There Is More to Dwell Than Meets the Eye: Toward Better Gaze-Based Text Entry Systems With Multi-Threshold Dwell

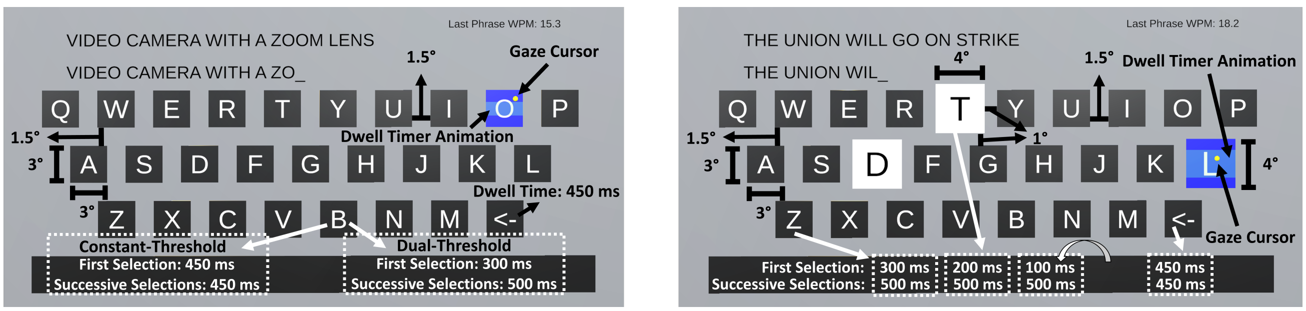
Aunnoy K Mutasim
School of Interactive Arts + Technology (SIAT)
Simon Fraser University
Vancouver, British Columbia, Canada
amutasim@sfu.ca

Mohammad Raihanul Bashar
Department of Computer Science & Software Engineering
Concordia University
Montreal, Quebec, Canada
mohammadraihanul.bashar@mail.concordia.ca

Christof Lutteroth
Computer Science
University of Bath
Bath, United Kingdom
c.lutteroth@bath.ac.uk

Anil Ufuk Batmaz
Department of Computer Science & Software Engineering
Concordia University
Montreal, Quebec, Canada
ufuk.batmaz@concordia.ca

Wolfgang Stuerzlinger
School of Interactive Arts + Technology (SIAT)
Simon Fraser University
Vancouver, British Columbia, Canada
w.s@sfu.ca



(a) Constant- and Dual-Threshold Dwell keyboards (CTD and DTD)

(b) Multi-Threshold Dwell keyboard (MTD)

Figure 1: We compared three keyboard designs: (a) The conventional Constant-Threshold Dwell (CTD) keyboard uses a single dwell threshold of 450 ms. The Dual-Threshold Dwell (DTD) keyboard reduces the dwell threshold for the first selection of a letter to 300 ms but increases the threshold to 500 ms for immediately following selections of that same letter, avoiding unintentional "double clicks" of the same letter. (b) The Multi-Threshold Dwell (MTD) keyboard additionally uses a third, further reduced, dwell threshold of 200 ms for up to three letters that are likely next targets. These keys are highlighted and enlarged. The spacebar is also assigned an even shorter threshold of 100 ms.

Abstract

Dwell-based text entry seems to peak at 20 words per minute (WPM). Yet, little is known about the factors contributing to this limit, except that it requires extensive training. Thus, we conducted a longitudinal study, broke the overall dwell-based selection time into six different components, and identified several design challenges and opportunities. Subsequently, we designed two novel dwell keyboards that use multiple yet much shorter dwell thresholds: Dual-Threshold Dwell (DTD) and Multi-Threshold Dwell (MTD). The performance analysis showed that MTD (18.3 WPM) outperformed both DTD (15.3 WPM) and the conventional Constant-Threshold Dwell (12.9 WPM). Notably, absolute novices achieved these speeds within just 30 phrases. Moreover, MTD's performance is also the fastest-ever reported average text entry speed for gaze-based keyboards. Finally, we discuss how our chosen parameters can be further optimized to pave the way toward more efficient dwell-based text entry.

CHI '25, April 26-May 1, 2025, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713781

CCS Concepts

• **Human-centered computing → Text input**; *Interaction techniques*; Pointing devices; Virtual reality.

Keywords

Text Entry, Eye Gaze, Eye-Tracking, Dwell Thresholds, Learnability, QWERTY

ACM Reference Format:
Aunnoy K Mutasim, Mohammad Raihanul Bashar, Christof Lutteroth, Anil Ufuk Batmaz, and Wolfgang Stuerzlinger. 2025. There Is More to Dwell Than Meets the Eye: Toward Better Gaze-Based Text Entry Systems With Multi-Threshold Dwell. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26-May 1, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3706598.3713781

1 Introduction

Entering text with one's eye gaze, which we call simply *gaze* here, has been investigated for a long time and is widely used in assistive gaze technologies [40, 59]. The most natural and common selection technique used in gaze keyboards and other gaze selection systems is *dwelling*, i.e., fixating one's gaze on a target for a certain dwell time [21, 29, 53, 59, 60, 79]. We refer to the required fixation time as the *dwell (time) threshold*. However, compared to button-based interfaces, dwell requires substantially more time to make a successful selection [62] and is thus typically perceived to be unnaturally slow [45, 81]. Although a longer dwell threshold slows down selection, shorter dwell results in unintentional non-target selections, i.e., increases the Midas Touch problem [34, 67, 88].

While previous work demonstrated text entry speeds of up to ~20 words per minute (WPM), this speed was only achievable with a fairly low dwell threshold of 300 ms, which required extensive training of users to shorten their fixations accordingly [53, 59, 60, 72]. Even after significant training, this speed still came at the cost of unintentional selections, requiring participants to make substantial error corrections as indicated by the high keystrokes per character (KSPC) measure (> 1.18) [21, 53, 72]. More importantly, there are indications that 20 WPM seems to be the maximum reachable for dwell-based text entry systems [31, 38, 53, 72], as evidenced by the plateauing performance of such systems and the shifting focus of researchers toward swipe-based gaze typing, e.g., [16, 31, 38, 41].

Still, we believe there is more to dwell than meets the eye as little is known about what factors contribute to the 20 WPM limit. It is well known that dwell selection performance is constrained by the need to fixate on the target for the duration of the dwell threshold [32]. Thus, previous work has explored different avenues to reduce this threshold. This includes but is not limited to adjusting the dwell threshold based on users' preference [21, 53, 72], their characteristics and performance [32, 59, 64, 66, 77, 89], and different features of the user interface [28, 59, 64, 69, 70]. To better understand dwell, researchers have also explored the components of dwell selection time, e.g., the time taken to point at the target [10, 24, 60, 72], to activate it with dwell [60], and then to exit the target [89]. However, these different components were investigated independently of each other, and the findings of the respective works are hard to compare due to differences in methodology, e.g., using dwell with or without varying dwell threshold, considering text entry or other applications, and/or testing either novice or trained users [10, 24, 60, 72, 89]. Thus, there is a gap in the literature as no one has systematically explored at a granular level how components of dwell selection time improve with training.

To address this gap, we conducted an 8-day longitudinal text entry study with a static dwell threshold of 450 ms, i.e., the conventional Constant-Threshold Dwell (CTD) keyboard shown in Figure 1a, with novices. In contrast to previous dwell-based longitudinal text entry studies [53, 59, 72], we kept the dwell time constant throughout the eight days to separate the learning associated with dwell selection without confounding it by the learning required for a variable/adjustable dwell time. Through the longitudinal study, we identified several design opportunities around challenges associated with the different components of the dwell selection time. For example, we verified that the burden of the (unnaturally long) 450 ms dwell threshold [45, 81] becomes more and more dominant as time progresses. We also verified that the time required to exit a key after successful selection with dwell can be substantial for novices and even approaches the dwell threshold. This, in turn, makes inadvertent "double clicks" more likely, particularly when the dwell threshold is reduced to 300 ms, making 300 ms dwell text entry systems practically unusable for novices, as demonstrated by our results and previous work [21, 44, 53, 59, 67, 72, 73].

By identifying suitable dwell thresholds that could prevent unintentional double clicks, we then designed two novel keyboards: Dual-Threshold Dwell (DTD) and Multi-Threshold Dwell (MTD; Figure 1). DTD, similar to CTD, uses a static dwell threshold. However, DTD increases the dwell threshold for *repeated* selections of the same letter: while the first selection of a letter uses a dwell threshold of 300 ms, any immediately successive selections of the same letter require an increased dwell threshold of 500 ms. This means that to select two 'O's (e.g., to type "ZOOM"), the first 'O' takes 300 ms, but the second 'O' takes 500 ms to select. MTD extends DTD by reducing the dwell threshold for the most likely letters: up to three such letters are highlighted and enlarged based on a word prediction algorithm. These likely letters can be selected in just 200 ms, while for other letters (except for repetitions of the last letter) the dwell time threshold is still 300 ms. For the spacebar, the threshold is also reduced even further to 100 ms.

Finally, we compared the CTD, DTD, and MTD keyboards in a user study. Absolute *novices* could type with an average dwell threshold of 313.4 ms and 233.9 ms, reaching 15.3 WPM and **18.3 WPM** with DTD and MTD, respectively. Not only were DTD and MTD significantly faster than CTD (12.9 WPM) while also requiring fewer error corrections compared to previous work [21, 53, 72], but *it took novices only 30 phrases to reach this level of performance*. MTD's exhibited performance is also the *fastest among previous gaze-based keyboards in terms of its average text entry speed*, with a last session/block WPM that is competitive even with multimodal approaches (Table 4). Thus, our results highlight the importance of better understanding the components of dwell to improve text entry performance. In summary, we make the following main contributions:

(1) A longitudinal analysis of dwell selection, which systematically investigates the different components of selection time and how they are affected by training.
(2) Two novel designs for dwell-based keyboards, Dual-Threshold Dwell (DTD) and Multi-Threshold Dwell (MTD), which address common problems and bottlenecks of conventional Constant-Threshold Dwell (CTD) keyboards.
(3) Evaluation results illustrating the utility of using multiple thresholds in dwell keyboards.

2 Literature Review

2.1 The Apparent 20 WPM Typing Speed Limit

Here, we review previous work on gaze-based text entry and its learnability.

2.1.1 Dwell-Based Text Entry. Dwell-based gaze keyboards for novices typically use dwell thresholds of 450–1000 ms, resulting in typing speeds between 5-10 WPM [44, 59]. To enhance their efficiency, Špakov and Miniotas [89] analyzed the time it takes for users to exit a key after it has been selected with different dwell thresholds, ranging from 300-900 ms. They found substantial variation in the exit time across users. Thus, they proposed an algorithm to select a user's dwell threshold based on their previous exit times, resulting in an average dwell threshold of 533 ms and 12.1 WPM. We also measure Exit Time (ET) in Study 1. However, instead of measuring the ET for different dwell thresholds, we analyzed how training with the same dwell threshold affects ET and then used this to guide the design of two novel keyboards, DTD and MTD. EyeBoard [66] and EyeBoard++ [77] changed dwell thresholds dynamically for the entire keyboard based on a user's performance. Yet, the demonstrated speed was just 5.02 WPM for EyeBoard and 9.63 WPM for EyeBoard++.

Majaranta et al. [53] conducted a longitudinal study where participants were allowed to adjust the dwell threshold according to their preference. Eleven participants typed on a QWERTY keyboard over ten separate days, with each day/session lasting 15 minutes. The findings demonstrated a notable improvement in typing speed, increasing from 6.9 WPM in the initial session to 19.9 WPM in the final one. Furthermore, the average dwell threshold decreased substantially from an initial average of 876 ms to 282 ms in the 10^{th} session. Even more extensive training in a longitudinal study with 19 fifteen-minute sessions over several days was conducted by Räihä and Ovaska [72]. Similar to Majaranta et al.'s study [53], their [72] participants were allowed to adjust the dwell threshold in the first 10 sessions, also yielding similar results: an average typing speed of about 20 WPM with dwell thresholds set between 240 and 340 ms.

Although decreasing the dwell time achieved a speed of 20 WPM [53, 72], the shorter dwell thresholds increased the Midas Touch problem, i.e., the unintentional selection of non-targets [57, 67]. This, in turn, increased the KSPC from 1.09 in the first to 1.18 in the last session [53] as users needed more corrective actions to fix the typos. An even higher average KSPC of about 1.25 was reported by Räihä and Ovaska [72].

Räihä and Ovaska [72] also discovered that the *slack*, i.e., the time needed to find and fixate on a key, remains relatively consistent across different dwell times, accounting for a portion of the total key selection time. The slack was ~300 ms for users undergoing training and ~250 ms for trained users. However, the authors imposed "slack", i.e., a minimum wait time, of 150 ms between consecutive selections of the same key to avoid inadvertent double clicks of that key. Similarly, we also make consecutive selections of the same key harder in DTD and MTD by increasing the dwell threshold of these consecutive selections to 500 ms.

Diaz-Tula and Morimoto [21] also allowed participants to adjust their dwell time. However, their main contribution was the AugKey system that "augments" keys with a prefix, allowing continuous review of the text typed, and suffixes, providing word predictions to speed up typing. In an evaluation, participants typed significantly faster (16.7 WPM) while requiring fewer corrections (~0.8 KSPC) with AugKey compared to two other baseline conditions.

To improve the performance of dwell-based systems, Mott et al. [59] experimented with dynamically adjusting the dwell threshold proportional to the likelihood of a given key. Their approach made it more difficult to select keys that are less likely while making it easier to select more likely keys. The authors argue that this should not disrupt users' typing rhythm, despite the dwell threshold changing for every key press. Yet, even after training participants in a longitudinal study, the maximum average speed of 13.7 WPM (334 ms dwell threshold) stayed far below the 20 WPM mark. This suggests that an always-changing dwell threshold may potentially be disruptive. In contrast, our MTD keeps the dwell thresholds more consistent by using a 200 ms threshold for a few likely letters and 300 ms for all others.

2.1.2 Dwell-Free Text Entry. Addressing the apparent plateauing of the performance of dwell-based systems, and to avoid the burden of a lengthy dwell threshold, several dwell-free approaches have been proposed. Urbina and Huckauf's pEYEwrite [87] used bigrams and word prediction, which yielded 13.47 WPM. Patidar et al.'s improved pEYE (pie) layout [67] demonstrated 6.1 WPM. Bee and Andre's Quikwriting [8] yielded 7.8 WPM. Sarcar et al.'s EyeK [78] achieved 6 WPM. Morimoto and Amir's context switching (CS) interface [57] with a duplicated QWERTY keyboard yielded 12 WPM. In later work [58], the original CS keyboard (13.4 WPM) outperformed two alphabetically ordered CS keyboards: single-line (8.7 WPM) and dual-line (9.5 WPM).

Dasher [90] is a popular text entry method in the literature. However, there is currently no consensus on its speed [41]. Previous studies [38, 76] report a range of 16–26 WPM, where 26 WPM was achieved by just a single user. Averaging across the range of results, we agree with previous work [38] in that Dasher's typical maximum performance is most likely comparable to the 20 WPM demonstrated by previous dwell-based studies [53, 72].

Recent gaze-based text entry research has shifted focus to gaze-based swipe typing. The primary reason for this shift is the work by Kristensson and Vertanen [38], where the authors showed that if a "perfect recognizer" is employed, swipe typing can reach 46 WPM. In swipe typing, users have to look at or near the vicinity of the target key, then move over to the next target key, and so on, without having to stop to dwell over each key. In other words, looking at each letter of the target word one key at a time enters that word. However, without a "perfect recognizer", the challenge for such systems is to identify when the gaze path started and ended during typing [16].

Pedrosa et al.'s [68] solution to this problem was to filter out letters inadvertently chosen from the sequence of letters the user gazed at. This filtering was done with the help of a word prediction algorithm. Evaluation of this dwell-free system, Filteryedping, resulted in participants typing at an average of 15.95 WPM. EyeSwipe [41] took a different approach by requiring users to explicitly mark the first and last letters of the intended word using target reverse crossing [24]. Similar to Filteryedping [68], the system then shortlists and suggests a set of candidate words using the gaze path and the first and last letters. A user study demonstrated 11.7 WPM after 30 minutes of practice. A similar study [39] with gaze and pressing a button on a touchscreen device for the first/last characters, named TAGSwipe, showed that TAGSwipe achieved a typing speed of 15.5 WPM in comparison to EyeSwipe (8.84 WPM) and dwell (8.48 WPM).

Several other solutions to identify the first/last character have been proposed. This includes pressing and holding a physical button [92], employing probabilistic algorithms [16], 'nod' and 'shake' head gestures [25], and dwell and context switching [42]. So far, the fastest speed in the literature was achieved by Hedeshy et al. [31]: in Hummer, users signpost word boundaries by humming continuously from the start to the end of their gaze path. Hummer showed a commendable speed of 20.5 WPM, outperforming EyeSwipe (12.0 WPM) after participants typed just 30 phrases with each technique.

Yet, users are limited by the word prediction algorithm with swipe keyboards and thus can only type dictionary/corpus words. While we also use word predictions like swipe keyboards and other work [1, 2, 28, 51, 59, 70], users can still type whatever they need with MTD at relatively high speeds (i.e., similar to the speed of DTD), including but not limited to names, passwords, special characters, numbers, and transliterations of a different language. In swipe keyboards, users must also constantly switch their attention back and forth between typing and searching for the target word in the list of predicted words. Not only does this switching and searching add cognitive load [9], but users also waste valuable time in the process [1]. In contrast, MTD compensates for this time loss by implicitly incorporating word predictions into the natural flow of typing.

2.2 Improving Dwell Selection Efficiency

Dwell selection performance is limited by the requirement of dwelling the gaze over the target for the entirety of the dwell threshold [32]. Thus, and as discussed in the previous subsection, researchers have investigated how this dwell threshold can be reduced by adjusting it based on different criteria [21, 28, 53, 59, 66, 70, 72, 77, 89].

Researchers have also attempted to reduce dwell thresholds beyond text entry systems. Penkar et al. [69] adjusted the dwell threshold depending on the size of the buttons. In addition to adapting to the type of the button, Nayyar et al. [64] took a user-specific approach to dynamically adjust the dwell threshold. Isomoto et al. [32] used Fitts' law estimates to reduce the dwell threshold. An evaluation of their technique revealed an average dwell threshold of just 86.7 ms but at the cost of an error rate of 10.0%. This high error rate limits the applicability of this technique for many use cases, including text entry.

To reduce the Midas Touch problem, i.e., unintentional selections, Becker [7] suggested assigning higher dwell thresholds for similar-looking targets. Isomoto et al. [33] investigated machine learning techniques to predict users' intent to avoid inadvertent dwell activations. MacKenzie and Zhang [51] used letter prediction to predict and highlight 3 letters, similar to our MTD. They used this prediction to steer the gaze cursor away from unlikely letters and toward likely letters to improve pointing and, therefore, reduce typos.

2.3 Learnability in Text Entry

Human learning substantially improves text entry performance over time [35]. For this reason, the most valid way to evaluate text entry systems is to train participants over a significant period, i.e., through longitudinal studies [61]. Analyzing the learnability of a system by fitting a regression based on the power law of learning through the longitudinal WPM data and using this learning curve to predict expert user performance is also common in the literature [13, 14, 35, 47, 49, 61, 63, 65]. The other approach to predicting expert performance is model-based approaches, such as those building on the KLM model [12] or similar variants [46, 74]. However, these works primarily focus on movement time, and therefore text entry speed by modeling visual search [35, 50, 80], human memory [3], and other factors [17–20]. Instead of directly focusing on typing speed overall, we focus on the different components of each dwell interaction, i.e., at a more granular level, with the overall objective of designing better dwell-based (text entry) systems.

3 User Study 1 – Longitudinal

As mentioned, little is known about what factors contribute to the 20 WPM limit achievable by dwell-based text entry systems [38, 53, 72]. To address this issue, we systematically investigated the time components of dwell-based selection and how these components improve with training by conducting an 8-day longitudinal study in Virtual Reality (VR). We chose to conduct the study in VR because the eye trackers in today's VR headsets are fairly easy to use and calibrate, while also providing overall good performance. Moreover, previous work [73] found that gaze-based text entry is a viable input method for VR, with a novice typing speed (9.36 WPM) that is in line with findings of non-VR-based gaze typing studies (typically 5-10 WPM) [44, 59].

3.1 Keyboard Design

Figure 1a shows our keyboard design. This keyboard layout was designed based on related work [29, 58, 61, 71, 73, 79]. The width and height of each key were set to 3°, and we decided to use a 1.5° gap between keys to avoid unintended triggering of neighboring keys in case of low tracking accuracy [61, 73]. Like in some smartphone keyboard layouts, the backspace key was added next to the 'M' key in the QWERTY layout. The spacebar was 43.5° wide and 3° high. The keyboard was world-fixed in VR and its center was placed two meters away from the participant at eye level, i.e., well out of arms' reach.

We consistently used a novice-friendly dwell time threshold of 450 ms [44, 59] throughout the 8-day experiment. Whenever the user's gaze cursor came in contact with a key, the system highlighted that key in blue and started an animation showing the progress of the dwell timer [55]. When the dwell threshold was reached, the key was selected, and users were given confirmatory auditory and visual feedback, highlighting the key in green for 100 ms [85]. To encourage participants to improve their speed, we showed them the typing speed for their last completed phrase.

3.2 Apparatus

The study was conducted on a computer with an i7-12700H processor, 16 GB RAM, and an RTX 4060 graphics card using Unity. We used the Meta Quest Pro VR headset, which has a resolution of 1800×1920 pixels per eye, 95.57° (diagonal) FOV, and a 90 Hz refresh rate. Before every session, the eye tracker was calibrated and validated using Meta's built-in calibration and validation methods.

3.3 Procedure

On the first day, participants signed a consent form and completed a demographic questionnaire about their age, gender, and QWERTY typing experience. Subsequently, they performed their first 15-minute session, where they typed randomly selected phrases from MacKenzie and Soukoreff's corpus [48] (500 phrases with minimum, maximum, and average phrase lengths of 16, 43, and 28.6 characters, respectively). These 15-minute training sessions continued for seven more days, with the last session being on the eighth day. To account for weekends, we allowed gaps of up to two days between sessions within a 12-day window.

If the participants were in the middle of typing a phrase when the 15-minute timer expired, the session was ended by the software only upon the completion of that phrase. Participants ended every phrase by typing an extra "space". They were instructed to correct any mistakes they noticed immediately, i.e., within the current word, but to ignore errors that occurred two or more words back [53, 58, 61, 63, 87]. Also, to speed up the typing process, participants were discouraged from looking back at the (partially) typed phrase, i.e., they were instructed to keep in mind what they had typed so far and what to type next. We also told them to take as long as they needed to memorize each target phrase and (if needed) its spelling before entering it (and this time was not counted). Once they started typing the phrase, they were asked to finish it as fast as possible.

Following previous work [53, 72, 76, 86] and to minimize participants' (eye) fatigue, we limited a session to 15 minutes. The eight 15-minute sessions yielded a total training time of (8 × 15 = 120 minutes =) 2 hours, which is comparable to related works (Table 4).

3.4 Performance Metrics

We chose the following metrics to evaluate performance in the longitudinal study:

- *Words per minute (WPM)*, which is the average number of words typed every minute. Here, the definition of a word is the sequence of any five characters [4] including spaces but excluding backspaces. For example, "A DAY" is one word, and "THE" is 0.6 words.
- *Keystrokes per character (KSPC)* represents the average number of keys selected to (correctly) type a single character. More precisely, KSPC is the ratio of the total number of keys selected to the length of the typed text [83]. This means that KSPC includes the number of times the backspace key was hit.
- *Minimum String Distance Error Rate (MSD ER)*, where MSD is the minimum amount of changes required – insertions, deletions, and substitutions – to convert one string to another. Here, we use the MSD ER metric formulation proposed by Soukoreff et al. [84] to compute the difference between the target and the typed text.

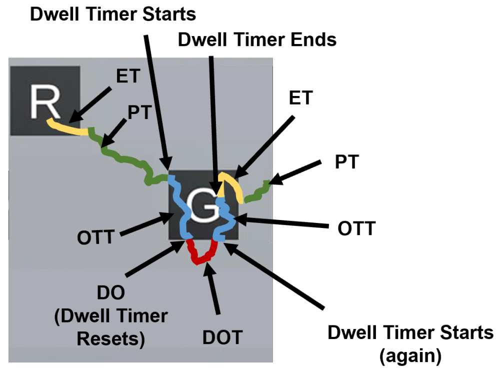


Figure 2: The Components of Dwell Selection — Exit Time (ET; yellow), Pointing Time (PT; green), On Target Time (OTT; blue), Drop-off Time (DOT; red), and Drop-offs (DOs). Here, 'R' and 'G' are the last and next selected keys, respectively, e.g., while typing "TARGET". Exactly one DO is shown.

In addition to the above metrics commonly used to evaluate text entry systems, we further analyzed the data by breaking down the actions involved in a dwell-based selection of each letter, step-by-step (Figure 2):

(1) *Exit Time (ET)*: ET, following previous work [89], is defined as the time taken to exit the key after it has been selected. Technically, this is the time from when a key was selected until the first time the gaze leaves the key.

(2) *Pointing Time (PT)*: The next step is pointing at the next key. Thus, PT represents the time participants take to point to the next selected key after the gaze cursor has left the previously selected key. Unlike previous work [10, 24, 60, 72], we do not include the ET in the PT to enable more fine-grained analysis. Note that, because participants could have made one or more mistakes, the two selected keys might not be the same as the target keys.

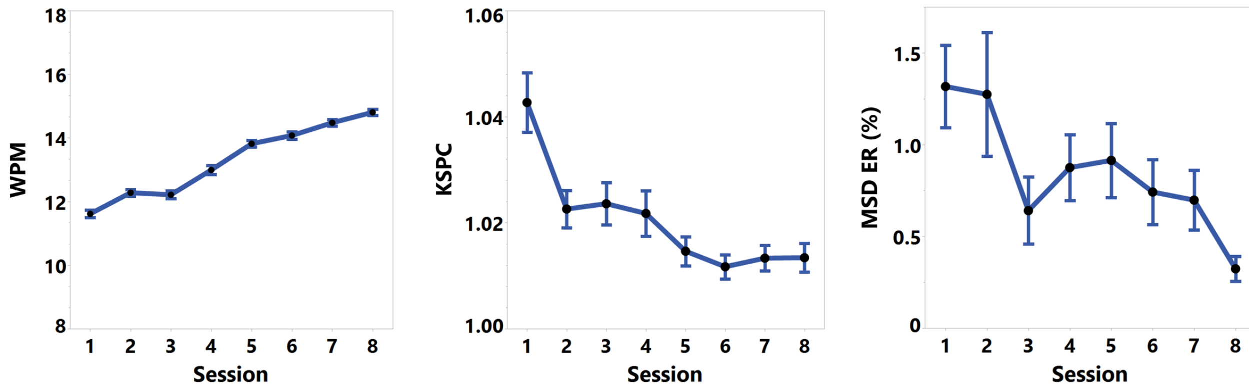
(3) *Activation Time (AT)*: The final step in key selection is activation. AT represents the time taken to activate a key after the first time the gaze reached that key. We break AT down into On Target Time (OTT) and Drop-Off Time (DOT), with

$$AT = OTT + DOT \tag{1}$$

where,

(a) *On Target Time (OTT)* is the whole time spent by participants dwelling on a key to select it. The motivation for measuring the OTT is that jitter is inevitable in eye-tracking [9] and is the primary cause for (undesired) resetting of the (450 ms) dwell timer [30]. Thus, the time users have to dwell on a key can, in reality, be larger than the (450 ms) dwell threshold.

(b) *Drop-off Time (DOT)* is the time participants lost due to jitter. In other words, this is the time lost due to involuntarily falling off the selected key after the gaze cursor had reached the key for the first time.



(a) (b) (c)

Figure 3: Study 1 results for (a) WPM, (b) KSPC, and (c) MSD ER by Sessions. The error bars show the standard error of means.

Further, *Total Time (TT)* is the combined time required to select a key and is defined as:

$$TT = ET + PT + AT = ET + PT + (OTT + DOT) \tag{2}$$

Finally, to better understand how eye-tracking jitter affects selection time, we also analyzed:

- *Drop-Offs (DOs)*, i.e., how many times the gaze fell off the selected key after reaching it.

3.5 Participants

We recruited 9 novice participants (6 males, 3 females), aged 28.2 ± 4.99 years, who all had no visual impairments or corrected-to-normal vision and at least 9 years of QWERTY typing experience. None of the participants had prior experience with eye-tracking. They were remunerated with the equivalent of US $15 in the local currency for every set of four 15-minute sessions and an additional $15 for completing the study.

3.6 Results

3.6.1 WPM, KSPC, and MSD ER. Participants' average typing speed started at 11.6 ± 1.77 WPM (mean ± SD) in the first session and reached 14.8 ± 1.63 WPM in the last one (see Figure 3a). The fastest participant typed at 16.9 ± 1.30 WPM in session 7.

The data was analyzed in SPSS 29 using repeated measures (RM) ANOVA with **Session** as the independent variable and $\alpha = .05$. The data was considered to be normal when Skewness (S) and Kurtosis (K) values were between ±1.5 [27, 56]. Upon violation of Mauchly's sphericity test, we applied Huynh-Feldt correction where $\epsilon < 0.75$. We transformed the data using ART [91] for dependent variables that did not have a normal/log-normal distribution. Post-hoc analyses were conducted using the Bonferroni method.

One-way RM ANOVA identified a significant difference between the sessions for WPM ($F_{4.26,34.1} = 21.4$, $p < .001$, $\eta^2 = 0.728$). While post-hoc analysis revealed several significant differences between sessions, we highlight that there was no significant improvement in typing speed starting from the 6$^{\text{th}}$ session onwards, i.e., we started to see signs of the performance plateauing.

Participants improved their average KSPC from 1.043 ± 0.082 in the first session to 1.013 ± 0.045 in the last session (Figure 3b), with the differences being overall significant ($F_{7,56} = 2.52$, $p < .05$, $\eta^2 = 0.240$). However, post-hoc analysis did not identify significant differences between sessions. MSD ER improved from 1.32 ± 3.30% to 0.32 ± 1.14% (Figure 3c), with overall significant differences

Table 1: Study 1 results for average Exit Time (ET), Pointing Time (PT), On Target Time (OTT), Drop-off Time (DOT), Activation Time (AT), Total Time (TT), and Drop-offs (DOs) over the eight sessions.

	Exit Time (ET)		Pointing Time (PT)		On Target Time (OTT)		Drop-off Time (DOT)		Activation Time (AT)		Total Time (TT)	Drop-offs (DOs)
Session	Time (s)	% of TT	Time (s)	% of TT	Time (s)	% of TT	Time (s)	% of TT	Time (s)	% of TT	Time (s)	#
1	0.220 ± 0.10	22.1%	0.168 ± 0.24	16.8%	0.531 ± 0.16	53.2%	0.079 ± 0.21	7.87%	0.609 ± 0.34	61.1%	0.997 ± 0.43	0.498 ± 1.09
2	0.229 ± 0.09	23.9%	0.139 ± 0.20	14.5%	0.524 ± 0.16	54.5%	0.068 ± 0.19	7.11%	0.592 ± 0.32	61.6%	0.961 ± 0.40	0.478 ± 1.08
3	0.230 ± 0.09	24.0%	0.133 ± 0.19	13.9%	0.527 ± 0.16	54.9%	0.070 ± 0.21	7.27%	0.597 ± 0.35	62.1%	0.961 ± 0.42	0.518 ± 1.19
4	0.208 ± 0.09	22.7%	0.129 ± 0.19	14.1%	0.519 ± 0.15	56.7%	0.059 ± 0.19	6.47%	0.578 ± 0.31	63.2%	0.915 ± 0.40	0.444 ± 1.08
5	0.203 ± 0.08	23.7%	0.107 ± 0.16	12.4%	0.507 ± 0.13	59.2%	0.040 ± 0.13	4.71%	0.547 ± 0.24	63.9%	0.857 ± 0.30	0.339 ± 0.87
6	0.188 ± 0.08	22.3%	0.102 ± 0.14	12.1%	0.508 ± 0.13	60.2%	0.046 ± 0.15	5.44%	0.554 ± 0.26	65.6%	0.844 ± 0.32	0.376 ± 0.99
7	0.177 ± 0.08	21.7%	0.096 ± 0.13	11.8%	0.503 ± 0.12	61.7%	0.039 ± 0.13	4.79%	0.542 ± 0.24	66.5%	0.815 ± 0.29	0.317 ± 0.84
8	0.171 ± 0.08	21.4%	0.095 ± 0.13	11.9%	0.499 ± 0.12	62.4%	0.034 ± 0.12	4.27%	0.533 ± 0.23	66.7%	0.799 ± 0.27	0.280 ± 0.81
$S1 - S8$	0.049	24.7%	0.073	36.7%	0.032	16.1%	0.044	22.4%	0.076	38.5%	0.199	0.218
Average	0.203	22.8%	0.121	13.6%	0.515	57.6%	0.054	6.09%	0.569	63.7%	0.894	0.406

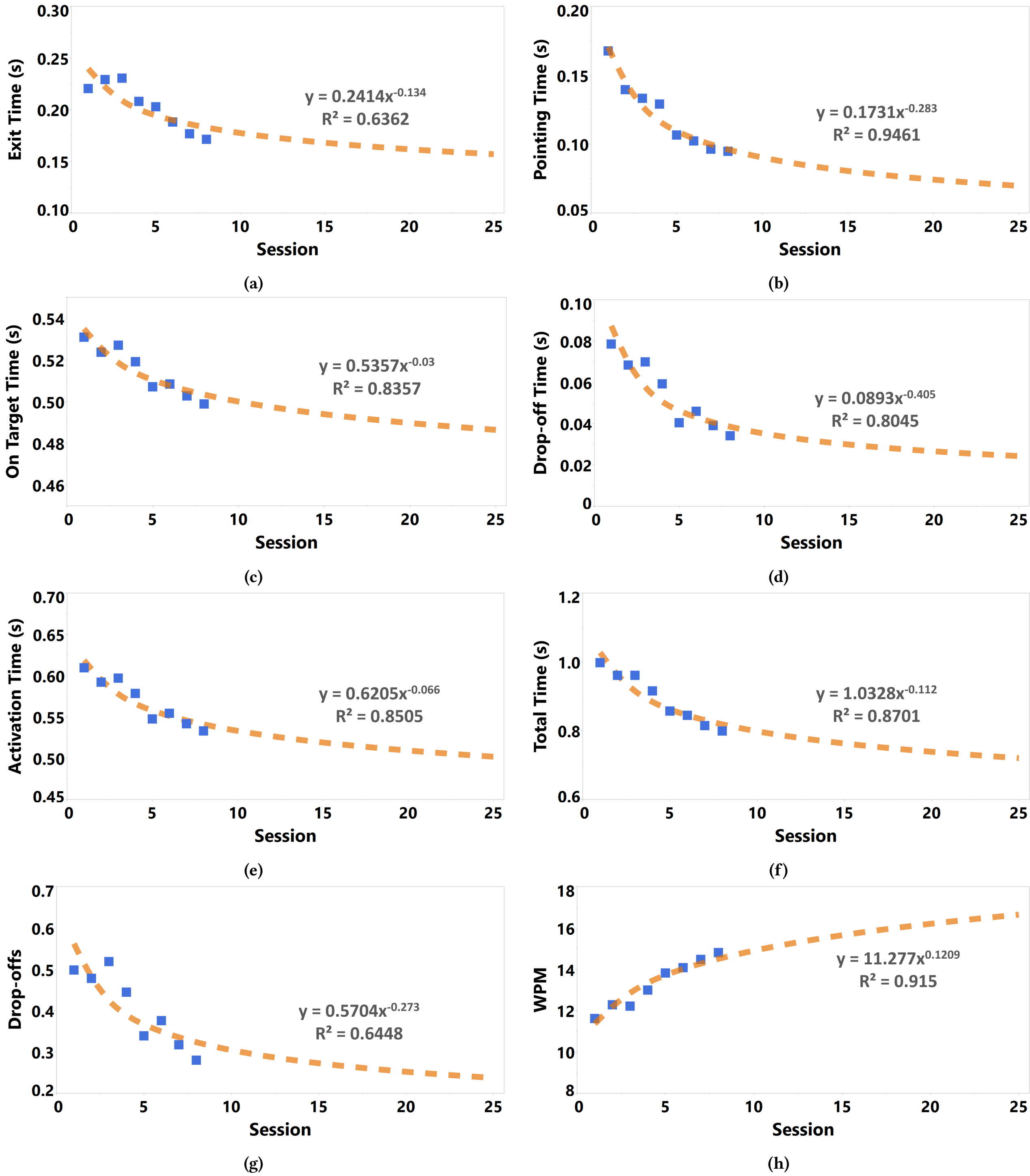


Figure 4: Study 1 results for average (a) Exit Time (ET), (b) Pointing Time (PT), (c) On Target Time (OTT), (d) Drop-off Time (DOT), (e) Activation Time (AT), (f) Total Time (TT), (g) Drop-offs (DOs), and (h) WPM over the eight sessions along with an extrapolation of the learning curve to the 25th session.

($F_{7,56} = 3.58$, $p < .01$, $\eta^2 = 0.309$). Except for session 2 being significantly different from session 8 ($p < .01$), there were no significant differences between sessions.

3.6.2 Components of Dwell Selection. As shown in Table 1 and Figure 4, all the seven components involved in the dwell selection

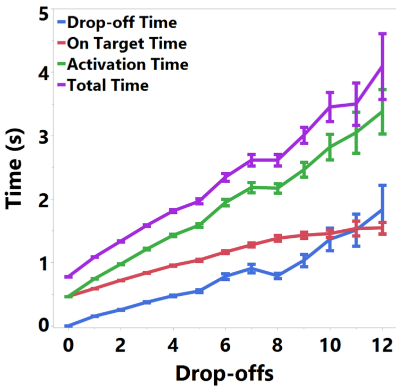


Figure 5: Study 1 results for On Target Time (OTT), Drop-off Time (DOT), Activation Time (AT), and Total Time (TT) over Drop-offs (DOs). The error bars show the standard error of means.

of each key exhibited a decreasing trend. In other words, the training made participants faster in terms of ET, PT, OTT, and DOT (and, therefore, AT and TT), as well as decreased DOs.

Total Time (TT) decreased from 0.997 s to 0.799 s, with the highest contributor to this decrease being PT which improved by 0.073 s (i.e., 36.7% of the overall improvement of TT). ET and DOT were the two next highest contributors, improving by 0.049 s and 0.044 s, corresponding to 24.7% and 22.4% of the overall improvement, respectively. The least improvement was observed for OTT, which improved by 0.032 s, i.e., 16.1% of TT's change. DOs reduced by about 44%, going hand in hand with reduced OTT and DOT, and therefore, AT and TT (Figure 5).

The results in Table 1 also show what percentage each of the components contributes toward TT. All components show a decreasing trend across sessions except for ET, OTT, and AT. An increasing trend is observed for OTT and AT. However, ET roughly stays the same, i.e., around 22% throughout.

3.6.3 Learning Curves. Using the regression function of SPSS 29, we fitted power law learning curves to the WPMs and the selection time components [35, 47, 49] (Figure 4). This yielded Equations 3-10, where *session* is the number of 15-minute typing sessions and R^2 is the coefficient of determination, i.e., a measure of model fit describing the proportion of the variation in the respective dependent variable that can be explained by *session*. All R^2 values, except for ET and DOs, are "substantial" [15], suggesting that the learning curves describe the changes in the variables well.

$$ET = 0.2414 \times session^{-0.134}, R^2 = 0.6362 \tag{3}$$

$$PT = 0.1731 \times session^{-0.283}, R^2 = 0.9461 \tag{4}$$

$$OTT = 0.5357 \times session^{-0.03}, R^2 = 0.8357 \tag{5}$$

$$DOT = 0.0893 \times session^{-0.405}, R^2 = 0.8045 \tag{6}$$

$$AT = 0.6205 \times session^{-0.066}, R^2 = 0.8505 \tag{7}$$

$$TT = 1.0328 \times session^{-0.112}, R^2 = 0.8701 \tag{8}$$

$$DOs = 0.5704 \times session^{-0.273}, R^2 = 0.6448 \tag{9}$$

$$WPM = 11.277 \times session^{0.1209}, R^2 = 0.915 \tag{10}$$

3.7 Study 1 Discussion

We investigated how the components of dwell selection improve with training over eight days. The results show that participants' performance improved across all seven components of dwell selection (Table 1 and Figure 4), leading to an improvement in text entry speed from 11.6 WPM to 14.8 WPM (Figures 3a and 4h).

Pointing Time (PT) showed the largest improvement (Table 1), suggesting that participants learned the exact locations of the keys and required fewer corrective secondary saccades [79]. Exit Time (ET) also improved substantially, suggesting participants' anticipation of when the dwell threshold was reached got better with training, and they could anticipate more accurately when they could move toward the next key. Interpolating the ET for a 450 ms dwell threshold from the corresponding figure in previous work [89] to range between ∼210 ms and ∼ 250 ms, our observed ETs are still ∼40 ms and ∼30 ms faster for trained and novice users, respectively. We attribute this difference to the fact that only four phrases had been typed in each of the seven different dwell threshold conditions in said previous work [89]. In contrast, our result of 266 ms "slack", i.e., $ET + PT$ in the last session, is quite similar to the previously reported value of ∼250 ms for trained users [72], suggesting our findings are representative and reliable.

Based on our observations, participants also learned to better interact with dwell selection more generally, such as learning how to deal with eye-tracking jitter and accuracy limitations, and learning how to control their eyes for not just perceiving the environment but also for interacting with it [54]. Moreover, eye-tracking improved because participants learned to position the VR headset more appropriately on their heads, resulting in better eye-tracking calibration. Improved calibration contributed to the reduction in Drop-offs (DOs) (Table 1), which in turn improved Drop-off Time (DOT), On Target Time (OTT), and therefore, Activation Time (AT) and Total Time (TT). Besides illustrating improvements through training, Study 1 also highlighted some challenges and opportunities in dwell keyboard design:

ET While ET decreases with training, it can still be substantial for novices and come close to the dwell threshold. For example, in the first three sessions, ET was longer than 250 ms and 300 ms for ∼**50%** and ∼**20%** of the total selections, respectively. Thus, the ET can easily surpass the dwell threshold and lead to inadvertent double clicks of letters, especially if the dwell threshold is set to, e.g., 300 ms [53, 59, 72]. This could result in making mistakes ∼20% of the time, which is too high for practically relevant text entry solutions [5]. This points to a potential design opportunity by preventing the naturally longer ET from causing inadvertent double clicks.

PT PT for novices was still substantial, likely because quickly finding the right key with the gaze is quite different from finding it with the fingers. This raises the question of whether visual search can be aided in situations where some keys are much more likely than others.

OTT Although OTT improved over the sessions, it contributed more heavily toward TT as the sessions progressed, because improvements in the other components overshadowed the improvement for OTT. The main "culprit" for this is the dwell threshold. This highlights the importance of reducing the dwell threshold whenever possible but judiciously without compromising typing rhythm or accuracy.

DOs Unsurprisingly, fewer DOs resulted in better Drop-off Time (DOT) and, as the dwell timer was reset less frequently, also led to shorter OTT and therefore, AT and TT (Figure 5), thus identifying another design opportunity.

For the learning curves in Equations 3-10, a linear regression results in a better fit for a few of these measures. Yet, we are skeptical that any of these measures could reach zero, as it is unrealistic to expect a linear behavior for human learning.

To more directly compare our results with previous work [53, 72] we also calculated an estimated typing speed based on a dwell threshold of 282 ms and an additional KSPC of 1.18. We estimated the average time per character by subtracting the difference in dwell thresholds from TT for each key press and multiplying it by 1.18. This calculation predicted an average last-day typing speed of 19.5 ± 3.17 WPM. This result is very close to the ~20 WPM reported by previous work [53, 72], suggesting that our findings are comparable.

4 The Dual-Threshold and Multi-Threshold Dwell Keyboards

Guided by the results of Study 1, we designed two novel dwell keyboards, Dual-Threshold Dwell (DTD) and Multi-Threshold Dwell (MTD). DTD addresses the challenge of the ET, reducing the chance of inadvertent double clicks of a letter by introducing a second, larger dwell threshold for repeated clicks. In addition, MTD addresses the challenges of the PT by highlighting likely keys, and the OTT by offering reduced dwell thresholds for likely keys and the spacebar. The reduced dwell threshold for a key also addresses the DO challenge as a shorter threshold typically means fewer DOs [30]. These two new designs improve the design of the conventional Constant-Threshold Dwell (CTD) keyboard by judiciously introducing new types of well-defined and predictable dwell thresholds.

4.1 Dual-Threshold Dwell (DTD)

For the Dual-Threshold Dwell (DTD) keyboard, we chose a static dwell threshold of 300 ms, a threshold previously known to be impractical for novices [21, 44, 53, 59, 67, 72, 73]. As found in Study 1, one reason behind this issue is the inadvertent double clicks of the same key due to the naturally longer ET of novices. To address this, we averaged across the ETs of all the sessions of Study 1 (Table 1) and increased the dwell threshold for successive selections of the *same* key by an additional 200 ms, i.e., (300 + 200 =) 500 ms (see Figure 1a). We found during our early pilots (that followed the same experimental design as in Study 2, involving the research team and some novice users) that this 200 ms delay worked well with novices.

We also observed in such pilots that changes in the dwell threshold between successive backspace selections confused participants as to how many characters that action deleted, especially when they attempted to delete multiple characters to fix typos. When using a larger threshold for repeated backspace selections, users tended to go back and forth between checking the typed text and using the backspace key, thus wasting time. Consequently, we chose a *constant* dwell threshold of 450 ms for backspaces, even for repeated clicks.

4.2 Multi-Threshold Dwell (MTD)

To further reduce the dwell time threshold, we experimented with dwell thresholds below 300 ms during early design testing. However, even for trained users (from our research team), this led to frequent unintentional non-target selections when looking for and attempting to select the next target key. Thus, inspired by text entry systems that rely on word predictions [2, 9, 16, 21, 31, 39, 41, 52, 68, 90] we explored a different avenue: reducing the dwell threshold of only a few, clearly highlighted keys that are most likely to be the next target.

The MTD keyboard primarily involves two different static dwell thresholds — 300 ms and 200 ms. The first letter of any word must be selected by dwelling on the corresponding key for 300 ms. Then, unlike Dasher [90] and SliceType [9] that both allow only the selection of the next probable characters, our system highlights [52, 61] and enlarges [26] (at most) the 3 most probable letters, while still keeping all the other keys the same size, color and, more importantly, "selectable" as well. Similar to related work [51], these 3 letters are the corresponding next characters of the most probable words predicted by a word prediction algorithm. For example, if the user has typed "C", the next probable words are (in order of frequency) "C**O**ULD, C**A**N, C**A**ME, C**O**ME, C**A**LLED, C**O**UNTRY, C**O**URSE, C**A**NNOT, C**E**RTAIN, ...". Thus, the 3 unique predicted letters, and therefore, highlighted and enlarged keys, are 'O', 'A', and 'E'. Only these 3 letters can then be selected within just 200 ms. For all other letters (except space and backspace), the user has to dwell for 300 ms.

MTD shares the design features of DTD. For consecutive selections of the same key, all characters (except backspace) again use an increased dwell threshold of 500 ms (Figure 1b). We also consistently use a dwell threshold of 450 ms for the backspace key. However, to enable faster selection of the spacebar, the most frequently typed key [49, 82], we reduced its dwell threshold for MTD to 100 ms but set it to 500 ms for *repeated* space selections. In short, if the predicted letters in MTD are *always incorrect*, e.g., while typing passwords, other than the spacebar and distractions from highlighting keys, MTD is (in essence) the same as DTD. Thus, DTD and its performance can be regarded as a baseline for MTD.

For the word predictions, we used Python's Fast Autocomplete 0.9.0 library[1], which relies on a Directed Word Graph (DWG) and Levenshtein distance. We populated this DWG with the 40,000 most frequent words from project Gutenberg[2]. Since our keyboards and the experimental task were implemented using Unity, we set up a UDP connection to send the word that is being typed to and receive the predictions from Python.

To clearly show the user which (up to) three keys have a shorter dwell threshold of 200 ms, we initially experimented with just highlighting the corresponding keys [26, 52, 61]. However, participants were unable to identify the predicted keys sufficiently easily during early pilot testing. The reason was that whenever the gaze cursor hovered over a key, the highlight disappeared, as hovering turned the key blue and also simultaneously started the dwell timer animation (Figure 1). Thus, we decided to also enlarge the predicted keys by 33% to 4°, i.e., 1° larger than the other keys. Besides making the

[1]https://pypi.org/project/fast-autocomplete
[2]https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/English/Project_Gutenberg

multiple dwell thresholds easier to see and more predictable to the user, highlighting and enlarging the most likely keys potentially also addresses the challenges of PT and DOs, and, therefore DOT.

In small pilots of the study design detailed in the next section, we experimented with highlighting and enlarging 1, 3, and 5 keys with 100 ms, 150 ms, 200 ms, and 250 ms dwell thresholds, respectively. We found that all novices struggled with too many unintentional selections below a dwell threshold of 200 ms, making typing impractical. Similarly, when the number of predicted letters was set to 5, some of the other short-dwell-time keys got too often in the way of the user's gaze when moving from one key to another. This resulted in too many incorrect selections of non-desired but still predicted keys. In contrast, when only a single prediction was offered to the user, this letter did not turn out to be the desired one sufficiently often. Thus, to provide enough realistic options and, at the same time, enable faster typing for novice users, we chose to present (at most) 3 predicted letters that can be selected by dwelling on them for 200 ms.

5 User Study 2 – Evaluating Dual-Threshold and Multi-Threshold Dwell Keyboards

We evaluated the performance of DTD and MTD against the conventional Constant-Threshold Dwell (CTD) baseline, using a 3×5 within-subjects design with the two independent variables being **Keyboard** (CTD, DTD, and MTD) and **Block** (1-5), where each block comprises typing 5 phrases. Our dependent variables were WPM, KSPC, and MSD ER. Furthermore, we compared the different time components of dwell for the different keyboards.

5.1 Procedure

After giving informed consent, participants provided demographic information, including QWERTY experience, age, and gender. After a demonstration of the keyboards, the study task, and eye-tracking calibration, participants typed with each of the three keyboards (CTD, DTD, and MTD) in a counterbalanced order. Similar to related work [31, 39], participants entered 6 blocks of 5 phrases each with each keyboard, with short breaks between blocks. The first block (i.e., the 0^{th} block) was considered as practice and was discarded before analysis [31, 39, 94].

For all other aspects, we followed the same procedure as Study 1: using phrases from MacKenzie and Soukoreff's corpus [48], memorizing the target phrase and spelling of the words, focusing on correcting errors only within the last word, typing an extra "space" at the end of every phrase, and encouraging participants to beat their text entry speed on every new phrase.

Afterward, participants completed a post-questionnaire where they were asked about their preferences and for feedback on aspects such as ease of interaction, mental and physical fatigue, frustration, perceived precision, and perceived speed for each of the three keyboard layouts using 7-point Likert scales. Each keyboard condition lasted about 10-15 minutes and the entire experiment took about an hour.

5.2 Hypotheses

A 300 ms dwell threshold for text entry is known to be impractical for novices [21, 44, 53, 59, 67, 72, 73]. In Study 1, we identified that one reason for this is the (too) many inadvertent repeated selections caused by the naturally higher ET of novices. Thus, effectively reducing the ET requires substantial training, e.g., [21, 53, 72]. Still, if (an adapted version of a) shorter dwell threshold of 300 ms *were practical and usable*, i.e., did not create too many inadvertent selections, such a system should perform better than those with a higher dwell threshold, e.g., 450 ms as in CTD.

Given our design of an increased dwell threshold for repeated selections of the same key in DTD and MTD, we hypothesize that *a reduction of the potential of such repeated selections not only improves the text entry performance but also the usability of 300 ms dwell threshold text entry systems without requiring extensive participant training*. Given the judicious use of even lower dwell thresholds in MTD, we expect the *reductions in the dwell threshold further improve text entry performance.*

Given that the benefits of a system with less potential for inadvertent repeated selections should be visible with little training, we designed a single-day experiment similar to many previous studies [22, 25, 31, 38, 39, 43, 44, 57, 58, 61, 93]. Since we designed the new keyboards specifically for novices, we evaluated them with a representative group of participants, i.e., a group that is not as extensively trained as in a longitudinal study.

5.3 Apparatus

We conducted the study on a computer with a 13th Gen Intel(R) Core(TM) i9-13900KF processor running at 3.0 GHz, 32 GB RAM, and an NVIDIA GeForce RTX 4070 graphics card. We again used Unity, the Meta Quest Pro VR headset, and its built-in eye-tracking calibration method.

5.4 Participants

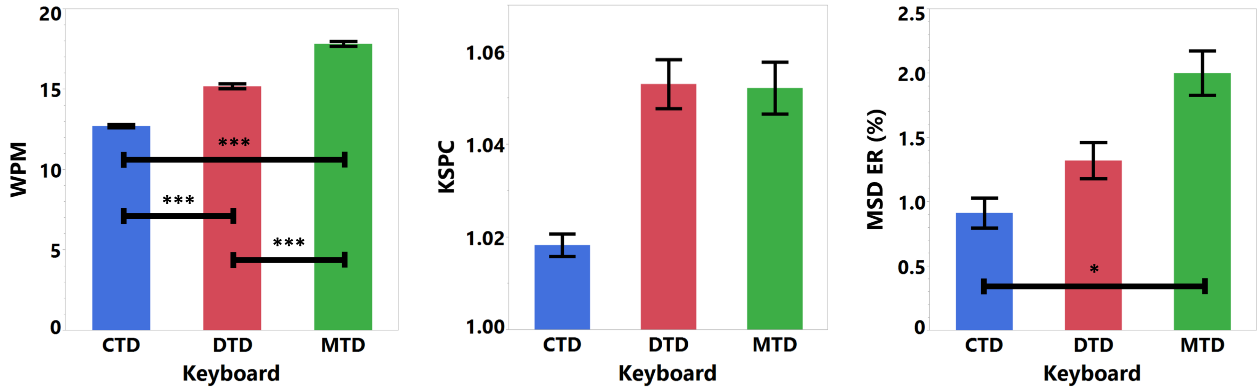
15 novice participants (10 males, 5 females), aged 25 ± 2 years, took part in this study. They all had no visual impairments or corrected-to-normal vision. 11 participants had over nine years of experience typing on a QWERTY keyboard and the other four had 7-9 years. 8 participants had never experienced eye-tracking systems while the others had experienced it at most 1-2 times before the experiment. However, this was the first time all participants used dwell-based keyboards. The participants were paid the equivalent of US $15 in the local currency for their time.

5.5 Results

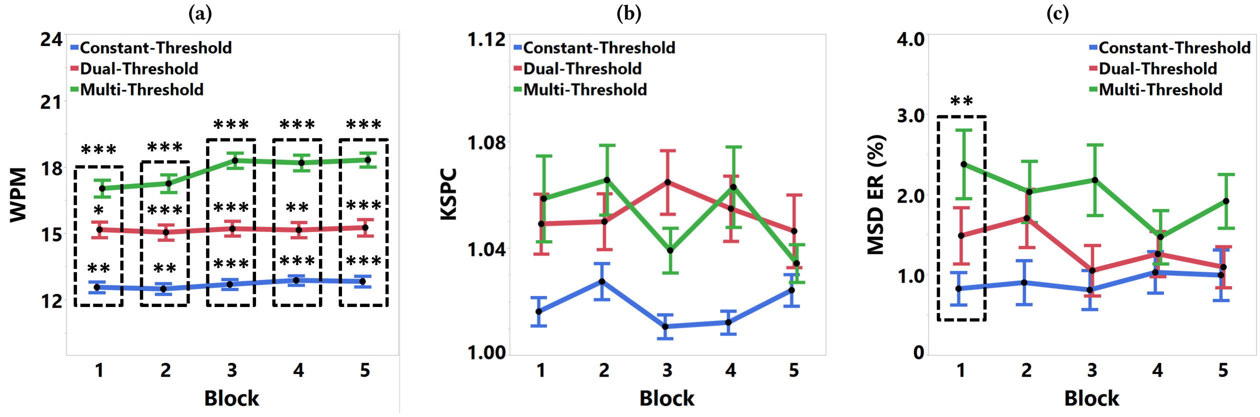
As in Study 1, we analyzed the quantitative data using RM ANOVAs with post-hoc tests. The results are presented in Tables 2-3, and Figures 6-8. For brevity, we focus on the statistically significant results.

5.5.1 WPM, KSPC, and MSD ER. While there is substantial variability in the literature, text entry studies typically report either the average results for all the blocks, one block at a time (to show the learnability of the system), or both (Table 4). To support comparability with other work, we report here both the overall average and for each block separately.

Participants' average typing speed across all blocks was 12.7 ± 2.02 WPM for CTD, significantly slower than DTD (15.2 ± 3.00 WPM) and MTD (17.8 ± 3.12 WPM). DTD was also significantly slower than MTD (Figure 6a).



(a) (b) (c)



(d) (e) (f)

Figure 6: Study 2 results for average WPM, KSPC, and MSD ER across all blocks (a-c) and for each block separately (d-f) for the Constant- (CTD), Dual- (DTD), and Multi-Threshold Dwell (MTD) keyboards. Significance levels are shown as * for $p < .001$, ** for $p < .01$, and * for $p < .05$. In (d) and (f) the asterisks above the blue line represent the difference between CTD and DTD, the red line between DTD and MTD, and the green line between CTD and MTD keyboards. The error bars show the standard error of means.**

For CTD, participants' text entry speed started at 12.6 ± 2.08 WPM in block 1 and ended at 12.8 ± 2.11 WPM on the 5th block (fastest: 12.9 ± 1.85 WPM in block 4). For DTD, participants consistently typed at about 15 WPM throughout the five blocks: 15.2 ± 2.99 WPM in block 1 and the fastest speed of 15.3 ± 3.23 WPM in block 5. Finally, with MTD, participants started at 17.0 ± 3.26 WPM, quickly improved to 18.3 ± 2.89 WPM in block 3, and ended at the fastest speed of 18.3 ± 2.71 WPM in the final block. DTD and MTD consistently exhibited significantly faster text entry speeds than the baseline CTD in every block. Similarly, MTD significantly outperformed DTD in every block (Figure 6d). The fastest a participant was able to type in a single block was 15.5 ± 0.32 WPM in block 4, 18.6 ± 0.74 WPM in block 5, and 21.8 ± 1.45 WPM in block 3, for CTD, DTD, and MTD, respectively.

Table 2: RM ANOVA results for the three dwell Keyboards and Blocks.

	Keyboard	Block	Keyboard × Block
WPM	$F_{2,28} = 71.9$ $p < .001$ $\eta^2 = 0.837$	$F_{4,56} = 4.58$ $p < .01$ $\eta^2 = 0.247$	$F_{5.1,71.9} = 1.34$ *n.s.* $\eta^2 = 0.088$
MSD ER	$F_{2,28} = 6.59$ $p < .01$ $\eta^2 = 0.320$	$F_{4,56} = 1.29$ *n.s.* $\eta^2 = 0.085$	$F_{7.02,98.3} = 0.75$ *n.s.* $\eta^2 = 0.051$

The average KSPC over all blocks was 1.018 ± 0.047, 1.053 ± 0.102, and 1.052 ± 0.107 for CTD, DTD, and MTD, respectively (Figure 6b). In the first and last block, participants' average KSPC were 1.016 ± 0.046 and 1.024 ± 0.051 for CTD, 1.049 ± 0.096 and 1.046 ± 0.118 for DTD, and 1.059 ± 0.138 and 1.034 ± 0.060 for MTD, respectively (Figure 6e). Yet, two-way RM ANOVA did not reveal significant differences for Keyboard, Block, or their interaction for KSPC.

The average MSD ER across all blocks was 0.91 ± 2.24 for CTD, 1.32 ± 2.72 for DTD, and 2.00 ± 3.31 for MTD. Participants' average MSD ERs were 0.82 ± 1.75 and 0.99 ± 2.72 for CTD, 1.49 ± 2.98 and 1.09 ± 2.22 for DTD, and 2.38 ± 3.67 and 1.92 ± 2.86 for MTD, in the 1st and 5th block, respectively (Figure 6f). Post-hoc analysis revealed significant differences between CTD and MTD for the average MSD ER across all blocks (Figure 6c). However, further analysis revealed that only the differences for block 1 are different between CTD and MTD (Figure 6f).

Table 3: Study 2 results for average Exit Time (ET), Pointing Time (PT), On Target Time (OTT), Drop-off Time (DOT), Activation Time (AT), Total Time (TT), and Drop-offs (DOs) for each of the three dwell keyboards.

	Exit Time (ET)		Pointing Time (PT)		On Target Time (OTT)		Drop-off Time (DOT)		Activation Time (AT)		Total Time (TT)	Drop-offs (DOs)
Keyboard	Time (s)	% of TT	Time (s)	% of TT	Time (s)	% of TT	Time (s)	% of TT	Time (s)	% of TT	Time (s)	#
CTD	0.235 ± 0.09	24.8%	0.129 ± 0.197	13.6%	0.516 ± 0.164	54.3%	0.069 ± 0.240	7.28%	0.585 ± 0.381	61.6%	0.949 ± 0.456	0.428 ± 1.209
DTD	0.236 ± 0.10	30.0%	0.135 ± 0.238	17.1%	0.356 ± 0.122	45.2%	0.060 ± 0.232	7.63%	0.416 ± 0.325	52.8%	0.787 ± 0.435	0.355 ± 1.055
MTD	0.237 ± 0.11	35.8%	0.130 ± 0.215	19.7%	0.255 ± 0.133	38.5%	0.040 ± 0.165	6.05%	0.295 ± 0.251	44.5%	0.663 ± 0.338	0.284 ± 0.799
RM ANOVA	$F_{2,28} = 0.13$ *n.s.* $\eta^2 = 0.009$		$F_{2,28} = 0.60$ *n.s.* $\eta^2 = 0.041$		$\boldsymbol{F_{2,28} = 1088.6}$ $\boldsymbol{p < .001}$ $\boldsymbol{\eta^2 = 0.987}$		$\boldsymbol{F_{2,28} = 6.55}$ $\boldsymbol{p < .01}$ $\boldsymbol{\eta^2 = 0.319}$		$\boldsymbol{F_{2,28} = 247.3}$ $\boldsymbol{p < .001}$ $\boldsymbol{\eta^2 = 0.946}$		$\boldsymbol{F_{2,28} = 81.0}$ $\boldsymbol{p < .001}$ $\boldsymbol{\eta^2 = 0.853}$	$\boldsymbol{F_{2,28} = 6.06}$ $\boldsymbol{p < .01}$ $\boldsymbol{\eta^2 = 0.302}$

5.5.2 Components of Dwell Selection. The results for the different components of dwell selection for each of the three keyboards are presented in Table 3. MTD exhibited significantly fewer DOs than CTD, resulting in MTD being significantly faster than CTD in terms of DOT. For OTT and therefore also AT and TT, all three keyboards were significantly different from each other (Figure 7). All other differences were not significant.

5.5.3 Effective Dwell Threshold. We also *computed* the *effective* dwell threshold by averaging the dwell thresholds that the system used across all entered keys over all blocks. This resulted in an effective average dwell threshold of 454.7 ± 4.13 ms for CTD, 313.4 ± 39.3 ms for DTD, and 233.9 ± 89.6 ms for MTD.

5.5.4 Subjective Results. Most (9 of 15) participants preferred MTD, 4 participants DTD, and just 2 participants CTD. Typical comments of participants who chose MTD as their preferred keyboard were "*Highlighted letters helped me type faster*", "*Suggested letters helps me type faster although correcting my mistakes was a painful task*", "*Easier to type with it...*", and "*It was very quick ... and reduced the*

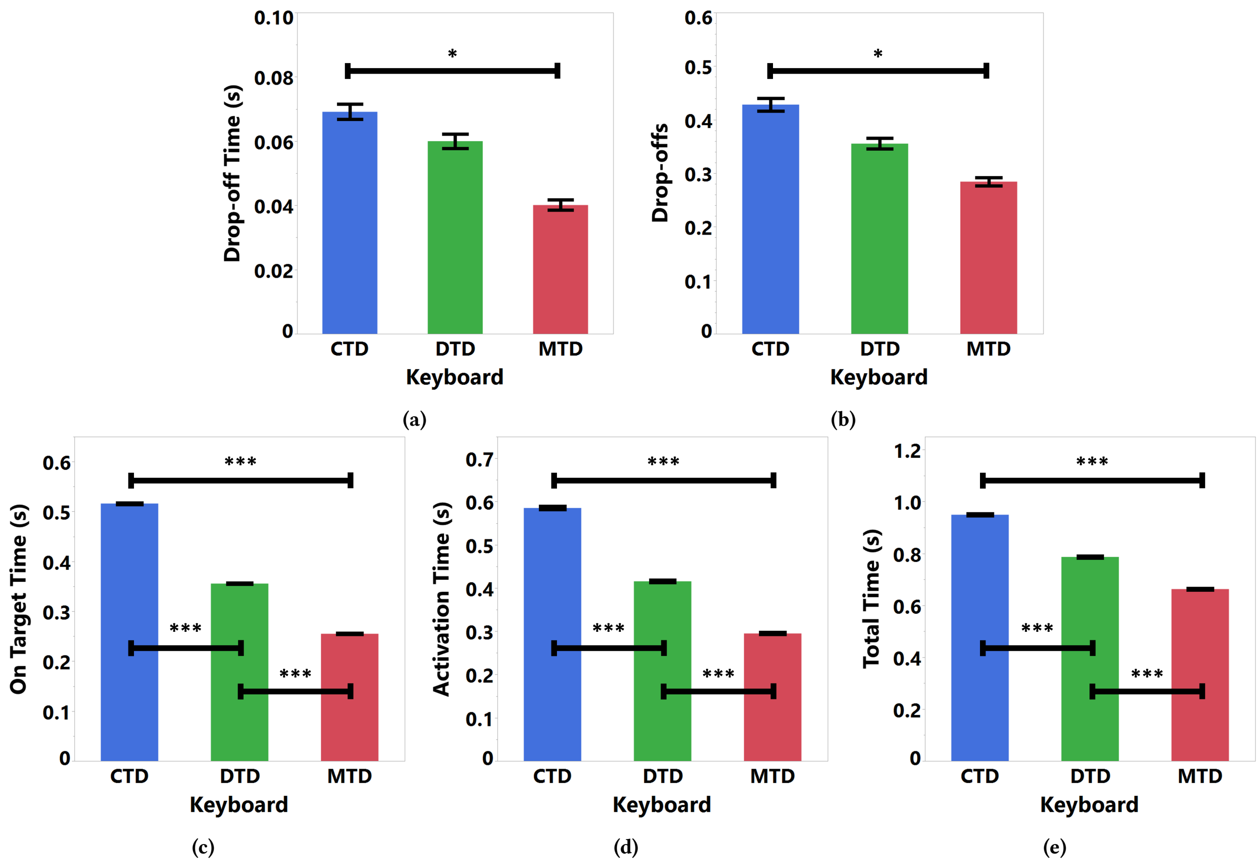


Figure 7: Study 2 results for (a) Drop-off Time (DOT), (b) Drop-offs (DOs), (c) On Target Time (OTT), (d) Activation Time (AT), and (e) Total Time (TT) for Constant- (CTD), Dual- (DTD), and Multi-Threshold Dwell (MTD) keyboards. Significance levels are shown as * for $p < .001$, ** for $p < .01$, and * for $p < .05$. The error bars show the standard error of means.**

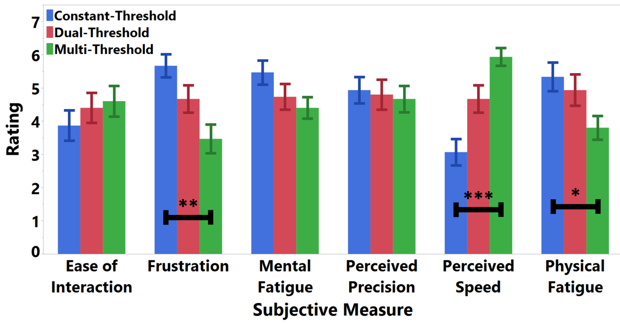


Figure 8: Subjective results for the three keyboards. Significance levels are shown as * for $p < .001$, ** for $p < .01$, and * for $p < .05$. The error bars show the standard error of means.**

cognitive load." Representative comments for the choice of DTD were "*I felt like with short selection time it was easier and smooth experience, I did not like [CTD] as it was tiring and boring and also [MTD] made me confused*", "*same speed helps me focus and make [it] a habit...*', and "*I think I performed better [with DTD]. For [MTD], it was confusing sometimes because of the big white squares. I think I mistakenly entered some letters when I was just passing by.*" Participants who chose CTD stated that "*Since it takes time for the selection, eye movement between the letters won't be captured fast and can easily point to the correct letter and avoid mistakes*" and "*[CTD] is less prone to mistakes.*"

When asked about the ease of interaction, frustration, mental and physical fatigue, and perceived speed and precision on a 7-point Likert scale, with 7 signifying very easy, frustrating, fatiguing, fast, and precise, respectively, participants rated MTD to be better on average than CTD and DTD for all subjective measures except perceived precision (Figure 8). However, only the differences for frustration ($F_{2,28}$ = 7.77, $p < .01$, $\eta^2 = 0.357$), perceived speed ($F_{2,28}$ = 13.0, $p < .001$, $\eta^2 = 0.481$), and physical fatigue ($F_{2,28}$ = 4.26, $p < .05$, $\eta^2 = 0.233$) were significant. Post-hoc analysis showed only MTD to be significantly better than CTD in terms of frustration, perceived speed, and physical fatigue, but no significant differences between CTD and DTD, nor DTD and MTD.

5.6 Study 2 Discussion

In Study 2, we evaluated two novel dwell keyboards — Dual-Threshold Dwell (DTD) and Multi-Threshold Dwell (MTD) — against the conventional Constant-Threshold Dwell (CTD) keyboard baseline with novice users. MTD (18.3 WPM) was significantly faster than DTD (15.3 WPM), which in turn was significantly faster than CTD (12.9 WPM). Although participants made significantly more errors with MTD over CTD in terms of the overall average MSD ER, further analysis suggested that, as the difference was only noteworthy in the first block (Figure 6f), participants were still getting used to MTD. The MSD ER for all keyboards was well below 3% throughout the five blocks. This level of mistakes is typically deemed (somewhat) acceptable for gaze-based text entry systems [31].

The computed effective average dwell threshold of 233.9 ms for MTD is *the shortest dwell threshold ever published* for a dwell-based text entry system [21, 53, 59, 72]. Moreover, according to Table 3, the OTT, i.e., the actual average (measured) time participants dwelled on the selected keys, was 516 ms for CTD (~60 ms longer than the 454.7 ms effective average dwell threshold), 356 ms for DTD (~40 ms longer than the 313.4 ms threshold), and only 255 ms for MTD (~20 ms longer than the threshold of 233.9 ms). The reduced dwell threshold of MTD also resulted in fewer DOs than CTD which in turn significantly improved MTD's DOT (Figures 7a and 7b). These results suggest that our MTD design substantially reduced both the OTT and DO challenges identified in the first study. The short dwell thresholds for both DTD and MTD also did *not* result in significantly higher KSPC compared to CTD. We found 1.046 KSPC for DTD and 1.034 KSPC for MTD, both of which are substantially lower than the previously reported KSPC of 1.18 (or more) for text entry systems that use a dwell threshold of ~300 ms [21, 53, 72].

Most importantly, our participant pool comprised only novice eye-tracking users and they typed just 30 phrases (10-15 minutes) on each keyboard. Thus, our results provide evidence that the reason why a 300 ms dwell threshold was deemed suitable only for experienced users [60] seems to be mainly due to the naturally longer ET of novice users (Table 1). In other words, our results suggest that the *Exit Time seems to be the key factor why text entry systems that utilize a 300 ms dwell threshold were deemed unusable for novices.* Without the suggested DTD mechanism, such short dwell thresholds often result in unintentional double clicks of the same key, and in the process, require more error corrections [21, 53, 72]. DTD thus makes a dwell threshold of 300 ms (or lower) substantially more usable, especially for novices.

The further improvement of dwell-based text entry system usability with MTD resulted in very high text entry speeds, enabling *novices* to approach the 20 WPM limit. Thus, the results support our hypotheses that reducing inadvertent repeated selections improves the performance and usability of 300 ms dwell threshold text entry systems and also requires little training. Moreover, as exhibited in Table 4, *MTD is the fastest gaze-based keyboard in terms of average text entry speed, and its text entry speed in the last block seems to even be competitive with multimodal approaches.* Thus, we believe that if participants were trained as extensively as in other work [53, 72], MTD might even surpass the 20 WPM mark. However, further work is required to verify this speculation.

Although most participants (9 of 15) preferred MTD, three participants noted that MTD was confusing. The most common reason given was situations where two neighboring keys were predicted by the prediction algorithm. For example, if the user had typed "EXP", the predicted words were "EXP**R**ESSION, EXP**E**CTED, ..., EXP**L**AINED, ...", and thus the next predicted letters were 'R', 'E', and 'L'. Attempting to type 'E', e.g., as in "EXPENSIVE", could then result in an accidental selection of 'R'. Yet, since the dwell threshold was so low, it was sometimes hard to tell which letter got selected other than by looking at the typed text, which we discouraged. Such instances of confusion are a probable reason why participants' Pointing Time (PT) was not significantly faster for MTD, i.e., was similar to the other two keyboards, thus (somewhat) contradicting our expectation that highlighting/enlarging the predicted letters would improve PT. As the dwell threshold for some keys was so low, participants were likely slightly more careful with MTD and took their time to point more accurately to the next key. Adding autocorrect features [2, 68] and/or using Mott et al.'s probabilistic

Table 4: Summary of text entry speeds and study characteristics reported in the literature. This table is an extended version of the one shown in previous work [31]. Some entries are estimated based on figures and data from the original papers, indicated by the '∼' symbol. Unavailable entries are marked with the '–' symbol. Certain papers report only the mean performance from the last session(s)/block(s), with the session/block count noted in parentheses. A few studies provide WPM solely from the last session/block, which is listed in the "Last WPM" column. Some works report the maximum WPM of a single participant or the fastest session/block averaged over all participants. We listed whichever is higher and clarified in parentheses. The practice time reflects the approximate effort participants needed to reach the entry rate observed in the last session/block.

Authors	Method	Multimodal	Average WPM	Last WPM	Maximum WPM	Practice Time
Tuisku et al. [86]	Dasher	No	–	17.26	∼23.11 (participant)	150 min
Rough et al. [76]	Dasher with adjustable dwell threshold	No	–	14.2	∼19.5 (participant)	∼7.5 hours
Mott et al. [59]	Cascading dwell time	No	12.39	∼12.5	13.7 (session/block)	20 phrases + ∼150 min
Urbina & Huckauf [87]	pEYEWrite with bigrams and word prediction	No	13.47 (last 3 sessions/blocks)	–	17.26 (participant)	15 min + ∼150 phrases
Majaranta et al. [53]	Adjustable dwell threshold	No	∼15	19.9	∼23 (participant)	10 days (∼150 min)
Räihä & Ovaska [72]	Adjustable dwell threshold	No	–	∼19	∼24 (participant)	150 min + 6 days (135 min)
Diaz-Tula & Morimoto [21]	AugKey	No	16.72 (last 3 sessions/blocks)	∼17	∼17 (session/block)	72+ min
Morimoto & Amir [57]	Context switching	No	12	∼13	∼22 (participant)	5 min + 70 min
Morimoto et al. [58]	Context switching with dynamic targets	No	13.1	13.42	∼14 (session/block)	5 min + ∼35 phrases
Pedrosa et al. [68]	Filteryedping	No	14.75	15.95	19.28 (participant)	100 min
Kurauchi et al. [41]	EyeSwipe	No	10.68	11.7	20.6 (author)	2 phrases + 30 min
Kurauchi et al. [42]	Swipe&Switch	No	13.74	–	33 (participant)	80 min
Kumar et al. [39]	TAGSwipe	Yes, Gaze + Touch	15.46	∼16	20.5 (participant)	5 + 25 phrases
Hedeshy et al. [31]	Hummer	Yes, Gaze + Humming	15.48	20.45	30.64 (mean of users' max.)	5 + 25 phrases
Our work	Multi-Threshold Dwell	No	17.8	18.3	21.8 (participant)	5 + 25 phrases

solution for adjacent keys [59] in the future could potentially mitigate some of the confusion by making such errors less likely to occur.

Unsurprisingly, the differences in dwell threshold between the keyboards resulted in all keyboards being significantly different from each other in terms of On Target Time (OTT), and therefore Activation Time (AT) and Total Time (TT). Although the number of Drop-offs (DOs) and thus the Drop-off Time (DOT) decreased for DTD, DTD was not significantly different from either of the other two keyboards. This suggests that small differences in the effective dwell thresholds, in this case, ∼140 ms with CTD and ∼80 ms with MTD, do not significantly affect DOs and DOT. Yet, this also requires further investigation.

6 General Discussion, Limitations, and Future Work

In this work, we empirically investigated the factors that contribute to the apparent limit of 20 WPM for dwell-based text entry systems with eye gaze. Toward this goal, we analyzed the different components of the dwell selection time in a longitudinal study. We identified design opportunities around challenges arising from Exit Time (ET), Pointing Time (PT), On Target Time (OTT), and

Drop-offs (DOs). Guided by this, we designed two novel keyboards with lower dwell thresholds — *Dual-Threshold Dwell* (DTD) and *Multi-Threshold Dwell* (MTD) (Figure 1). These new keyboards improve the performance of the conventional Constant-Threshold Dwell (CTD) keyboard by judiciously introducing new types of well-defined and predictable dwell thresholds, targeting in particular the challenges encountered by novices. Tested with novice users and within just 30 phrases of practice, MTD (18.3 WPM) outperformed DTD (15.3 WPM), which in turn outperformed CTD (12.9 WPM). Despite our participants being novices and having received comparatively little training, MTD's demonstrated text entry speed is the fastest gaze-based keyboard reported to date in terms of its average text entry speed, with a last text entry speed that is competitive even with multimodal approaches (Table 4).

Although participants' Pointing Time (PT) in MTD (0.130 s) was not significantly faster than CTD (0.129 s) and DTD (0.135 s) in Study 2, MTD's PT is somewhat similar to participants' PT with CTD in the early sessions, i.e., 2-4, of Study 1 (~0.130 s). Moreover, Exit Time (ET) was similar across all keyboards in Study 2, which in turn is similar to the first three sessions in Study 1 (i.e., ~0.230 s). Thus, to estimate MTD's potential long-term performance, we applied the learning curves for ET and PT (Equations 3 and 4) to extrapolate our results. Using the observed On Target Time (OTT; 0.255 s) and Drop-off Time (DOT; 0.040 s) for MTD in Study 2, and assuming thirteen 15-minute training sessions (i.e., the 15-minute sessions employed by previous work [53, 72, 76, 86] and Study 1), this resulted in a predicted average Total Time (TT) of 0.550 s and typing speed of ~22 WPM, which indicates that this approach might even exceed the 20 WPM limit. Note that our extrapolation assumes that OTT and DOT for MTD do not improve, i.e., it likely still underestimates the maximum typing speed. Additionally, this does not account for potential further dwell threshold reductions that likely become feasible with increased training.

Thus, instead of training participants on the 300/200 ms dwell thresholds in MTD to beat the current limit of 20 WPM, we encourage future work to investigate whether the two dwell thresholds for normal/predicted keys can be further reduced through training while appropriately accounting for ET, i.e., ≥ 200 ms for novices and similarly, ≥ 150 ms for trained users (Figure 4a and Tables 1 and 3). Other than normal/predicted key dwell thresholds and ET, there is also much scope for improvement of the other aspects of MTD. This includes but is not limited to the number of letters predicted, whether to highlight/enlarge the predicted letters and by how much they should be highlighted/enlarged, whether to shorten the dwell time threshold for two adjacent keys even though both are in the predicted set, the dwell threshold for the space and backspace keys, a better word prediction algorithm and/or corpus, a better layout than QWERTY, e.g., OPTI II [75], and different keyboard dimensions including key sizes and the gaps between them.

One feature of MTD that limits its performance is that the first character of every word always requires a dwell threshold of 300 ms. This limitation could be addressed by incorporating word- or phrase-level predictions instead of just letter predictions, e.g., based on the recent advances in large language models (LLMs) [36]. Further, LLMs alongside autocorrection [2, 68] could also reduce the delay for intentional consecutive same-character selection, e.g., by completely disabling the potential for double clicks in some situations and relying on autocorrect to add the second character.

In short, while our MTD parameters for Study 2 were guided by our design process, we acknowledge that they may not yet be optimal. Yet, identifying optimal settings was beyond the scope of this work. Our primary goal for Study 2 was to evaluate specific design ideas that were informed by Study 1, i.e., by insights into the different time components of dwell selection, which potentially have far-reaching implications beyond just dwell-based (text entry) systems.

Thus, another exemplar application of our results could be gaze-based swipe keyboards [16, 31, 39, 41]. Swipe-based typing also relies on some implicit dwelling, in the form of saccade latencies (i.e., the natural wait time in between two saccades [23]), over (or near the vicinity of) each key. Our results could thus contribute to improving the gaze gesture recognition algorithm, a critical component for such keyboards. Moreover, while swipe typing has been demonstrated to be better than tap typing for touchscreen devices [75], the same may not be true for gaze-based typing. In touch-based tap typing, the physical movement of raising a finger and bringing it back down every time to select a key is a potential reason why swipe typing achieves faster speeds, as the up-and-down finger movement in swipe typing is required only once every word. However, no such physical movement is involved with eye gaze. In essence, the difference between dwell typing and swipe typing with gaze is how long the user has to dwell over each key. We hope that our findings, specifically the need to account for the ET, will encourage the investigation of a combination of dwell- and swipe-based text entry systems with gaze. Yet, we reiterate that all this is speculation, and the ideas presented in this paragraph need to be verified in future work. Still, beyond our comparison presented in Table 4, it would be interesting to compare MTD with swipe keyboards, too.

Although previous work [73] found VR to be a viable option for gaze-based text entry, the vergence-accommodation conflict [6] could still pose a potential limitation of such systems if the gaze has to (repeatedly) jump between different visual depths. Another possible shortcoming is the limited number of participants in our two studies. Still, the significant differences exhibited large effect sizes ($\eta^2 > 0.14$) in both studies. More importantly, our number of participants is consistent with much other work [11, 37] indicating robust findings and suggesting that our results are likely replicable. Finally, we acknowledge that, like most text entry studies [22], our results are also limited to the transcription typing task, which is not always representative of real-world typing. Moreover, we discouraged participants from going back and forth to verify the typed text while typing, which is again different from actual typing. We encourage future work to investigate these limitations.

7 Conclusion

In this paper, we examined which factors limit the speed of dwell-based text entry systems, analyzing the time components of dwell selection in a longitudinal study. Driven by an increasing dominance of the dwell threshold as users become more accustomed to a system, our results highlight challenges and design opportunities around Exit Time, Pointing Time, On Target Time, and Drop-offs.

Guided by these observations, we designed two novel keyboards — Dual-Threshold Dwell (DTD) and Multi-Threshold Dwell (MTD), which feature multiple and substantially shorter dwell thresholds compared to a typical Constant-Threshold Dwell (CTD) keyboard (Figure 1). With a speed of 18.3 WPM, MTD substantially outperformed DTD and CTD, making it the fastest gaze-based keyboard in terms of its average text entry speed, with a last text entry speed competitive even with multimodal approaches. Remarkably, this performance was achieved by novice users within only 30 phrases. These results highlight the benefits of using multiple judiciously chosen dwell thresholds. This approach opens up the way for further optimizations addressing the performance impacts of the different components of dwell selection time.

References

[1] Jiban Adhikary, Max Isom, and Keith Vertanen. 2022. The Impact of Number of Predictions on User Performance in a Dwell Keyboard. In *MobileHCI 2022 Workshop on Shaping Text Entry Research for 2030*. MobileHCI 2022 Workshop on Shaping Text Entry Research for 2030, Vancouver, BC.

[2] Ohoud Alharbi, Wolfgang Stuerzlinger, and Felix Putze. 2020. The Effects of Predictive Features of Mobile Keyboards on Text Entry Speed and Errors. *Proceedings of the ACM Human-Computer Interaction* 4, ISS, Article 183 (Nov 2020), 16 pages. https://doi.org/10.1145/3427311 ISS '20.

[3] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An integrated theory of the mind. *Psychological Review* 111, 4 (2004), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036

[4] A. S. Arif and W. Stuerzlinger. 2009. Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. IEEE, New York, USA, 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

[5] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2010. Predicting the cost of error correction in character-based text entry technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 5–14. https://doi.org/10.1145/1753326.1753329

[6] Anil Ufuk Batmaz, Mayra Donaji Barrera Machuca, Junwei Sun, and Wolfgang Stuerzlinger. 2022. The Effect of the Vergence-Accommodation Conflict on Virtual Hand Pointing in Immersive Displays. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 633, 15 pages. https://doi.org/10.1145/3491102.3502067

[7] Stefanie I. Becker. 2011. Determinants of Dwell Time in Visual Search: Similarity or Perceptual Difficulty? *PLOS ONE* 6, 3 (03 2011), 1–5. https://doi.org/10.1371/journal.pone.0017740

[8] Nikolaus Bee and Elisabeth André. 2008. Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze. In *Perception in Multimodal Dialogue Systems*, Elisabeth André, Laila Dybkjær, Wolfgang Minker, Heiko Neumann, Roberto Pieraccini, and Michael Weber (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 111–122.

[9] Burak Benligiray, Cihan Topal, and Cuneyt Akinlar. 2019. Slicetype: fast gaze typing with a merging keyboard. *Journal on Multimodal User Interfaces* 13, 4 (2019), 321–334.

[10] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of Eye-Gaze over Head-Gaze-Based Selection in Virtual and Augmented Reality under Varying Field of Views. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 9 pages. https://doi.org/10.1145/3206343.3206349

[11] Kelly Caine. 2016. Local Standards for Sample Size at CHI. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 981–992. https://doi.org/10.1145/2858036.2858498

[12] Stuart K. Card, Thomas P. Moran, and Allen Newell. 1980. The Keystroke-Level Model for User Performance Time with Interactive Systems. *Commun. ACM* 23, 7 (jul 1980), 396–410. https://doi.org/10.1145/358886.358895

[13] Steven J. Castellucci, I. Scott MacKenzie, Mudit Misra, Laxmi Pandey, and Ahmed Sabbir Arif. 2019. TiltWriter: Design and Evaluation of a No-Touch Tilt-Based Text Entry Method for Handheld Devices. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia* (Pisa, Italy) *(MUM '19)*. Association for Computing Machinery, New York, NY, USA, Article 7, 8 pages. https://doi.org/10.1145/3365610.3365629

[14] Edward Clarkson, James Clawson, Kent Lyons, and Thad Starner. 2005. An Empirical Study of Typing Rates on Mini-QWERTY Keyboards. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA) *(CHI EA '05)*. Association for Computing Machinery, New York, NY, USA, 1288–1291. https://doi.org/10.1145/1056808.1056898

[15] Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. Routledge, NJ, USA.

[16] Wenzhe Cui, Rui Liu, Zhi Li, Yifan Wang, Andrew Wang, Xia Zhao, Sina Rashidian, Furqan Baig, IV Ramakrishnan, Fusheng Wang, and Xiaojun Bi. 2023. GlanceWriter: Writing Text by Glancing Over Letters with Gaze. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 719, 13 pages. https://doi.org/10.1145/3544548.3581269

[17] Arindam Das and Wolfgang Stuerzlinger. 2007. A cognitive simulation model for novice text entry on cell phone keypads. In *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!* (London, United Kingdom) *(ECCE '07)*. Association for Computing Machinery, New York, NY, USA, 141–147. https://doi.org/10.1145/1362550.1362579

[18] Arindam Das and Wolfgang Stuerzlinger. 2008. Modeling learning effects in mobile texting. In *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia* (Umeå, Sweden) *(MUM '08)*. Association for Computing Machinery, New York, NY, USA, 154–161. https://doi.org/10.1145/1543137.1543169

[19] Arindam Das and Wolfgang Stuerzlinger. 2010. Proactive Interference in Location Learning: A New Closed-Form Approximation. In *International Conference on Cognitive Modeling (ICCM '10)*. D. D. Salvucci & G. Gunzelmann (Eds.), Philadelphia, PA: Drexel University, 37–42. https://iccm-conference.neocities.org/2010/proceedings

[20] Arindam Das and Wolfgang Stuerzlinger. 2013. Unified Modeling of Proactive Interference and Memorization Effort: A New Mathematical Perspective Within ACT-R Theory. In *Annual Meeting of the Cognitive Science Society (CogSci '13)*. eScholarship, University of California, Merced, CA, USA, 358–363. https://escholarship.org/uc/item/1p05s7db

[21] Antonio Diaz-Tula and Carlos H. Morimoto. 2016. AugKey: Increasing Foveal Throughput in Eye Typing with Augmented Keys. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3533–3544. https://doi.org/10.1145/2858036.2858517

[22] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text Entry in Virtual Reality: A Comprehensive Review of the Literature. In *Human-Computer Interaction. Recognition and Interaction Technologies*, Masaaki Kurosu (Ed.). Springer International Publishing, Cham, 419–437.

[23] Andrew T Duchowski. 2017. *Eye tracking methodology: Theory and practice*. Springer, New York, USA. https://books.google.ca/books?id=0Ge2tAEACAAJ

[24] Wenxin Feng, Ming Chen, and Margrit Betke. 2014. Target Reverse Crossing: A Selection Method for Camera-Based Mouse-Replacement Systems. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments* (Rhodes, Greece) *(PETRA '14)*. Association for Computing Machinery, New York, NY, USA, Article 39, 4 pages. https://doi.org/10.1145/2674396.2674443

[25] Wenxin Feng, Jiangnan Zou, Andrew Kurauchi, Carlos H Morimoto, and Margrit Betke. 2021. HGaze Typing: Head-Gesture Assisted Gaze Typing. In *ACM Symposium on Eye Tracking Research and Applications*. Association for Computing Machinery, New York, NY, USA, Article 11, 11 pages. https://doi.org/10.1145/3448017.3457379

[26] Dominik Grüneis, Marc Kurz, and Erik Sonnleitner. 2023. Let Me Help You: Improving the Novice Experience of High-Performance Keyboard Layouts with Visual Clues. *Applied Sciences* 13, 16 (2023), 16 pages. https://doi.org/10.3390/app13169391

[27] Joseph F Hair Jr, William C Black, Barry J Babin, and Rolph E. Anderson. 2014. Multivariate data analysis.

[28] John Paulin Hansen, Anders Sewerin Johansen, Dan Witzner Hansen, Kenji Ito, and Satoru Mashino. 2003. Command Without a Click: Dwell Time Typing by Mouse and Gaze Selections. In *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*, Matthias Rauterberg, Marino Menozzi, and Janet Wesson (Eds.). IOS Press, Zurich, Switzerland, 121–128.

[29] John Paulin Hansen, Vijay Rajanna, I. Scott MacKenzie, and Per Bækgaard. 2018. A Fitts' Law Study of Click and Dwell Interaction by Gaze, Head and Mouse with a Head-Mounted Display. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. Association for Computing Machinery, New York, NY, USA, Article 7, 5 pages. https://doi.org/10.1145/3206343.3206344

[30] Mehedi Hassan, John Magee, and I Scott MacKenzie. 2019. A Fitts' law evaluation of hands-free and hands-on input on a laptop computer. In *International Conference on Human-Computer Interaction*. Springer, New York, USA, 234–249.

[31] Ramin Hedeshy, Chandan Kumar, Raphael Menges, and Steffen Staab. 2021. Hummer: Text Entry by Gaze and Hum. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 741, 11 pages. https://doi.org/10.1145/3411764.

3445501

[32] Toshiya Isomoto, Toshiyuki Ando, Buntarou Shizuki, and Shin Takahashi. 2018. Dwell Time Reduction Technique Using Fitts' Law for Gaze-Based Target Acquisition. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research and Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 26, 7 pages. https://doi.org/10.1145/3204493.3204532

[33] Toshiya Isomoto, Shota Yamanaka, and Buntarou Shizuki. 2022. Dwell Selection with ML-based Intent Prediction Using Only Gaze Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3, Article 120 (sep 2022), 21 pages. https://doi.org/10.1145/3550301

[34] Robert JK Jacob. 1995. Eye tracking in advanced interface design. *Virtual environments and advanced interface design* 258, 288 (1995), 2.

[35] Jussi P. P. Jokinen, Sayan Sarcar, Antti Oulasvirta, Chaklam Silpasuwanchai, Zhenxin Wang, and Xiangshi Ren. 2017. Modelling Learning of New Keyboard Layouts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4203–4215. https://doi.org/10.1145/3025453.3025580

[36] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences* 103 (2023), 102274. https://doi.org/10.1016/j.lindif.2023.102274

[37] Maria Kjærup, Mikael B. Skov, Peter Axel Nielsen, Jesper Kjeldskov, Jens Gerken, and Harald Reiterer. 2021. *Longitudinal Studies in HCI Research: A Review of CHI Publications From 1982–2019.* Springer International Publishing, Cham, 11–39. https://doi.org/10.1007/978-3-030-67322-2_2

[38] Per Ola Kristensson and Keith Vertanen. 2012. The Potential of Dwell-Free Eye-Typing for Fast Assistive Gaze Communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 241–244. https://doi.org/10.1145/2168556.2168605

[39] Chandan Kumar, Ramin Hedeshy, I. Scott MacKenzie, and Steffen Staab. 2020. TAGSwipe: Touch Assisted Gaze Swipe for Text Entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376317

[40] Chandan Kumar, Raphael Menges, Daniel Müller, and Steffen Staab. 2017. Chromium Based Framework to Include Gaze Interaction in Web Browser. In *Proceedings of the 26th International Conference on World Wide Web Companion* (Perth, Australia) *(WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 219–223. https://doi.org/10.1145/3041021.3054730

[41] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-Free Text Entry Using Gaze Paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1952–1956. https://doi.org/10.1145/2858036.2858335

[42] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos H. Morimoto, and Margrit Betke. 2020. Swipe&Switch: Text Entry Using Gaze Paths and Context Switching. In *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 84–86. https://doi.org/10.1145/3379350.3416193

[43] Frank Chun Yat Li, Richard T. Guy, Koji Yatani, and Khai N. Truong. 2011. The 1line Keyboard: A QWERTY Layout in a Single Line. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 461–470. https://doi.org/10.1145/2047196.2047257

[44] X. Lu, D. Yu, H. N. Liang, W. Xu, Y. Chen, X. Li, and K. Hasan. 2020. Exploration of Hands-free Text Entry Techniques For Virtual Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, New York, USA, 344–349. https://doi.org/10.1109/ISMAR50242.2020.00061

[45] Christof Lutteroth, Moiz Penkar, and Gerald Weber. 2015. Gaze vs. Mouse: A Fast and Accurate Gaze-Only Click Alternative. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 385–394. https://doi.org/10.1145/2807442.2807461

[46] I Scott MacKenzie. 2012. *Human-computer interaction: An empirical research perspective.* Newnes, Boston. https://doi.org/10.1016/B978-0-12-405865-1.00007-8

[47] I. Scott MacKenzie and R. William Soukoreff. 2002. Text Entry for Mobile Computing: Models and Methods,Theory and Practice. *Human–Computer Interaction* 17, 2-3 (2002), 147–198. https://doi.org/10.1080/07370024.2002.9667313 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/07370024.2002.9667313

[48] 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* (Ft. Lauderdale, Florida, USA) *(CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971

[49] I. Scott MacKenzie and Shawn X. Zhang. 1999. The Design and Evaluation of a High-Performance Soft Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 25–31. https://doi.org/10.1145/302979.302983

[50] I. Scott Mackenzie, Shawn X. Zhang, and R. William Soukoreff. 1999. Text entry using soft keyboards. *Behaviour & Information Technology* 18, 4 (1999), 235–244. https://doi.org/10.1080/014492999118995 arXiv:https://doi.org/10.1080/014492999118995

[51] I. Scott MacKenzie and Xuang Zhang. 2008. Eye typing using word and letter prediction and a fixation algorithm. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (Savannah, Georgia) *(ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 55–58. https://doi.org/10.1145/1344471.1344484

[52] Laurent Magnien, Jean Léon Bouraoui, and Nadine Vigouroux. 2004. Mobile Text Input with Soft Keyboards: Optimization by Means of Visual Clues. In *Mobile Human-Computer Interaction - MobileHCI 2004*, Stephen Brewster and Mark Dunlop (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 337–341.

[53] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast Gaze Typing with an Adjustable Dwell Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 357–360. https://doi.org/10.1145/1518701.1518758

[54] Päivi Majaranta and Andreas Bulling. 2014. *Eye Tracking and Eye-Based Human–Computer Interaction.* Springer London, London, 39–65. https://doi.org/10.1007/978-1-4471-6392-3_3

[55] Päivi Majaranta, I Scott MacKenzie, Anne Aula, and Kari-Jouko Räihä. 2006. Effects of feedback and dwell time on eye typing speed and accuracy. *Universal Access in the Information Society* 5, 2 (2006), 199–208.

[56] Paul Mallery and Darren George. 2003. *SPSS for Windows step by step: a simple guide and reference.* Pearson, London, UK. 1–402 pages.

[57] Carlos H. Morimoto and Arnon Amir. 2010. Context Switching for Fast Key Selection in Text Entry Applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (Austin, Texas) *(ETRA '10)*. Association for Computing Machinery, New York, NY, USA, 271–274. https://doi.org/10.1145/1743666.1743730

[58] Carlos H. Morimoto, Jose A. T. Leyva, and Antonio Diaz-Tula. 2018. Context Switching Eye Typing Using Dynamic Expanding Targets. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. Association for Computing Machinery, New York, NY, USA, Article 6, 9 pages. https://doi.org/10.1145/3206343.3206347

[59] Martez E. Mott, Shane Williams, Jacob O. Wobbrock, and Meredith Ringel Morris. 2017. Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2558–2570. https://doi.org/10.1145/3025453.3025517

[60] Aunnoy K Mutasim, Anil Ufuk Batmaz, Moaaz Hudhud Mughrabi, and Wolfgang Stuerzlinger. 2022. Performance Analysis of Saccades for Primary and Confirmatory Target Selection. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology* (Tsukuba, Japan) *(VRST '22)*. Association for Computing Machinery, New York, NY, USA, Article 18, 12 pages. https://doi.org/10.1145/3562939.3565619

[61] Aunnoy K Mutasim, Anil Ufuk Batmaz, Moaaz Hudhud Mughrabi, and Wolfgang Stuerzlinger. 2024. The Guided Evaluation Method: An easier way to empirically estimate trained user performance for unfamiliar keyboard layouts. *International Journal of Human-Computer Studies* 190 (2024), 103317. https://doi.org/10.1016/j.ijhcs.2024.103317

[62] Aunnoy K Mutasim, Anil Ufuk Batmaz, and Wolfgang Stuerzlinger. 2021. Pinch, Click, or Dwell: Comparing Different Selection Techniques for Eye-Gaze-Based Pointing in Virtual Reality. In *ACM Symposium on Eye Tracking Research and Applications* (Virtual Event, Germany) *(ETRA '21 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 15, 7 pages. https://doi.org/10.1145/3448018.3457998

[63] Aunnoy K Mutasim, Moaaz Hudhud Mughrabi, Anil Ufuk Batmaz, and Wolfgang Stuerzlinger. 2023. Does Repeatedly Typing the Same Phrase Provide a Good Estimate of Expert Text Entry Performance?. Extended Abstract. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, 8 pages. https://doi.org/10.1145/3544549.3585647

[64] Aanand Nayyar, Utkarsh Dwivedi, Karan Ahuja, Nitendra Rajput, Seema Nagar, and Kuntal Dey. 2017. OptiDwell: Intelligent Adjustment of Dwell Click Time. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces* (Limassol, Cyprus) *(IUI '17)*. Association for Computing Machinery, New York,

NY, USA, 193–204. https://doi.org/10.1145/3025171.3025202

[65] Allen Newell and Paul S Rosenbloom. 2013. Mechanisms of skill acquisition and the law of practice. In *Cognitive skills and their acquisition*. Psychology Press, Hillsdale, NJ: Erlbaum, 1–55.

[66] Prateek Panwar, Sayan Sarcar, and Debasis Samanta. 2012. EyeBoard: A fast and accurate eye gaze-based text entry system. In *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*. IEEE, Kharagpur, India, 1–8. https://doi.org/10.1109/IHCI.2012.6481793

[67] Pawan Patidar, Himanshu Raghuvanshi, and Sayan Sarcar. 2014. Quickpie: An Interface for Fast and Accurate Eye Gazed based Text Entry. arXiv:1407.7313 [cs.HC]

[68] Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N. Truong. 2015. Filteryedping: Design Challenges and User Performance of Dwell-Free Eye Typing. *ACM Trans. Access. Comput.* 6, 1, Article 3 (mar 2015), 37 pages. https://doi.org/10.1145/2724728

[69] Abdul Moiz Penkar, Christof Lutteroth, and Gerald Weber. 2012. Designing for the eye: design parameters for dwell in gaze interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference* (Melbourne, Australia) *(OzCHI '12)*. Association for Computing Machinery, New York, NY, USA, 479–488. https://doi.org/10.1145/2414536.2414609

[70] Jimin Pi and Bertram E. Shi. 2017. Probabilistic adjustment of dwell time for eye typing. In *2017 10th International Conference on Human System Interactions (HSI)*. IEEE, Ulsan, Korea (South), 251–257. https://doi.org/10.1109/HSI.2017.8005041

[71] Yuan Yuan Qian and Robert J. Teather. 2017. The Eyes Don't Have It: An Empirical Comparison of Head-Based and Eye-Based Selection in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 91–98. https://doi.org/10.1145/3131277.3132182

[72] Kari-Jouko Räihä and Saila Ovaska. 2012. An Exploratory Study of Eye Typing Fundamentals: Dwell Time, Text Entry Rate, Errors, and Workload. 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, 3001–3010. https://doi.org/10.1145/2207676.2208711

[73] Vijay Rajanna and John Paulin Hansen. 2018. Gaze Typing in Virtual Reality: Impact of Keyboard Design, Selection Method, and Motion. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research and Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 15, 10 pages. https://doi.org/10.1145/3204493.3204541

[74] Andrew D. Rice and Jonathan W. Lartigue. 2014. Touch-Level Model (TLM): Evolving KLM-GOMS for Touchscreen and Mobile Devices. In *Proceedings of the 2014 ACM Southeast Regional Conference* (Kennesaw, Georgia) *(ACM SE '14)*. Association for Computing Machinery, New York, NY, USA, Article 53, 6 pages. https://doi.org/10.1145/2638404.2638532

[75] Jochen Rick. 2010. Performance Optimizations of Virtual Keyboards for Stroke-Based Text Entry on a Touch-Based Tabletop. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 77–86. https://doi.org/10.1145/1866029.1866043

[76] Daniel Rough, Keith Vertanen, and Per Ola Kristensson. 2014. An Evaluation of Dasher with a High-Performance Language Model as a Gaze Communication Method. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (Como, Italy) *(AVI '14)*. Association for Computing Machinery, New York, NY, USA, 169–176. https://doi.org/10.1145/2598153.2598157

[77] Sayan Sarcar and Prateek Panwar. 2013. Eyeboard++: an enhanced eye gaze-based text entry system in Hindi. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction* (Bangalore, India) *(APCHI '13)*. Association for Computing Machinery, New York, NY, USA, 354–363. https://doi.org/10.1145/2525194.2525304

[78] Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: An Efficient Dwell-Free Eye Gaze-Based Text Entry System. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction* (Bangalore, India) *(APCHI '13)*. Association for Computing Machinery, New York, NY, USA, 215–220. https://doi.org/10.1145/2525194.2525288

[79] Immo Schuetz, T. Scott Murdison, Kevin J. MacKenzie, and Marina Zannoli. 2019. An Explanation of Fitts' Law-like Performance in Gaze-Based Selection Tasks Using a Psychophysics Approach. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300765

[80] Andrew Sears, Julie A. Jacko, Josey Chu, and Francisco Moro. 2001. The role of visual search in the design of effective soft keyboards. *Behaviour & Information Technology* 20, 3 (2001), 159–166. https://doi.org/10.1080/01449290110049790 arXiv:https://doi.org/10.1080/01449290110049790

[81] Ludwig Sidenmark and Hans Gellersen. 2019. Eye&Head: Synergetic Eye and Head Movement for Gaze Pointing and Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1161–1174. https://doi.org/10.1145/3332165.3347921

[82] R. William Soukoreff and I. Scott Mackenzie. 1995. Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard. *Behaviour & Information Technology* 14, 6 (1995), 370–379. https://doi.org/10.1080/01449299508914656 arXiv:https://doi.org/10.1080/01449299508914656

[83] R. William Soukoreff and I. Scott MacKenzie. 2001. Measuring Errors in Text Entry Tasks: An Application of the Levenshtein String Distance Statistic. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (Seattle, Washington) *(CHI EA '01)*. Association for Computing Machinery, New York, NY, USA, 319–320. https://doi.org/10.1145/634067.634256

[84] 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* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 113–120. https://doi.org/10.1145/642611.642632

[85] Robert J. Teather and Wolfgang Stuerzlinger. 2014. Visual Aids in 3D Point Selection Experiments. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction* (Honolulu, Hawaii, USA) *(SUI '14)*. Association for Computing Machinery, New York, NY, USA, 127–136. https://doi.org/10.1145/2659766.2659770

[86] Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash Away! Longitudinal Study of Fast Text Entry by Eye Gaze. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (Savannah, Georgia) *(ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 19–26. https://doi.org/10.1145/1344471.1344476

[87] Mario H. Urbina and Anke Huckauf. 2010. Alternatives to single character entry and dwell time selection on eye typing. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (Austin, Texas) *(ETRA '10)*. Association for Computing Machinery, New York, NY, USA, 315–322. https://doi.org/10.1145/1743666.1743738

[88] Boris Velichkovsky, Andreas Sprenger, and Pieter Unema. 1997. Towards gaze-mediated interaction: Collecting solutions of the "Midas touch problem". In *Human-Computer Interaction INTERACT'97*. Springer, New York, USA, 509–516.

[89] Oleg Špakov and Darius Miniotas. 2004. On-line adjustment of dwell time for target selection by gaze. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction* (Tampere, Finland) *(NordiCHI '04)*. Association for Computing Machinery, New York, NY, USA, 203–206. https://doi.org/10.1145/1028014.1028045

[90] David J Ward, Alan F Blackwell, and David JC MacKay. 2002. Dasher: A gesture-driven data entry interface for mobile computing. *Human–Computer Interaction* 17, 2-3 (2002), 199–228.

[91] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. ACM, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[92] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, Dwell or Gesture? Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4479–4488. https://doi.org/10.1145/3025453.3025964

[93] Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao, and Yuanchun Shi. 2016. One-Dimensional Handwriting: Inputting Letters and Words on Smart Glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 71–82. https://doi.org/10.1145/2858036.2858542

[94] Maozheng Zhao, Alec M Pierce, Ran Tan, Ting Zhang, Tianyi Wang, Tanya R. Jonker, Hrvoje Benko, and Aakar Gupta. 2023. Gaze Speedup: Eye Gaze Assisted Gesture Typing in Virtual Reality. In *Proceedings of the 28th International Conference on Intelligent User Interfaces* (Sydney, NSW, Australia) *(IUI '23)*. Association for Computing Machinery, New York, NY, USA, 595–606. https://doi.org/10.1145/3581641.3584072