm has been proven effective in many smart keyboard techniques (e.g.,[4, 6, 39, 40, 44]).

The principled probabilistic theory, which enables the calculation to be interpreted as a probability, is an advantage of the classical statistical decoding algorithm. Therefore, the algorithm can be easily incorporated with a language model or other input channels (e.g., accelerometer [5, 25]). Further, the calculation of $P(I|W)$ can be simplified to Equation 2 by assuming that touchpoints are independent of each other. As a result, the classical statistical decoding method models spatial touchpoint distribution (offset and spread of size) for each individual key using Bivariate Gaussian distribution [2, 7]. A language model can calculate the a list of probable next characters for each touchpoint. However, the classical statistical decoding algorithm can not correct insertion or omission errors [4, 6], which are common among users with the hand tremor.

$$P(I|W) = \prod_{i=1}^{N} P(I_i|W_i) \tag{2}$$

2.4 Correcting Insertion and Omission Errors

Pattern matching, which calculates the "distance" between the input pattern and the pattern of candidate words then recommends the word with the smallest distance, is one major approach for correcting insertion and omission errors. Pattern matching is the key algorithm for gesture keyboards [14, 46]. By comparison, less work has applied it to typing [12, 15]. The merit of pattern matching is that the calculation is based on the shape of the word patterns, therefore, is not restricted by the number of input points. However, the computed distance metric is usually nontrivial to be interpreted as probability [14]. Therefore, researchers have to use empirically determined parameters to incorporate it with language models.

To our knowledge, only a limited number of studies used machine learning based methods to correct insertion and omission errors [37, 40]. These were achieved by introducing corresponding penalties that were trained using machine learning or tuned manually (as opposed to summarized). Therefore, these methods lack physical interpretability and require initial calibration [37].

3 TEXT ENTRY OF PARKINSON'S USERS IN DAILY LIVES

We conducted a survey to explore how Parkinson's users input text on their smartphones daily. Specifically, we are interested in typing posture and keyboard layout.

We interviewed 16 participants (7 males, 9 females) with Parkinson's disease. Their average age was 67.9 ($s.d.$ = 10.6), the participants, on average, have had the disease for 8.4 years ($s.d.$ = 7.6), and were at stage 2.5 ($s.d.$ = 1.5) of Parkinson disease [2]. We conducted surveys over online video calls so that participants could demonstrate their daily-used typing postures and keyboard layouts. First we asked: "What kind of text input method you use in a daily scenario?" If the answer included a software keyboard we asked following two questions: 1) "Which layout do you utilize when entering text on your smartphone keyboard?" 2) "What typing postures do you use in daily lives? Please rate the frequency of usage (5-point Likert scale, 4 - all the time, 0 - never)." Each participant was given a 5 USD gift card as compensation.

3.1 Results

Among the participants, 13/16 used a QWERTY keyboard for entering text on their smartphones, while 3/16 used voice assistants for text entry. Surprisingly, although T9 keyboard has larger key sizes than QWERTY keyboards, none of the participants chose T9 for text entry. Through the interview, P1, P6, P7, P8, P12, and P13 commented that they can type faster/with more comfort on the QWERTY keyboard; P2, P6, P9, P14, and P15 commented that they have gotten used to the QWERTY keyboard layout after using it for many years. P3, P10, and P16 commented that the QWERTY keyboard layout was the default when they started to use the smartphone.

We studied five different postures that are common among the 13 participants as illustrated in Figure 1: 1) one hand holding the phone in the air while the other hand's index finger types; 2) one hand holding the phone on a stable surface (i.e., table) while the other hand's index finger types; 3) place the phone on a stable surface (i.e., table) while one index finger type; 4) two-thumbs typing in the air; and 5) place the phone on a stable surface (i.e., table) while two index fingers type.

Figure 1 shows the frequency of different typing postures. Noticeably, one index finger typing was preferred by Parkinson's users over two-thumb typing, which is different from non-Parkinson's users [2]. This result aligns with prior work [22, 23, 35] regarding dexterity impaired users' preferred typing posture. Specifically,

[2] https://www.healthline.com/health/parkinsons/stages

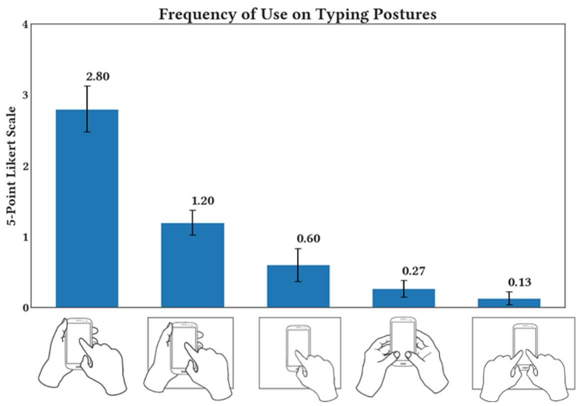

Figure 1: Frequency of different typing postures. Error bar indicates one standard error.

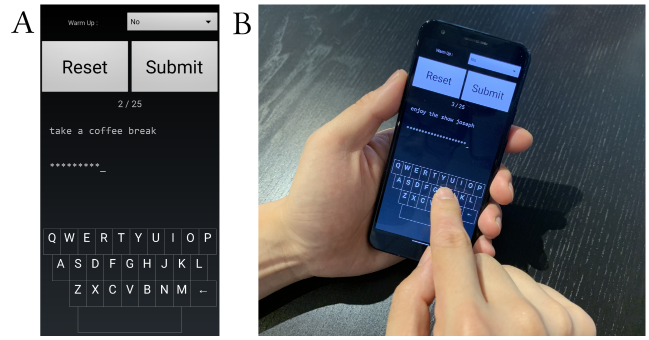

Figure 2: (A) Experiment platform, shows the target string and the current entered text in asterisk. (B) Typing posture adopted by all the participants.

12/13 participants chose posture (1) as the most frequently used typing posture. During the interview, P1, P3, P7, P8, P10, P12, P13 and P16 mentioned about reasons being the hand tremor and the stiffness issues. Therefore they preferred using the other hand to hold the phone steady. P2 and P12 both commented that *"I feel uncomfortable and stiff typing with two hands "*; Moreover, postures (2) and (3) all used a surface (i.e., table) to provide stable support for the phone, which *"was very helpful to relieve the shaking problem from the hand tremor"* (P1-P2, P7-P10, P12) *"helpful towards the fatigue after typing for a while"* (P3, P10, P14).

In conclusion, we believe that it is important to optimize the current QWERTY keyboard on smartphones for Parkinson's users since it is still the most dominant text entry method. Parkinson's users prefer a one-index-finger typing posture rather than two-thumb typing [23] due to the hand tremor and stiffness. They prefer to place the phone on a stable surface (i.e., table, leg) to relieve the fatigue and hand tremor problems. Therefore, we deploy on a one-index-finger typing posture in our studies.

4 MODELING THE TYPING BEHAVIOR

We conducted a user study to investigate Parkinson's users' typing behavior on smartphone keyboards and compared it to that of non-Parkinson's young adults. As a result of the previous section's findings, we chose the QWERTY keyboard layout in this study. Also, to ensure that we observed the most intrinsic user typing pattern without bias towards any specific input prediction algorithm, we used asterisk feedback (Figure 2) as with other works [2, 44]. We were interested in not only users' typing speed and touchpoint distribution, but also their error patterns (e.g., frequency and spatial/temporal features).

4.1 Participants

We recruited 8 Parkinson's users (3 females, 5 males, all right-handed) with a mean age of 60.5 ($s.d. = 9.2$, distributed between 47 and 72) from a local Parkinson's foundation. 7/8 participants were diagnosed with moderate or severe tremor symptoms in both hands. The remaining participant had a slight hand tremor symptom in his left hand. We also recruited 8 non-Parkinson young adults (5 females, 3 males, all right-handed) with a mean age of 23.6 ($s.d. = 3.7$) as our control group. All 16 participants utilized a QWERTY keyboard for smartphone text entry in their daily lives. Each participant was compensated with 20 USD.

4.2 Apparatus and Platform

We used a Google Pixel 3A phone ($PPI = 441$) in this study, with each pixel measuring 0.057mm. Figure 2 showed the experiment platform. Similar to commercial keyboards, we rendered each key on the keyboard to be $6mm(W) \times 9mm(H)$. During typing, the platform showed asterisk feedback upon each touch.

4.3 Experiment Design and Procedure

We used a between-subject design in this study, with *Hand Tremor (with vs. without)* as the only factor. Upon arrival, each participant provided his or her age, Parkinson's disease history, and hand tremor severity. Then they spent several minutes familiarizing themselves with the experiment platform. Each participant completed two blocks of text entry tasks during the study, each consisting of 25 phrases randomly sampled from the Mackenzie and Soukoreff phrase set [19]. A 5-minute break was enforced between the two blocks. We first confirmed with all the Parkinson's users that posture (1) and (2) (see Figure 1) were the most frequently used postures. However, to avoid muscle fatigue during this study, we asked

Table 1: Error rate of different typing error categories.

	Insertion		substitution		Omission		Transposition		Overall	
Parkinson's	yes	no	yes	no	yes	no	yes	no	yes	no
Mean Error Rate	6.63%	0.25%	12.38%	3.47%	1.13%	0.17%	0.10%	0.07%	20.24%	3.96%
SD	1.24%	0.08%	5.12%	1.05%	0.33%	0.08%	0.04%	0.04%	5.66%	1.17%

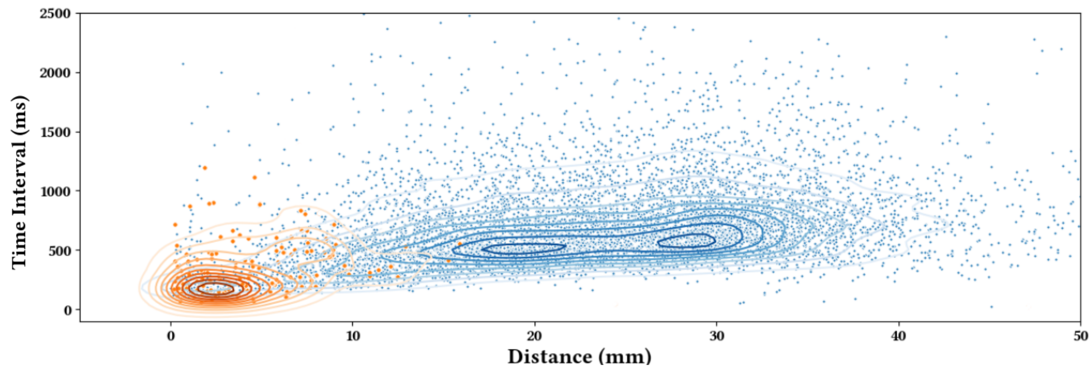

Figure 3: the density distribution of unintentional repetitive touches (orange) and regular intentional touches (blue).

all the 16 participants to adopt posture (2) with a table where they can rest their arms. When entering each phrase, the participants were asked to "type as quickly and accurately as possible" [23], and not to correct any errors.

4.4 Results

We collected 9,441 and 9,112 touch points from Parkinson's users and non-Parkinson's users, respectively. As hand tremors may cause various kinds of typing errors (e.g., insertion and omission), we manually labeled all the collected touch points with the target characters. Firstly, we mapped touchpoints to characters using the nearest distance to each key's centroid. Then, three annotators discussed and labeled the input string by comparing it to the target string, with the help of the video recording. We discarded data points that were not successfully labeled totalling 197 (2.1%) and 134 (1.4%) for Parkinson's and non-Parkinson's users respectively. All metrics are normally distributed (p > .05) based on the Shapiro-Wilk test. We used independent samples t-test (with Cohen's d) for statistical tests and reported a significant difference with $p < .05$.

4.4.1 Text Entry Speed. We measured the text entry speed in *Words Per Minute (WPM)*, which was calculated using the formula in [18]:

$$\text{WPM} = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5} \tag{3}$$

where $|T|$ denotes the length of target phrase, S denotes the time interval in seconds between the first and the last touch when entering the phrase.

Our results show that the average speed of Parkinson's and non-Parkinson's users were 19.8 WPM ($s.d. = 6.9$) and 29.4 WPM ($s.d. = 8.9$) respectively. As expected, Parkinson's users reached significantly lower text entry speed than non-Parkinson's users ($t_{14} = -4.64, d = 2.3, p < .001$). Our results differ from prior work [2, 17] evaluating the typing performance on both elder (22.8 WPM) and young groups (36.34 WPM). This is because we didn't take the SPACE character into consideration. When counting the SPACE character, we observed similar typing speed with prior work. However, inexperienced users with a hand tremor can only achieve 4.73 WPM [23], which indicates the necessity for this user study modeling experienced Parkinson's user's typing behavior.

4.4.2 Typing Errors. Table 1 showed the error rate of different types of typing errors. Overall, the error rate of participants with and without Parkinson's was 20.2% and 4.0%, respectively.

We analyzed the error rate of different types of typing errors. We found similar results as prior work [23] including that substitution and omission are the two most common typing errors in a mobile device text entry. Further, compared with non-Parkinson users, Parkinson users yielded significant more insertion ($t_{14} = 14.5, d = 7.3, p < .001$), substitution ($t_{14} = 4.8, d = 2.4, p < .001$), and omission errors ($t_{14} = 8.0, d = 4.0, p < .001$), but not transposition errors ($t_{14} = 1.5, p = .16$).

When comparing our results with prior work [23], we observed smaller overall error rate (20.24% v.s. 25.97%). Specifically, our results yielded a higher substitution error rate (12.38% v.s. 7.80%) and a higher insertion error rate (6.63% v.s. 5.50%), but a lower omission error rate (1.13% v.s. 12.65%). We believe the reasons for this discrepancy are: 1) experience in using the QWERTY keyboard; 2) cognitive issues; 3) keyboard size. Hugo Nicolau and Joaquim Jorge studied elder users with hand tremors who had no experience using the QWERTY keyboard on smartphones. Further, they concluded that forgetfulness and coordination issues caused the high omission error rate they observed. Finally, Nicolau and Jorge studied mobile keyboards in the landscape mode because they are larger than portrait mode. However, in our study, all participants use the QWERTY keyboard layout on their smartphones during daily life without reported cognitive issues. Further, participants in our study were familiar with the keyboard in portrait mode notwithstanding smaller key size.

4.4.3 Unintentional Repetitive Touch. Our results indicate that unintentional repetitive touches caused most of the insertion errors observed (480/613 = 78.3%). We noticed that unintentional repetitive touches have a relatively shorter time interval and shift from the last touch coordinate, whereas regular touches exhibited more diverse spatial-temporal correlation. To model this, we adopted Gaussian kernel density estimation (KDE) to estimate the probability density function — $P_H(\mathbf{x})$ of *time interval* ($x_1 = T(I_i, I_{i-1})$) × *distance* ($x_2 = D(I_i, I_{i-1})$) of two adjacent touch points — I_i and I_{i-1} respectively, as Equation 4 shows:

$$P_H(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_H(\mathbf{x} - \mathbf{x_i}) \tag{4}$$

Figure 3 shows the density distributions of unintentional repetitive touches in orange contours and regular intentional touches in blue contours. The centroid of unintentional repetitive touches density distribution is 2.4 mm and 185 ms. Compared with users without a hand tremor [10], hand tremors caused a longer distance between adjacent touches. Further, touch time interval aligns with the frequency (4 - 6 Hz) of the hand tremor in Parkinson's disease. Hugo Nicolau and Joaquim Jorge found similar results on the time interval [23].

Our work proves the feasibility of leveraging the spatial-temporal features to classify unintentional repetitive touches from regular touches. Results indicate that there are distinguishable distributions among these two types of touches, implying that we can filter out unintentional repetitive touches. We utilized second-order Gaussian kernel as our kernel function — K. The KDE bandwidth — H are counted from the unintentional repetitive touchpoints and regular touchpoints as 0.24 and 0.40.

4.4.4 Touch Point Distribution. Figure 4A and B illustrate the collective touchpoints merged from all Parkinson's and non-Parkinson's participants over a 1:1 sized keyboard. Similar to prior work [2, 41], even for Parkinson users, the endpoints for each key roughly followed a 2-dimensional Gaussian distribution. We report the results for the space key separately because it is different from the alphabetical keys in both form and function.

Systematic Offset. We define systematic offset as the distance between the touch point cloud centroid and the target key center. A positive offset in x and y dimension indicated that the users hit to the right and bottom of the target key center, respectively. Figure 4C and D show the average offsets among the alphabetic keys for Parkinson's and non-Parkinson's users.

Users tended to touch to the lower-right of the key center. The average Offset_x of Parkinson's users and the non-Parkinson's users was 1.19 mm ($s.d. = 0.59$) and 0.57 mm ($s.d. = 0.33$) respectively, and Offset_y was 0.82 mm ($s.d. = 0.23$) and 0.43 mm ($s.d. = 0.22$) respectively. Parkinson users yielded significantly greater Offset_x ($t_{50} = 4.7, d = 1.3, p < .001$) and Offset_y ($t_{50} = 6.2, d = 1.7, p < .001$) than non-Parkinson's users, confirming that they had more difficulty performing fine motor control tasks.

For Parkinson users, Offset_x tended to be smaller for keys on the right side of the keyboard. The average Offset_x of left-end keys ('Q', 'A' and 'Z') and right-end keys ('P', 'L' and 'M') were 1.55 mm ($s.d. = 0.36$) and 0.67 mm ($s.d. = 0.44$), respectively. A significant effect of side on Offset_x was found ($t_{24} = 5.6, d = 2.2, p < .001$). However, we did not find such effect on Offset_y ($t_{24} = 0.4, p = .71$). For non-Parkinson users, there was no significant effect of side on either Offset_x or Offset_y.

Spread of Touch Point. To measure spread size, we calculated SD_x and SD_y, the standard deviations of the touch point locations in the x and y directions. The average SD_x and SD_y across all keys were 1.17 mm ($s.d. = 0.13$) and 1.14 mm ($s.d. = 0.10$) for Parkinson's users, and were 1.00 mm ($s.d. = 0.13$) and 0.97 mm ($s.d. = 0.11$) for non-Parkinson's users. Parkinson's users yielded a significantly wider spread on touch points on both x ($t_{50} = 4.7, d = 1.3, p < 0.001$) and y ($t_{50} = 5.8, d = 1.6, p < 0.001$) directions (see Figure 4).

Space Key. Similar to other keys, users tended to touch to the lower-right of the space key center. The average Offset_x of Parkinson's users and the non-Parkinson's users was 2.42 mm and 1.63

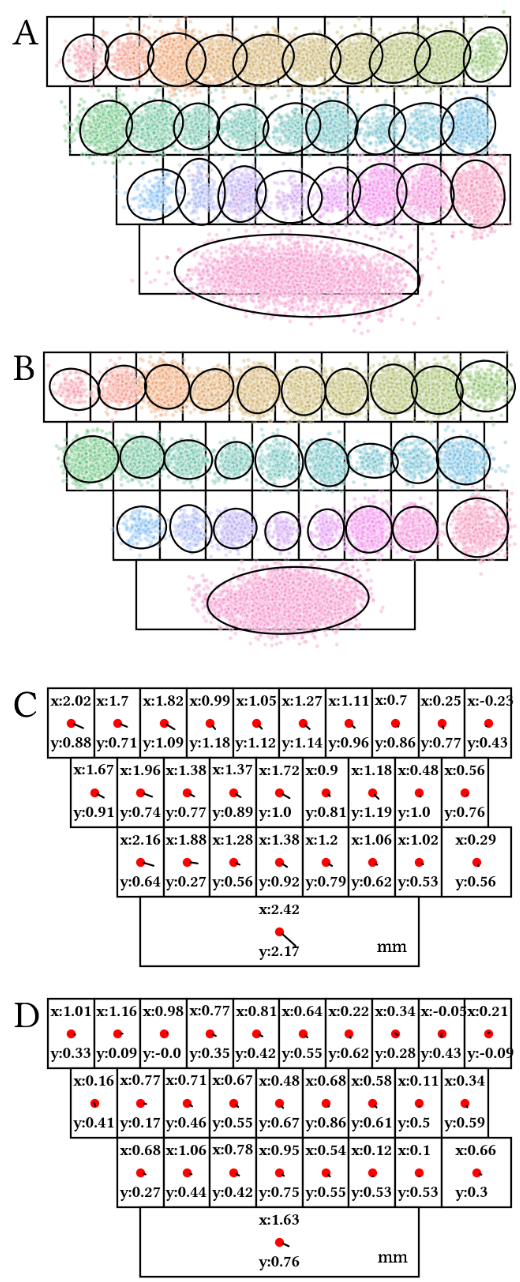

Figure 4: (A) Touch point distribution across all Parkinson's users. (B) Touch point distribution across all non-Parkinson's users. (C) Systematic offset of each key across all Parkinson's users. (D) Systematic offset of each key across all non-Parkinson's users. We show 95% confidence ellipses over a 1:1 outline on each key.

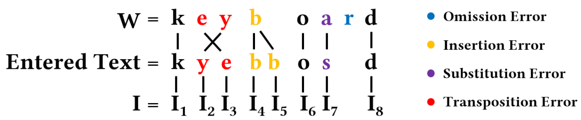

Figure 5: One example demonstrating different error types highlighted in different colors.

mm respectively, and Offset$_y$ was 2.17 mm and 0.76 mm respectively. Parkinson users have greater SD$_x$ and SD$_y$ than non-Parkinson's users on the space key. The average SD$_x$ of Parkinson's users and the non-Parkinson's users was 15.87 mm and 10.44 mm respectively, and SD$_y$ was 5.36 mm and 4.35 mm respectively.

5 ELASTIC PROBABILISTIC MODEL

In this section, we propose an elastic probabilistic model to predict users' input when various types of typing errors exist at the same time. As mentioned in Related Works section, the classical statistical decoding algorithm can only compute the probability of different words whose length was identical to the input sequence (e.g. [7, 44]), which limited the usability of such techniques, especially when the frequency of these types of errors were relatively high (e.g. for Parkinson's users). And although some algorithms can handle insertion/omission errors by introducing penalties (e.g., VelociTap [37]), the value of the parameters was manually optimized, which was not designed to have a direct physical interpretation. Our goal is to generalize the classical statistical model based on probabilistic theory, which will introduce two advantages: 1) All the parameters have direct physical interpretation, which can be easily calculated (as opposed to be trained) based on collected user data; 2) The calculated result can be interpreted as "probability", making it possible to incorporated with other signal data (e.g. accelerator data [5]).

5.1 Problem Definition

We first revisit input prediction problem: Given a series of touch input points $I = I_1 I_2 ... I_n$, for each candidate word $W = W_1 W_2 ... W_m$ in a predefined language model, calculate the probability $P(W|I)$. According to Bayesian rule, the key is to calculate $P(I|W)$ (see Equation 1). Notice that m and n can be unequal when assuming insertion/omission errors may occur. Therefore the major challenge is to deduce the equation of calculation for each specific category of typing errors.

Based on different mappings between I_i and W_j, typing errors can be categorized into four kinds, as illustrated in Figure 5:

(1) **Insertion Error**: a redundant touch I_i that does not map to any of the target characters.
(2) **Omission Error**: a character W_j that does not map to any of the input touch points.
(3) **Substitution Error**: a touch I_i for character W_j, but landed on other keys.
(4) **Transposition Error**: swap touches I_{i-1} and I_i for characters W_{j-1} and W_j.

5.2 Algorithm Deduction

To calculate the probability $P(I|W)$, we first assign P_i, P_o, P_s and P_t to the probability of insertion, omission, substitution and transposition errors, respectively. In practice, these values can all be easily calculated based on the user's typing data. Assuming that the mapping between I and W is known (i.e., we can determinately label all kinds of typing errors), the calculation of $P(I|W)$ is intuitive:

We define $F_{i,j}$ as the conditional probability for $I = I_1 I_2 ... I_i$ and $W = W_1 W_2 ... W_j$, where $1 \le i \le n$ and $1 \le j \le m$, then we can calculate the value of $F_{i,j}$ as:

$$F_{i,j} = \begin{cases} F_{i-1,j} \times P_i(I_i|I_{i-1}) \\ F_{i,j-1} \times P_o \\ F_{i-1,j-1} \times (1 - P_o) \times P_s(I_i|W_j) \times P_{con} \\ F_{i-2,j-2} \times (1 - P_o)^2 \times P_{trans} \end{cases} \tag{5}$$

where

$$P_{con} = \begin{cases} 1 - P_t & \text{if } I_{i-1} \text{ matches } W_{j-1} \\ 1 & \text{else} \end{cases} \tag{6}$$

and

$$P_{trans} = \begin{cases} P_t & \text{if } I_{i-1} I_i = W_j W_{j-1} \\ 0 & \text{else} \end{cases} \tag{7}$$

Given that $F_{0,0} = 1$, we can iteratively use Equation 5 to calculate $P(I|W) = F_{n,m}$. It is easy to find that the calculation of $F_{i,j}$ generalized the classical statistical decoding algorithm [7] based on probabilistic theory, therefore the result has direct physical interpretation. Specifically, if we assume that $P_i = P_o = P_t = 0$ (i.e., there are no insertion/omission/transposition errors), Equation 5 would degenerate to Equation 2.

So far, we have assumed that the mapping between I and W is known. However, in practice, the mapping should be inferred by the algorithm. To achieve this, we used dynamic programming similar to Levenshtein distance [16]: upon each touch input, we update $F_{i,j}$ as the highest value of the four possibilities. To accelerate the real-time computation speed, we can build a tree based on the corpus and use incremental computing and pruning techniques (e.g., beam pruning) to avoid repetitive searches and prune impossible results.

5.3 Dynamic Parameter Adjustment

Equation 5 generalized the classical statistical decoding algorithm to correct insertion, omission and transposition errors. However, a limitation of this model is that the probability of different types of errors are fixed, which ignores some useful features exhibited in user's typing behaviors. For example, in Figure 3, unintentional repetitive touches and regular touches are highly distinguishable based on spatial-temporal features. To achieve this, we further improved EPM by dynamically calculating P_i based on the input as $\widetilde{P}_u(\boldsymbol{x}) + P_i^{'}$. $P_i^{'}$ represents a fixed insertion error rate (133/9244 = 1.44%) that excludes all unintentional repetitive touches. $\widetilde{P}_u(\boldsymbol{x})$ is calculated as:

$$\widetilde{P}_u(\mathbf{x}) = \frac{\omega P_u(\mathbf{x})}{P_u(\mathbf{x}) + P_r(\mathbf{x})} \tag{8}$$

Where $P_u(\mathbf{x})$ and $P_r(\mathbf{x})$ represent the probability density function of unintentional repetitive touches and regular touches (Equation

4). $x = [T(I_i, I_{i-1}), D(I_i, I_{i-1})]$. ω denotes as the weight (5.19%, 480/9244) counted from the unintentional repetitive touches.

6 USER EVALUATION STUDY

In this section, we describe a user study designed to compare and evaluate the performances of the proposed auto-correction techniques. Then we report the major results and findings of our method versus other baseline auto-correction methods. We used RM-ANOVA ($p < .05$) with the Tukey post-hoc analysis and (partial) eta squared as the effect size for parametric analysis. We utilized Friedman test ($p < .05$) and Wilcoxon signed-rank test ($p < .05$) for non-parametric analysis. We compared the typing speed, character error rate, word-level error rate, keystrokes per character, and user feedback. All metrics fall into a normal distribution (p > .05) using the Shapiro-Wilk test.

6.1 Participants

We recruited 8 participants with Parkinson's disease (3 females, 5 males, all right-handed) with an average age of 59.9 ($s.d. = 7.0$). All participants were regular users of the QWERTY keyboard on their smartphones. 3 (4) of the 8 participants have severe (moderate) hand tremor symptoms. The remaining participant was treated with deep brain surgery and has slight hand tremor symptoms. 4 participants attended our prior study. Each participant was compensated 30 USD.

6.2 Experiment Design

We used a within-subjects single-factor design, with *Technique* being the only factor. Specifically, we tested four different techniques: EPM with dynamic parameter adjustment feature (D-EPM), EPM with constant error probabilities (C-EPM), basic language model [7] (BLM) and elastic pattern matching (EM) [12, 15, 45]. For language model, we used the top 60,000 words and their corresponding frequencies in the American National Corpus frequency data [1]. In our implementation, we set P_s to 12.38%, P_o to 1.13%, and P_t to 0.1% according to the typing data from section 4 for C-EPM and D-EPM. We set P_i to 6.63% for C-EPM and $P_i^{'}$ to 1.44% for D-EPM (see section 4.4.2).

We chose the Basic Language Model as the first baseline, as it is the most widely adopted algorithm. We adopted the Bayesian decoding algorithm (zero-order) proposed by Goodman et al. [7]. This method assumes that the user manually corrects insertion, omission, and transposition errors. Therefore, it could only correct substitution errors. To establish a more cutting-edge technique for other typing errors, we built another baseline method — elastic pattern matching.

Elastic Pattern Matching was the second baseline. It has been proved effectively correct typing errors caused by the motor impairment [12]. Elastic Pattern Matching calculates the similarity between candidate words and the user touch sequence through pattern matching. To measure the similarity, we utilized a weighted minimum string distance (wMSD) score proposed by [12].

Our keyboard interface allowed Users to tap on the top 5 candidate words to select. Further, they are free to correct the input or not. We took mis-taps into the calculation for all metrics.

6.3 Procedure

First, we introduced participants to the purpose of the user study and text entry application on the Google Pixel 3A smartphone. We only informed them that we want them to experience our keyboard without telling them there is a difference among methods. We then collected information using a background information survey, including participant age, gender, keyboard layout, and typing posture. Participants were instructed to experience four auto-correction techniques during the warm-up session. Finally, they finished two test sessions covering all four auto-correction methods. Each session comprised 25 sentences, which were randomly sampled from the phrase set proposed by Mackenzie and Soukoreff [19]. We then repeated another four test sessions. The order of auto-correction method was counter-balanced among users. We instructed the participants to "type as quickly and accurately as possible". The candidate words and characters in the current word were shown in plain text. We asked all participants to take a 5-minutes break between each session. The experiment lasted around 60 minutes in total. Therefore, we obtained *4 methods* × *2 repeats* × *25 sentences* = *200 sentences* for each participant.

All participants completed a survey directly following each auto-correction method. We did not mention any method difference among sessions. They were free to change their previous ranked scores. After the experiment, participants were asked to revise the final scores of all four surveys. We asked them to rank the following 5 questions in a 7-point Likert scale (1: strongly disagree, 4: neutral, 7: strongly agree): 1. I think I can type accurately using this keyboard; 2. I think I can type fast using this keyboard; 3. This keyboard would correctly predict the word I'm typing; 4. I spend less mental effort to tap each key accurately; 5. I prefer using this keyboard.

6.4 Typing Speed

Figure 6 compares participants' typing speed while using 4 auto-correction methods. Results show that D-EPM achieves the highest typing speed with an average of 22.8 WPM ($s.d. = 10.0$) while BLM, EM and C-EPM achieve 18.0 ($s.d. = 9.5$), 19.9 ($s.d. = 10.4$), and 21.3 ($s.d. = 9.0$) WPM respectively. All participants achieved a higher average typing speed with the elastic probability model than the basic language model. RM-ANOVA analysis indicates that there are significant differences ($F_{3,60} = 4.51, \eta^2 = 0.18, p < .05$) among auto-correction methods. D-EPM achieves significantly higher typing speed than BLM ($p < .001$) and EM ($p < .01$) with an average increase of 26.8% and 14.7%. However, there is no significant difference between D-EPM and C-EPM ($P = .07$) with an average increase of 7.0%.

6.5 Character Error Rate

We measured the character error rate (CER), which is a common metric for evaluating text entry techniques (e.g., [37, 44]). CER can be interpreted as the minimum number of insertions, substitutions, and deletions required to transform the transcribed string into the target string divided by the number of characters in the target string.

Results show that D-EPM achieves the lowest CER — 22.4% ($s.d. = 15.7\%$) while BLM, EM and C-EPM achieve 29.2% ($s.d. =$

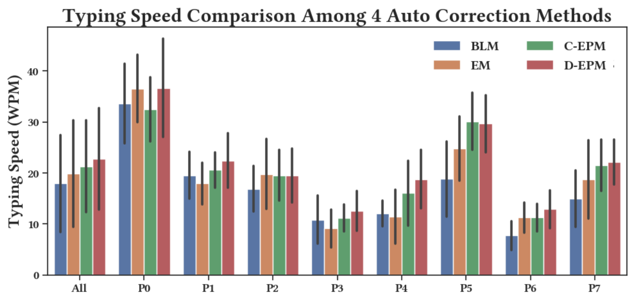

Figure 6: Typing speed comparison among 4 auto-correction techniques for each participant. The error bars represent the standard deviations.

20.5%), 34.5% ($s.d. = 22.9\%$), and 23.0% ($s.d. = 16.2\%$) respectively. There is significant effect ($F_{3,60} = 5.12, \eta^2 = 0.20, p < .01$) of auto correction methods on the CER. Tukey post-hoc analysis shows that both C-EPM and D-EPM yeild significantly CER than BLM and EM ($p < .05$). However, there is no significant difference between the D-EPM and C-EPM methods ($F_{3,60} = 1.22, p = .08$).

6.6 Word-Level Error Rate

We measured word-level error rate calculated as the probability of the incorrect input word that is not in the targeted string. Results show that D-EPM achieves the lowest error rate — 8.0% ($s.d. = 14.1\%$) while BLM, EM and C-EPM achieve 15.1% ($s.d. = 21.5\%$), 13.5% ($s.d. = 21.8\%$), and 9.6% ($s.d. = 17.7\%$) error rates respectively. Auto-correction method significantly affects ($F_{3,60} = 7.34, \eta^2 = 0.27, p < .001$) word-level error rate. Tukey post-hoc analysis shows that both C-EPM and D-EPM yield significantly lower error rate than BLM and EM ($p < .05$). However, there is no significant difference between the D-EPM method and C-EPM method ($F_{3,60} = 1.09, p = .28$).

6.7 KSPC

We used keystroke per character (KSPC) [33] to measure keyboard performance. A lower KSPC indicates that the user expends less effort in modifying the input. KSPC is calculated as the ratio of the number of touches over the length of the targeted word. Results show that D-EPM achieves the best KSPC performance with an average value of 1.06 ($s.d. = 0.17$). There are significant differences ($F_{3,60} = 15.8, \eta^2 = 0.44, p < .001$) among auto-correction methods. Tukey post-hoc analysis indicates that D-EPM achieves significantly lower KSPC than BLM (KSPC = 1.15, $p < .01$) as well as EM (KSPC = 1.23, $p < .01$), suggesting that our method is more effective in terms of correcting typing errors. Therefore, Parkinson users perform fewer keystrokes for the same task, resulting in a faster typing speed.

6.8 Top-K Accuracy

Top-K accuracy is an important metric to measure the keyboard performance since auto-correction algorithms provide a list of candidate words from the user's touch inputs. Figure 8 shows the top-1 to top-5 error rates of four auto-correction methods. We report the results of four methods, BLM, EM, C-EPM, and D-EPM. Results show that all methods have high top-1 error rates, 14.05%, 12.86%, 13.81% and 15.94% respectively, without significant difference ($F_{3,60} = 1.1, p = 0.34$). However, C-EPM and D-EPM significantly outperform BLM and EM between top-2 and top-5 error rates ($p < 0.05$ using Turkey post-hoc analysis). The four methods achieve top-2 error rates at 6.47%, 5.91%, 4.19% and 4.43% respectively and further drop to 4.83%, 5.76%, 1.07% and 1.37%. Therefore, both C-EPM and D-EPM are effective methods to predict the target word when top-5 candidate words are provided.

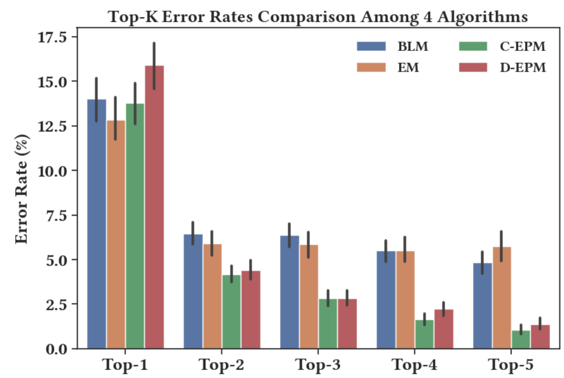

Figure 7: Top-k error rate comparison. Errors bars represent the standard errors of the mean.

6.9 Time interval of adjacent touch points

We measured the time interval between adjacent touchpoints (TI), which indicates a user's typing speed and typing confidence. Results show that both C-EPM and D-EPM have short TI, achieving an average time of 577 ms ($s.d. = 488$) and 612 ms ($s.d. = 537$). BLM and EM achieve 687 ms ($s.d. = 628$) and 715 ms ($s.d. = 676$) TI respectively. Auto-correction method has a significant effect on TI ($F_{3,60} = 12.2, \eta^2 = 0.38, p < .01$). Turkey post-hoc analysis indicates that C-EPM yeilds the best performance with respect to other methods ($p < .01$), and D-EPM is better than the two baseline methods ($p < .01$) we reviewed. We believe that the effectiveness of KDE method at filtering out the unintentional repetitive touches explains why D-EPM has a longer timer interval between adjacent touchpoints than C-EPM. Considering both the KSPC and the TI, D-EPM enables Parkinson users typing quicker with fewer keystrokes, which explains the high typing speed.

6.10 User Experience and Feedback

We collected participants' subjective ratings of different auto-correction methods using a 7-point Likert-scale questionnaire. The metrics we collected included perceived accuracy, perceived speed, perceived correction performance, self-confidence and overall preference. Cronbach's reliability α for the questionnaire was 0.95, confirming the internal consistency of the survey. Figure 8(right) shows the score of each metric. Results indicate that auto-correction technique significantly effects perceived accuracy ($\chi^2(3) = 18.8, p < .001$),

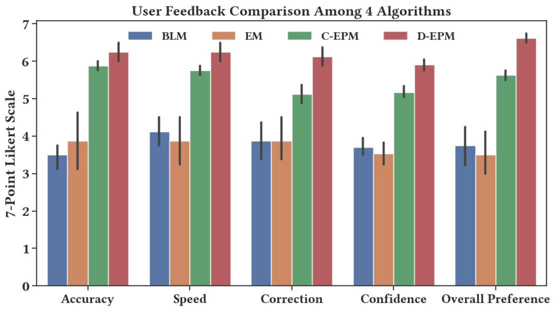

Figure 8: User feedback comparison among 4 auto-correction techniques. Errors bars represent the standard errors of the mean.

perceived speed ($\chi^2(3) = 17.2, p < .001$), perceived correction performance ($\chi^2(3) = 10.6, p < .05$), self-confidence ($\chi^2(3) = 49.9, p < .001$), and overall preference ($\chi^2(3) = 21.9, p < .001$). Wilcoxon signed-rank tests show that both C-EPM and D-EPM outform BLM and EM in perceived accuracy, perceived speed, self-confidence, and overall preference with $p < .05$. However, there is no significant difference between C-EPM and BLM ($p = .08$) or EM ($p = .10$). D-EPM outperforms C-EPM in perceived accuracy ($p < .05$), perceived correction performance ($p < .05$), self-confidence ($p < .001$), and overall preference ($p < .05$). There is no significant difference between BLM and EM methods on all metrics.

We also received positive feedback from users when they experienced the D-EPM method. P1 mentioned that "I think I did better on this keyboard." P2 mentioned that "I am sure that I typed faster on this keyboard. I would love to use it in the future." P4 mentioned that "I want to use this keyboard on my phone although I didn't like the interface. It is way more easier to use than the keyboard on my Android phone." P5 mentioned that "I thought I typed more accurate. I gained more and more confidence while I used it. I didn't need to correct the typos since it will show me what I wanted to type at last." P6 mentioned that "I definitely typed faster on this keyboard."

7 DISCUSSION AND FUTURE WORK

This paper has explored the feasibility of improving Parkinson's users' text entry performance on smartphone QWERTY keyboards. We not only analyzed their typing behaviors but also proposed algorithms to predict their input with various kinds of typing errors. We discuss the results and design implications in this section.

7.1 QWERTY Input for Parkinson's Users

Due to motor impairment symptoms, users with Parkinson's disease face greater challenges during touchscreen text entry than others. Existing works have found that these symptoms limit the usage of touchscreen phones among elderly people [8, 23] as well as that text entry and text correction [22] were the most challenging tasks. However, in our interviews, we found that Parkinson's users exhibited a strong preference the QWERTY keyboard over other text input techniques (e.g., T9 or voice input). This supports our motivation, facilitating Parkinson's users' text entry on smartphones with a QWERTY keyboard.

In Study 1, assuming text input can be corrected by the algorithm, experienced Parkinson's users could type 19.8 WPM, which is more comparable to non-Parkinson's users (29.4 WPM) than inexperienced Parkinson's users (4.73 WPM) [23]. Also, based on our touchpoint distribution results, Parkinson's users' systematic offset and size of spread were only about 20% greater than non-Parkinson users, which was relatively small compared with key sizes (6×9mm). This makes it possible to correct their input effectively.

However, users with Parkinson's disease committed more insertion, omission, and substitution errors (see Table 1) than other users, a major challenge. By generalizing the classical statistic decoding algorithm and modeling the spatial-temporal features of touch errors, our method achieved 22.8 WPM with 8% error rate in the evaluation study. This confirmed that achieving fluent text entry experience on a touchscreen keyboard for Parkinson's users is feasible.

7.2 Effectiveness of EPM

We proposed EPM, which generalized the classical statistic decoding algorithm, to correct Parkinson's users typing errors. The advantages of this model include: 1) all parameters have a direct physical interpretation and therefore can be easily summarized based on participants' typing data, making it more generalizable to different typing scenarios (e.g., touch-typing on tablets); 2) the calculated result can be interpreted as a probability, making it compatible with other auxiliary channels (e.g., accelerometer [5, 25]).

In Study 1, users with Parkinson's disease yielded an error rate of 20.2%. By comparison, in study 2, D-EPM reduced the CER to 22.4%, which significantly outperformed the classical statistical decoding algorithm and the pattern matching technique.

One feature of D-EPM is the dynamic parameter adjustment strategy, which leveraged spatial-temporal features to distinguish regular and unintentional repetitive touches. To quantitatively evaluate its effect, we compared C-EPM and D-EPM in the evaluation study. As expected, both of the techniques outperformed the two baseline techniques in terms of input speed, accuracy, and user preference. And although not significant, D-EPM achieved slightly better typing speed and error rate performance than C-EPM. This confirmed that leveraging spatial-temporal features is worthwhile.

7.3 Design Implications

Our findings on Parkinson's users' typing behavior and interviews on their preference provided a number of design implications for accessible keyboard techniques.

First, Parkinson's users preferred to use the QWERTY keyboard rather than T9 keyboard, mainly due to their experience and habit. Also, they usually use the hand with less severe hand tremor to perform one-hand typing on the keyboard, while the other hand holds the phone. This suggests that a QWERTY layout with a bigger key size would be helpful, rather than a small-sized keyboard to facilitate one-hand typing.

In study 2, D-EPM effectively corrected the participant's typing errors, enabling them to type fluently. Compared with BLM and EM, the results highlighted the necessity of correcting insertion and substitution errors for Parkinson's users. During interviews, some participants commented that although the algorithm can correct tolerant their input errors, they would welcome some more designs in typing feedback, e.g., large word font.

The top-k result in study 2 indicates that D-EPM can achieve good performance to predict the correct word at a low error rate after top-3. So we would recommend a future design to provide at least 4 top-ranked candidate words to the users.

7.4 Limitations and Future Work

We acknowledge that we compared to young adults rather than non-Parkinson's elders in section 4. Age difference has strongly correlates with typing performance [23]. The reason is that we supplemented the control group last to explore the unique touch behavior of Parkinson's users. By then, recruiting elderly users during the Covid-19 pandemic became very challenging. We recruited young adults in lieu of non-Parkinson's elders, which was a sub-optimal solution. However, this does not affect our major results on modelling Parkinson's users' typing behavior.

In future work, a larger user group could be recruited to further prove the idea that in particular contexts, a combination of general possibility models, e.g., EPM, and context-oriented models, could have better performances than pure general possibility techniques. Second, a dataset could be established for future text entry research concerning Parkinson's patients' specific demographic. In the future, we would like to investigate gesture typing for Parkinson's users [17]. Finally, we would like to investigate the performance of EPM on a broader group of users so we can evaluate our method's generalization ability.

8 CONCLUSION

In this paper, we proposed a smartphone QWERTY keyboard for users with Parkinson's disease based on the elastic probabilistic model. We first investigated Parkinson's users' typing behavior on smartphones. In particular, we identified and compared the typing characteristics generated by users with and without Parkinson's disease. We found that user with Parkinson's disease generated many more insertion, omission and substitution errors than young adults. We then proposed an elastic probabilistic model with a dynamic parameter adjustment feature (D-EPM) for input prediction. By incorporating both spatial and temporal features, D-EPM generalized the classical statistical decoding algorithm to correct various of typing errors, while maintaining direct physical interpretability. Our evaluation user study results confirmed that the proposed algorithm outperformed baseline techniques: users reached 22.8 WPM typing speed with a significantly lower error rate (8.0%) and better user experience and feedback than baseline techniques. We concluded that our technique could effectively facilitate Parkinson's users' text entry experience on smartphones.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China under Grant No. 2019YFF0303300, the Natural Science Foundation of China under Grant No. 62002198, No. 61902208, the grant from the Institute for Guo Qiang, Tsinghua University No. 2019GOG0003, and the China Postdoctoral Science Foundation under Grant No. 2019M660647. Our work is also supported by the Beijing Key Lab of Networked Multimedia, Undergraduate / Graduate Education Innovation Grants, Tsinghua University. We would like to thank all participants for their time, effort.

REFERENCES

[1] 2016. American National Corpus. http://www.anc.org/.

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

[3] Mohammed Belatar and Franck Poirier. 2008. Text Entry for Mobile Devices and Users with Severe Motor Impairments: Handiglyph, a Primitive Shapes Based Onscreen Keyboard. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '08)*. Association for Computing Machinery, New York, NY, USA, 209–216. https://doi.org/10.1145/1414471.1414510

[4] Leah Findlater and Jacob O. Wobbrock. 2012. Personalized input: Improving ten-finger touchscreen typing through automatic adaptation. In *Conference on Human Factors in Computing Systems - Proceedings*. 815–824. https://doi.org/10.1145/2207676.2208520

[5] Mayank Goel, Leah Findlater, and Jacob Wobbrock. 2012. WalkType: Using Accelerometer Data to Accomodate Situational Impairments in Mobile Touch Screen Text Entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2687–2696. https://doi.org/10.1145/2207676.2208662

[6] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. 2013. ContextType: Using Hand Posture Information to Improve Mobile Touch Screen Text Entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2795–2798. https://doi.org/10.1145/2470654.2481386

[7] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language Modeling for Soft Keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces* (San Francisco, California, USA) *(IUI '02)*. Association for Computing Machinery, New York, NY, USA, 194–195. https://doi.org/10.1145/502716.502753

[8] Toshiyuki Hagiya, Toshiharu Horiuchi, and Tomonori Yazaki. 2016. Typing tutor: Individualized tutoring in text entry for older adults based on input stumble detection. *Conference on Human Factors in Computing Systems - Proceedings* (2016), 733–744. https://doi.org/10.1145/2858036.2858455

[9] Toshiyuki Hagiya, Toshiharu Horiuchi, Tomonori Yazaki, Tsuneo Kato, and Tatsuya Kawahara. 2017. Assistive typing application for older adults based on input stumble detection. *Journal of Information Processing* 25, June (2017), 417–425. https://doi.org/10.2197/ipsjjip.25.417

[10] Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2501–2510. https://doi.org/10.1145/1978942.1979308

[11] Farzana Jabeen, Linmi Tao, Lin Tong, and Shanshan Mei. 2019. Modeling Chinese Input Interaction for Patients with Cloud Based Learning. (2019), 1–8. https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00206

[12] Shaun K. Kane, Jacob O. Wobbrock, Mark Harniss, and Kurt L. Johnson. 2008. TrueKeys: Identifying and correcting typing errors for people with motor impairments. In *International Conference on Intelligent User Interfaces, Proceedings IUI*. 349–352. https://doi.org/10.1145/1378773.1378827

[13] Masatomo Kobayashi, Atsushi Hiyama, Takahiro Miura, Chieko Asakawa, Michitaka Hirose, and Tohru Ifukube. 2011. Elderly User Evaluation of Mobile Touchscreen Interactions. In *Human-Computer Interaction – INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Springer Berlin Heidelberg, 83–99.

[14] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK 2: a large vocabulary shorthand writing system for pen-based computers. In *Proc. UIST'04*. ACM, 43–52.

[15] Per Ola Kristensson and Shumin Zhai. 2005. Relaxing stylus typing precision by geometric pattern matching. *International Conference on Intelligent User Interfaces, Proceedings IUI* (2005), 151–158. https://doi.org/10.1145/1040830.1040867

[16] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

[17] Yu Hao Lin, Suwen Zhu, Yu Jung Ko, Wenzhe Cui, and Xiaojun Bi. 2018. Why is gesture typing promising for older adults? comparing gesture and tap typing behavior of older with young adults. In *ASSETS 2018 - Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. 271–281. https://doi.org/10.1145/3234695.3236350

[18] I. Scott MacKenzie. 2015. https://www.yorku.ca/mack/RN-TextEntrySpeed.html

[19] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Conference on Human Factors in Computing Systems - Proceedings (CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971

[20] Kyle Montague, Hugo Nicolau, and Vicki L Hanson. 2014. Motor-Impaired Touchscreen Interactions in the Wild. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '14)*. Association for Computing Machinery, New York, NY, USA, 123–130. https://doi.org/10.1145/2661334.2661362

[21] Lilian Genaro Motti, Nadine Vigouroux, and Philippe Gorce. 2013. Interaction techniques for older adults using touchscreen devices: A literature review. *IHM 2013 - Actes de la 25ieme Conference Francophone sur l'Interaction Homme-Machine* (2013), 125–134. https://doi.org/10.1145/2534903.2534920

[22] Maia Naftali and Leah Findlater. 2014. Accessibility in Context: Understanding the Truly Mobile Experience of Smartphone Users with Motor Impairments. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '14)*. Association for Computing Machinery, New York, NY, USA, 209–216. https://doi.org/10.1145/2661334.2661372

[23] Hugo Nicolau and Joaquim Jorge. 2012. Elderly text-entry performance on touchscreens. In *ASSETS'12 - Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*. 127–134. https://doi.org/10.1145/2384916.2384939

[24] Francisco Nunes, Paula Alexandra Silva, João Cevada, Ana Correia Barros, and Luís Teixeira. 2016. User interface design guidelines for smartphone applications for people with Parkinson's disease. *Universal Access in the Information Society* 15, 4 (2016), 659–679. https://doi.org/10.1007/s10209-015-0440-1

[25] Katrin Plaumann, Milos Babic, Tobias Drey, Witali Hepting, Daniel Stooss, and Enrico Rukzio. 2018. Improving Input Accuracy on Smartphones for Persons who are Affected by Tremor using Motion Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 1–30. https://doi.org/10.1145/3161169

[26] Ondrej Polacek, Adam J Sporka, and Pavel Slavik. 2017. Text input for motor-impaired people. *Universal Access in the Information Society* 16, 1 (2017), 51–72. https://doi.org/10.1007/s10209-015-0433-0

[27] Élvio Rodrigues, Micael Carreira, and Daniel Gonçalves. 2014. Improving text-entry experience for older adults on tablets. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8515 LNCS, PART 3 (2014), 167–178. https://doi.org/10.1007/978-3-319-07446-7_17

[28] Élvio Rodrigues, Micael Carreira, and Daniel Gonçalves. 2016. Enhancing typing performance of older adults on tablets. *Universal Access in the Information Society* 15, 3 (2016), 393–418. https://doi.org/10.1007/s10209-014-0394-8

[29] Sayan Sarcar, Jussi Jokinen, Antti Oulasvirta, Xiangshi Ren, Chaklam Silpasuwanchai, and Zhenxin Wang. 2017. Ability-Based Optimization: Designing Smartphone Text Entry Interface for Older Adults, Regina Bernhaupt, Girish Dalvi, Anirudha Joshi, Devanuj K. Balkrishan, Jacki O'Neill, and Marco Winckler (Eds.). Springer International Publishing, Cham, 326–331.

[30] S Sarcar, J P P Jokinen, A Oulasvirta, Z Wang, C Silpasuwanchai, and X Ren. 2018. Ability-Based Optimization of Touchscreen Interactions. *IEEE Pervasive Computing* 17, 1 (2018), 15–26. https://doi.org/10.1109/MPRV.2018.011591058

[31] Weinan Shi, Chun Yu, Shuyi Fan, Feng Wang, Tong Wang, Xin Yi, Xiaojun Bi, and Yuanchun Shi. 2019. Vipboard: Improving screen-reader keyboard for visually impaired people with character-level auto correction. In *Conference on Human Factors in Computing Systems - Proceedings*. 1–12. https://doi.org/10.1145/3290605.3300747

[32] Weinan Shi, Chun Yu, Xin Yi, Zhen Li, and Yuanchun Shi. 2018. TOAST: Ten-Finger Eyes-Free Typing on Touchable Surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–23. https://doi.org/10.1145/3191765

[33] 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 *Conference on Human Factors in Computing Systems - Proceedings*. 113–120.

[34] Shari Trewin. 2003. Automating Accessibility: The Dynamic Keyboard. *SIGACCESS Access. Comput.* 77–78 (2003), 71–78. https://doi.org/10.1145/1029014.1028644

[35] Shari Trewin, Cal Swart, and Donna Pettick. 2013. Physical Accessibility of Touchscreen Smartphones. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2513383.2513446

[36] Radu-Daniel Vatavu and Ovidiu-Ciprian Ungurean. 2019. Stroke-Gesture Input for People with Motor Impairments: Empirical Results & Research Roadmap. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300445

[37] 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* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 659–668. https://doi.org/10.1145/2702123.2702135

[38] Chat Wacharamanotham, Jan Hurtmanns, Alexander Mertens, Martin Kronenbuerger, Christopher Schlick, and Jan Borchers. 2011. Evaluating Swabbing: A Touchscreen Input Method for Elderly Users with Tremor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 623–626. https://doi.org/10.1145/1978942.1979031

[39] Yuntao Wang, Chun Yu, Jie Liu, and Yuanchun Shi. 2013. Understanding Performance of Eyes-Free, Absolute Position Control on Touchable Mobile Phones. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Munich, Germany) *(MobileHCI '13)*. Association for Computing Machinery, New York, NY, USA, 79–88. https://doi.org/10.1145/2493190.2493215

[40] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain text entry on mobile devices. In *Conference on Human Factors in Computing Systems - Proceedings*. 2307–2316. https://doi.org/10.1145/2556288.2557412

[41] Daryl Weir, Simon Rogers, Roderick Murray-Smith, and Markus Löchtefeld. 2012. A user-specific Machine Learning approach for improving touch accuracy on mobile devices. In *UIST'12 - Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. 465–475. https://doi.org/10.1145/2380116.2380175

[42] Wikipedia contributors. 2020. Parkinson's disease — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Parkinson%27s_disease&oldid=970177846 [Online; accessed 31-July-2020].

[43] 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)*. Association for Computing Machinery, New York, NY, USA, 61–70. https://doi.org/10.1145/964696.964703

[44] 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, April (2017), 44–62. https://doi.org/10.1016/j.ijhcs.2017.05.001

[45] Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: Reimagining keyboard interaction. *Commun. ACM* 55, 9 (2012), 91–101. https://doi.org/10.1145/2330667.2330689

[46] Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: reimagining keyboard interaction. *Commun. ACM* 55, 9 (2012), 91–101.

[47] Yu Zhong, Astrid Weber, Casey Burkhardt, Phil Weaver, and Jeffrey P. Bigham. 2015. Enhancing android accessibility for users with hand tremor by reducing fine pointing and steady tapping. In *W4A 2015 - 12th Web for All Conference*. 1–10. https://doi.org/10.1145/2745555.2747277

[48] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an invisible keyboard. In *Conference on Human Factors in Computing Systems - Proceedings (CHI '18, Vol. 2018-April)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174013