Check for updates

A Usability Study of Nomon: A Flexible Interface for Single-Switch Users

Nicholas Bonaker
Massachusetts Institute of Technology
Cambridge, MA, USA
nbonaker@mit.edu

Emli-Mari Nel
University of the Witwatersrand
Johannesburg, Gauteng, South Africa
emlimari.nel@gmail.com

Keith Vertanen
Michigan Technological University
Houghton, MI, USA
vertanen@mtu.edu

Tamara Broderick
Massachusetts Institute of Technology
Cambridge, MA, USA
tbroderick@mit.edu

ABSTRACT

Many individuals with severe motor impairments communicate via a single switch—which might be activated by a blink, facial movement, or puff of air. These switches are commonly used as input to scanning systems that allow selection from a 2D grid of options. Nomon is an alternative interface that provides a more flexible layout, not confined to a grid. Previous work suggests that, even when options appear in a grid, Nomon may be faster and easier to use than scanning systems. However, previous work primarily tested Nomon with non–motor-impaired individuals, and evaluation with potential end-users was limited to a single motor-impaired participant. We provide a usability study following seven participants with motor impairments and compare their performance with Nomon against a row-column scanning system. Most participants were faster with Nomon in a picture selection task, while entry rates varied more in a text-entry task. However, we found participants had to click more times per selection using Nomon, motivating future research into mitigating this increased click load. All but one participant preferred using Nomon; most reported it felt faster and had better predictive text.

ACM Reference Format:

Nicholas Bonaker, Emli-Mari Nel, Keith Vertanen, and Tamara Broderick. 2023. A Usability Study of Nomon: A Flexible Interface for Single-Switch Users. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '23), October 22–25, 2023, New York, NY, USA*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3597638.3608415

1 INTRODUCTION

Single-switch methods are Augmentative and Alternative Communication (AAC) methods that afford computer interaction to individuals with severe motor impairments. Conditions like cerebral palsy, muscular dystrophy, or strokes can leave individuals with severely limited movement, necessitating the use of "switches" that might be activated by blinking, small facial movements, or puffs of air [3, 13–15, 23, 30]. These switches are commonly connected to scanning systems, where a user can activate—or "click"—their switch to select between options that are highlighted sequentially [33, 35]. Whichever option is highlighted at the time of the user's click is selected. This sequential scanning method quickly becomes tedious when more options are added, so a variant called Row Column Scanning (RCS) is often used. RCS requires two clicks to select an option, where options are arranged in a grid. The first click selects between rows, which are highlighted in turn; the second click selects between the columns within the previously selected row. RCS has three main limitations. (1) The method requires options to be placed in a strict grid layout, while many computer interactions (web pages, games, drawing, operating system navigation) are not confined to the this structure. (2) The selection method is inflexible; a single erroneous click will select the wrong row or column. (3) While RCS can more efficiently handle larger numbers of options than sequential scanning, users still experience substantially increasing dead-time as more options are added.

An alternative single-switch method, Nomon [6–8], directly addresses the above limitations. (1) Nomon's indicator-based method allows the placement of options anywhere on the screen. This feature has already allowed the adaptation of Nomon to facilitate children's games [24–26] and drawing interfaces [7]. (2) Nomon uses a probabilistic selection mechanism that is built to be flexible by learning a user-error profile. A single erroneous click does not necessarily result in an incorrect selection. (3) Previous research suggests that Nomon can better handle large numbers of options (greater than sixty) without accruing large dead-times like RCS [6]. This benefit could be useful for a symbol-based AAC application. Further, previous research found that users both composed text faster using Nomon and also felt that Nomon was easier to use than RCS [6–8]. However, previous work on Nomon has the following limitations:

(I) Previous research directly comparing Nomon to other single-switch methods tested only non–motor-impaired users. However, Bonaker et al. [6] did employ an accuracy-reducing switch with their non–motor-impaired participants in order to approximate reaction times more in line with motor-impaired switch usage.

(II) Previous research evaluating Nomon with single-switch users in text-entry tasks or tasks with a large number of options

ASSETS '23, October 22–25, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0220-4/23/10.
https://doi.org/10.1145/3597638.3608415

has been limited. López et al. [25, 26] trialed an implementation of Nomon modified as a children's game with single-switch users—namely, children between 4 and 14 years of age. But there were no more than 5 selection targets in the game. Bonaker et al. [6] trialed a keyboard implementation of Nomon with an experienced switch user; however, further research is needed, as this switch user had a high accuracy and cannot represent the wide range of abilities present across switch users [6].

(III) Existing keyboard implementations of Nomon are not fully accessible via a single-switch [6–8]. Accessing options that control key functionalities like the clock rotation speed required the use of a mouse.

We address concerns (I) and (II) by performing a user study directly comparing Nomon and RCS with seven switch users. We worked with two charity partners specializing in the care of AAC users—SpecialEffect and the Ace Centre—to identify a set of participants with a wide range of experience and abilities to better represent the switch-using population. Some participants were extremely precise and autonomous with their AAC setup, while others were slower and more prone to errors. The breadth of participants allowed us to answer our driving research questions: *"How does Nomon compare against RCS as a single-switch access method for actual end users? And what types of users might benefit from using Nomon?"*

To address concern (III), we partnered with the same charities, as well as the pilot participant in our study (participant B), to design and test a study website and Nomon interface that were fully switch accessible. We detail this work in Section 3.1.2.

We present our results as a series of case studies on individual participant experiences with Nomon and RCS in Section 4.5. We analyse the key trends across these users in Section 4.6. We make the following contributions in this work:

- A user study comparing Nomon and RCS with seven switch users with motor impairments of varying abilities. We compare their performance in both a picture-selection task and a text-entry task.
- An update to the Nomon interface making it fully accessible via a single switch alongside a guided tutorial (doubling as a calibration phase) designed to introduce potential switch users to Nomon.
- A public dataset of real, switch-user interactions with Nomon.

We encourage the reader to try out our implementation and tutorial for Nomon at https://nomon.app and share any feedback. Our code for the Nomon application is open source and can be accessed via the same link.

2 RELATED WORK

2.1 Single Switch Scanning

Text Entry Rates. Switch scanning can be slow; a 2018 survey of studies on entry rates for switch scanning users found a mean text entry rate of 1.27 words per minute (wpm) for scanning systems containing only character options [16]. However, for users that require scan rates longer than 1.5s, entry rates can reach as low as 0.3–0.5 wpm [19]. There has been considerable research into increasing entry rates in scanning systems. The same survey of studies found that adding word predictions can increase entry rates to a mean of 2.49 wpm [16]. Lesher et al. [21], Trnka et al. [37] also found that adding word predictions increased entry rate. Further, arranging more common characters so they are faster to select [9, 38], using a staircase layout for options to minimize the required number of scan steps [2, 18–20, 22], and identifying an appropriate scan speed for the user [20, 34] have all been found to increase text entry rates.

User Errors. User errors in scanning systems can happen in two ways. (1) The user can click at an incorrect time and select the wrong row. In this case, many scanning interfaces fall back to row scanning after a set number of column scans with no user input [17], with a concomitant cost in terms of entry rate. (2) The user can select the wrong column and thereby make an incorrect selection [4]; in this case, an error will cost both additional clicks and time to correct. Bhattacharya et al. [4], Koester and Simpson [19] have shown that time spent correcting errors in scanning systems can substantially decrease entry rates. User error rates can be large with scanning systems; in one study, 15% of scans by participants contained a timing error, even after configurations were optimized for each participant [19]. Researchers have proposed various methods for mitigating errors in scanning systems. Koester and Simpson [19] found that modifying scan settings—particularly using a slower scan delay and simpler scan pattern—greatly reduced user error rates. Adding more visual feedback on the scan progress could further reduce user error [29]. Recent advances in gaze prediction have sparked interest in a multi-modal interface combining switch-scanning and eye-tracking that could increase entry rates and reduce error rates [5, 10].

Applications Beyond Text Entry. Researchers have investigated various applications for scanning systems beyond text entry—from playing and navigating in video games [11, 41] to web browsing [36]. However the mechanics of switch scanning place some harsh constraints on these applications. Actionable options (like moving forwards or turning in a game, selecting a button or link on a web page) must be placed in a strict grid layout for user selection. Recent work has investigated breaking from this grid layout by using two moving bars (one horizontal and one vertical), allowing users to select arbitrary points on the screen by clicking to fix each bar's position in turn [32]. However, these methods leave little room for user error, requiring either high precision or slow scan speeds.

2.2 Alternative Single Switch Methods

Beyond RCS and Nomon, researchers have proposed alternative approaches to single switch communication. Ticker [31] is a single-switch interface intended for the blind and visually impaired. Unlike Nomon and RCS, the software is not used with a visual interface. Another interface, Dasher, was originally designed for a pointing device like a joystick or eye-tracking system. But a version of Dasher has been adapted to use a single-switch [27, 39]. Finally, Williamson et al. introduced an interface designed for binary input devices in brain-computer interfaces [40]. Their interface is designed to handle extremely noisy input methods, particularly the signals that would occur when measuring brain activity. We chose to focus our

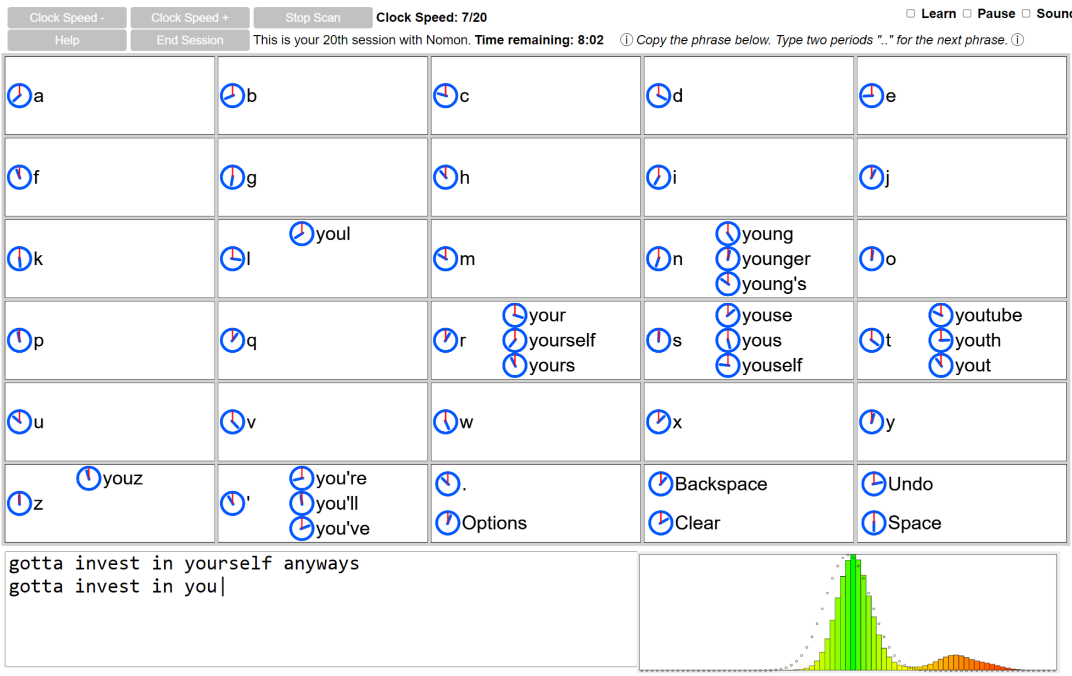

Figure 1: A screenshot of the Nomon keyboard interface used in our study. In this picture, the user is copying the phrase "gotta invest in yourself anyways" and has typed "gotta invest in you" so far. To continue this phrase, the user could select the letter "r" by clicking when the clock to the left of the letter passes noon. They may need to click multiple times when this clock passes noon to make the selection. The user also has the option to select the word prediction "yourself" to speed their text entry. The histogram in the bottom right shows the user's current click-time distribution—a representation of how accurately the user clicks relative to noon (the center of the histogram).

comparison to RCS as it is by far the most adopted approach in practice.

3 INTERFACE DESIGN

Here we detail the design choices made in our implementations of the Nomon and RCS interfaces.

3.1 Nomon

3.1.1 *Background.*

How Nomon works. The Nomon interface, shown in Figure 1, places an indicator clock next to each selectable option on the screen. The minute hands of all clocks rotate at the same speed, and each clock has a unique phase. To select an option in Nomon, the user needs to look only at the clock next to the desired option (unlike RCS, which requires the user to shift their visual attention around the screen to track highlighted rows). The user is instructed to click when the minute hand of their target clock passes the red "noon" line. After each click, the clock hands change phase to further narrow down the user's intended target. The user repeats this process, clicking each time the minute hand passes noon, until their target is selected. The number of clicks required to select a target depends on the user's precision and how probable the target is in Nomon's algorithm; the target probability in turn depends on the language model and Nomon's estimate of a particular user's error profile, described below. Experienced users are able to select options with around two clicks in a keyboard application that uses a language model and word predictions [6, 8]. A video demonstrating how to use Nomon is available in our supplementary materials.

Modeling user error in Nomon: the click-time distribution. We refer to the likelihood of when a user clicks relative to noon on their target clock as a user's click-time distribution. Estimating this likelihood is intrinsic to Nomon's operation. The histogram in the bottom right of the Nomon interface (Figure 1) provides a visual representation of Nomon's current estimate of the user's click-time distribution. The distribution is estimated from an initial calibration phase (where the user is instructed to click when particular clocks passes noon) and past clicks that led to the selection of a clock (therefore the location of the clock hand relative to noon at the time of the click can be assumed to be known). Nomon incorporates the click-time distribution as part of its probabilistic selection process. By applying Bayes' theorem to a series of user clicks, Nomon can calculate the posterior probability of each clock, ultimately selecting the one with the highest probability. The number of clicks required to make a selection with this probabilistic selection mechanism is determined by how quickly the posterior distribution over the clocks concentrates. Generally, users who click more precisely are able to select clocks in fewer clicks as their click-time distribution is less spread out.

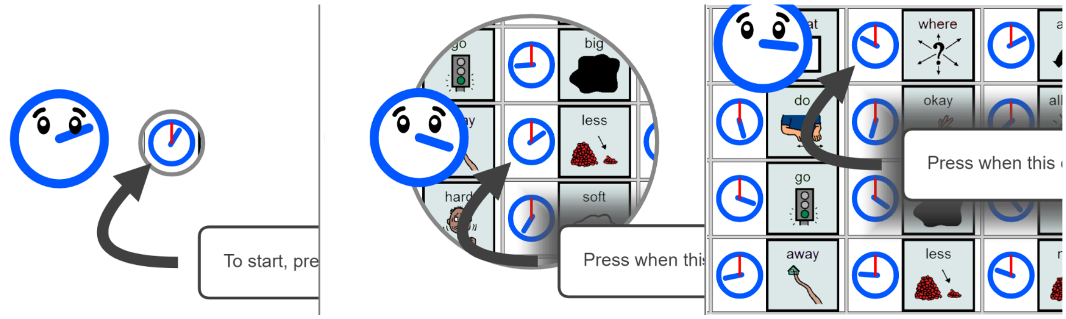

Figure 2: Three screenshots from our tutorial for Nomon. Progression through the tutorial is shown from left to right. In each frame, our helpful assistant, Norman the Nomon clock, points users to a specific clock to target. The user is instructed to click when this particular clock passes noon. The screen starts with a single clock visible, with more clocks revealed each time the user selects a clock. The tutorial ends with all clocks visible on the screen.

3.1.2 Making Nomon accessible via a single-switch. Prior to beginning this study, we consulted with staff at SpecialEffect and the Ace Centre to help us design appropriate study methods and ensure our study website was accessible to single-switch users. Part of this work involved developing an implementation of Nomon that was fully accessible via a single switch. Existing implementations of Nomon were not fully accessible; they required the use of a mouse to control the clock period and algorithm parameters [6, 8]. To address this issue, we created an options menu containing a grid of options that gave the user control of parameters like the clock rotation speed and the ability to ask for help or exit the study. Users could activate this menu by selecting the "options" clock seen in Figure 1. When the menu was activated, regular use of Nomon paused, and the options menu acted as a small RCS interface, highlighting the rows and columns in turn. We employed a similar form of RCS menu in both our RCS interface implementation and throughout the website to allow participants to navigate the study on their own. We then rigorously tested and iterated upon our design with the help of SpecialEffect staff. Following this development, our pilot participant (participant B, who uses single-switch methods daily) agreed to trial the study website and interfaces to flag any potential issues for switch users.

3.1.3 Designing a tutorial targeted for single-switch users. In addition to the accessibility concerns, we wanted to ensure the onboarding process of teaching participants how to use Nomon went smoothly. In initial discussions, our partner charities noted that teaching a switch user a new interface method can be a lengthy process. Some of our participants needed to switch away from their primary communication interface, which made conversation while using Nomon more difficult.

To help new users adjust to Nomon, we introduced a new tutorial before regular Nomon use began. SpecialEffect staff trialed our new tutorial and flagged for us that it needed important revisions to avoid confusion from new participants. In particular, as a result of their feedback, we identified two aspects in which using Nomon departs substantially from other switch-access methods. (1) There is no set number of clicks needed to make a selection on Nomon; rather the number is dependent on an option's probability and the user's precision. By contrast, RCS always requires exactly two clicks per selection. (2) Though there are many clocks on the screen, the user needs to look only at the clock they are trying to select.

Our SpecialEffect contacts cautioned that first-time users might become overwhelmed feeling that they needed to keep track of all the moving clocks.

We substantially expanded our tutorial to address the above concerns. The final version of our tutorial starts by asking users to click at noon for just a single onscreen clock. We then systematically increase the number of clocks on the screen. While the number of clocks increases, we still highlight a single target clock for the user to select. Since the target clock is known, the tutorial clicks can be used to jump-start the estimation of the user's click-time distribution—as was done for the calibration phase in [6].

To address (1), we added randomness to the number of clicks that users were asked to make on each target clock in the tutorial. Users clicked anywhere between two to four times for each clock. We added text prompts to the screen; we told users that they may need to click a few times to make a selection. Despite our efforts to highlight this variability in the tutorial, we found some participants in the study still had trouble understanding the varying number of clicks required to make a selection. Given this feedback, we are investigating ways to modify the tutorial to make this aspect more apparent.

We addressed (2) by starting with a single clock and obscuring the rest with a circular mask on top of the screen. Each time the user selected a target clock, we widened this mask to reveal more clocks. By the end of the tutorial, all clocks were visible on the screen. Snapshots from the tutorial visualizing this process are shown in Figure 2. After showing users how to select clocks, the tutorial continued by showing users where their selections are outputted and how to correct mistakes. It concluded by showing the user how to interface with the "Options" clock to change the clock rotation period and navigate the study website (described below in Section 3.1.4).

3.1.4 Our implementation. We heavily based our implementation of Nomon on the implementation used in [6]. We chose this implementation as a basis for three main reasons. (1) The authors used a simulation framework to optimize layout and algorithmic parameters for higher entry rate and lower click load. (2) The authors

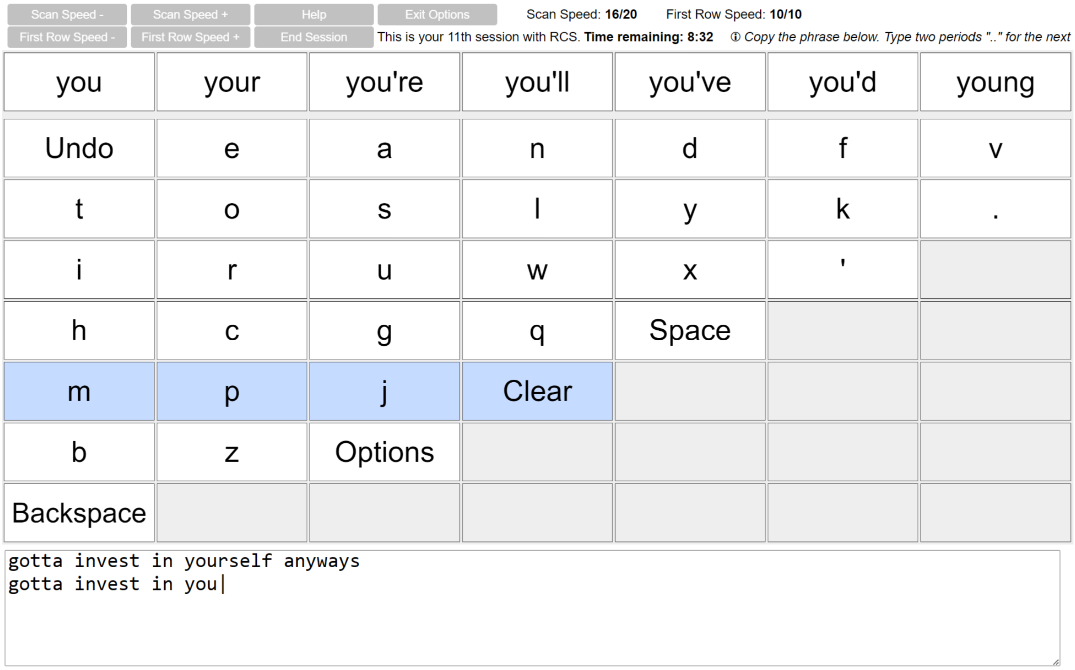

Figure 3: The row-column scanning keyboard used in our study. Here, the user is copying the phrase “gotta invest in yourself anyways” and has currently typed the text “gotta invest in you”. The interface works by progressively highlighting each row in turn. The currently highlighted row is selected when the user clicks their switch. The interface then progressively highlights the columns in this row, and the user clicks once more to select an option. The user could continue typing this phrase by selecting the letter “r”. Later, the user could select the word completion “yourself” if it appears in the top row.

involved AAC specialists and a switch-user in the design process. And (3) we wanted our results with switch users to be complementary to their results with non–motor-impaired users. We reproduce the main design choices and their implications below:

- The keyboard consisted of a 6×5 grid of main options. The main options consisted of: character options (the letters a–z); punctuation options (period, space, apostrophe); three corrective actions—backspace (removed the current last character), clear (removed all currently outputted text), and undo (reverted the last selection—be it a character, word, or corrective action); and an “Options” action (allowed access to the settings at the top of the page). Each character option plus apostrophe could display a maximum of three word completions to their right. In total, our implementation had a maximum of fifty options on the screen at any one time. Our keyboard layout differed from [6] in that we removed unnecessary punctuation symbols (comma, exclamation point, and question mark) that were not included in the target phrase set to make room for an additional “Options” action.
- The simulation study from [6] found that displaying a maximum of seventeen word completions on the Nomon interface struck a balance between higher entry rate and lower click load. We used the same maximum number of word completions in our implementation.
- There was one user-adjustable parameter for Nomon: the clock rotation period T. Bonaker et al. [6] allowed this value to range across $T \in [0.5, 4]$ seconds. Given the wide variation of precision and ability among our switch-user participants, we decided to modify this range to include longer clock rotation periods. We settled on values of $T = 6e^{-l/10}$ seconds for $l \in \{0, 1 \ldots 20\}$. Lower values of l corresponded to longer clock periods, with $T \in [0.82, 6]$ seconds. As shown in Figure 12, participant 7 used the fastest clock period of the group with $T = 1.64$ seconds, and participant 85 used the slowest of the group with $T = 5.43$ seconds.

3.2 Row Column Scanning

3.2.1 Background. RCS interfaces arrange options in a 2D grid for selection. The interface starts by highlighting rows of options in turn; it selects the currently highlighted row when a user clicks their switch. When a row is selected, the interface begins scanning through and highlighting the options in the selected row. The user can select the currently highlighted option by clicking their switch a second time.

3.2.2 Our implementation. Figure 3 shows our implementation of a RCS keyboard that we used in the study. As with our implementation of Nomon, we again based our RCS implementation heavily off of the one used by Bonaker et al. [6]. Rather than use a commercially available RCS software, the authors developed their own version to allow for as direct a comparison as possible between

Nomon and RCS. Their RCS implementation allowed both interfaces to use identical word prediction engines, experimental and logging controls, and selection options [6]. The authors based their design on The Grid 3 (a commercial scanning software) and followed configuration settings recommended by previous literature on switch scanning. We reproduce the main design choices and their implications below:

- The RCS keyboard consisted of a 8×7 grid of options. The main options (characters, punctuation, corrective actions) were identical to those in the Nomon implementation. Including seven word predictions, there were a maximum of forty options in the keyboard. This total represents two fewer options than in the implementation from [6]. As with our Nomon implementation, we removed 3 extraneous punctuation marks and added an options action.
- We used the staircase arrangement with characters sorted by frequency in the English language from [6]. These choices followed recommendations from [35, 38].
- Bonaker et al. [6] placed word predictions in the top row of the grid following the recommendation of [20].
- There were two user-adjustable parameters: scan time and extra delay. The scan time S controlled the length of time a particular row or column was highlighted. We followed Bonaker et al. [6] and allowed the scan time to vary as $S = 2e^{-j/14}$ seconds with $j \in \{0, 1, \ldots, 20\}$ [6]. Therefore, scan times ranged from 0.48 to 2.00 seconds, with longer scan times corresponding to smaller j. The extra delay D controlled an additional delay added to the first row or column scan. This parameter is common in RCS interfaces to afford the user more time to click [20]. Here we departed slightly from Bonaker et al. [6] and allowed the extra delay to range to longer values. The extra delay varied as $D = 0.2(10 - k)$ seconds with $k \in \{0, 1, \ldots, 10\}$. Therefore, extra delays ranged from 0 to 2 seconds, with longer extra delays corresponding to smaller k.

4 USER STUDY

4.1 Participants

With the help of SpecialEffect and the Ace Centre, we recruited seven participants (four male, three female) who regularly use AAC methods. The charities identified appropriate participants for our study from their respective client pools. SpecialEffect recruited six participants of which five were able to complete the user study in time for the paper submission deadline. The Ace Centre recruited four participants of which one user could finish in time for the paper submission. Both SpecialEffect and the Ace Centre provided email introductions for their participants. The authors had previously worked with the remaining participant. A more detailed overview of each participant can be found in Section 4.5. Each participant provided informed consent electronically.

4.2 Procedures

4.2.1 *Study Protocol.*

We designed our study protocol in collaboration with the charities that helped with participant recruitment. We took care to ensure our protocols were appropriate and accessible for all participants. These protocols were approved by both the MIT Committee on the Use of Humans as Experimental Subjects and the NHS Coventry and Warwick Research Ethics Committee prior to beginning the study. The study took place remotely; we met with participants over video conferencing, and all testing was completed through a website.

AAC Charity recommendations. When designing our study protocol, we had multiple discussions with our partner charities on how to accommodate the needs of participants in our procedures. These charities have worked extensively with the switch users they referred to our study and have previously supported these users in learning new AAC interfaces. They gave us a set of criteria that they believed would be flexible enough to suit the abilities of our diverse set of participants:

- Sessions should be broken into short periods of testing. Some of their switch-user clients experience fatigue when clicking their switches for extended periods of time. To mitigate the risk of fatigue and discomfort, they suggested testing periods should not exceed ten minutes.
- The spacing between sessions should be flexible. This recommendation served two purposes: to further mitigate the risk of participant fatigue or discomfort, and to allow participants leeway to fit the study sessions into their schedule. They mentioned that some participants would need to wait for a caretaker to be available to help them set up for sessions by opening the study website and ensuring their switch is working properly.
- Participants would need varying amounts of time to learn the new interfaces, and procedures should account for this variability. Some participants would have more experience trialing and learning single-switch interfaces, and some would even have experience with switch scanning. A fixed number of sessions—like the ten-session procedure used by Bonaker et al. [6] with non–motor-impaired participants—would not work well with our diverse participant pool.

Introduction session. The first session consisted of meeting potential participants (and their caretakers, if applicable) via video conferencing to explain the purpose and requirements of the study and obtain informed consent. Two of the authors were present in all video calls with participants at the request of SpecialEffect. For participants recruited from the Ace Centre, a staff member joined the initial meeting. The staff member helped introduce our team and ensure there were no issues with participants' switch-access setups. Time permitting, participants trialed both RCS and Nomon through our website to begin to familiarize themselves with the interfaces.

Practice sessions. The next phase of the study consisted of practice sessions. Data from these sessions was purely for practice, and we did not include it in our later analyses. For the first practice sessions, we again joined participants via Zoom to answer questions as they trialed both interfaces. At the recommendation of the charities, participants completed short, ten-minute picture selection tasks (see Section 4.3) with the Nomon and RCS interfaces. In later practice sessions, participants were free to choose which interface they wanted to practice more. We expected participants to need varying amounts of practice with both interfaces, particularly as some participants were experienced with switch scanning systems. We instructed participants to practice both interfaces until they felt ready to continue to the evaluation phase. They were free to space sessions as they saw fit; however, we asked that they complete at least two per week.

Evaluation sessions. In the evaluation phase, participants completed a picture-selection task identical to the task from the practice sessions. This phase consisted of a minimum of three sessions where participants completed the picture-selection task with both Nomon and RCS. Some participants elected to complete more than three evaluation sessions as they were willing and able.

Text entry sessions. This phase consisted of a minimum of three sessions lasting ten minutes where participants completed a text entry task (see Section 4.4) with both Nomon and RCS. Participant E opted not to complete this last phase of the study due to extenuating circumstances. We note this in his participant profile in Section 4.5.

Closing session. In the final session, we met with participants in a video conference to debrief. We showed participants a summary of their results from the study and collected a survey on their experience.

4.2.2 Finding an optimal clock period for Nomon. We worked with participants in the initial sessions to guide them towards an appropriate starting clock period. Finding an appropriate clock period is crucial for using Nomon effectively. If the clock period is too short relative to the width of a user's click-time distribution, the likelihood of a less precise click leading to an incorrect selection is higher. Conversely, if the clock period is too long for a precise user, they may miss out on potential gains in entry rate.

We started participants on the slowest clock period for their first time using Nomon; however, they were allowed to shorten the clock period at any point during the practice phase. If participants felt they were having to click too many times to make a selection, they were instructed to try a longer clock period. In addition, participants had visual feedback on their click-time distribution in the form of a colored histogram as shown in the bottom right corner of Figure 1. We instructed participants to aim to keep their click-time distribution close to the center and green. If they noticed their distribution had a significant amount of red (meaning they were clicking far away from noon), they could try lengthening the clock period (shrinking their click-time distribution relative to the clock rotation time).

4.3 Picture Selection Task

4.3.1 Procedure. The picture-selection task was designed to explore applications beyond text entry (like choosing among pictures in an album, selecting a file in a file system, or symbol-based AAC) where there are a large number of unordered options. Further, this task provides a direct comparison of the selection mechanisms in Nomon and RCS without the confounding effect of word completions on text entry rate. We adapted this task directly from the picture-selection task used in [6]. Picture options took the form of icons from a communication board that would be familiar to many AAC users (see Figure 4). In total, the interfaces contained sixty pictures available for selection. We instructed participants to select a series of five pictures and to type two periods ".." to signal they had completed this "phrase". We highlighted the next target picture in pink so that the user would not spend time searching for their target. We were primarily interested in the ability of users to select among these pictures, not memorize the layout.

Although Bonaker et al. [6] placed their picture-selection task in the final session of their study procedures, we decided to use it as a learning tool to teach participants how to use Nomon. The picture-selection task is simpler than the text-entry task for two reasons: (1) the picture options are static unlike word-prediction options, and (2) the current target option is highlighted for the user. We believed these simplifications made the task more conducive to learning to use Nomon for the first time.

4.3.2 Performance metrics. In what follows, we report performance according to the following metrics.

We calculated *entry rate* in selections per minute. We counted only selections present in the final composition; i.e., we excluded corrected selections. We measured the time spent on a phrase from the first switch activation up until the participant typed the last period to signal they were finished with a phrase.

We defined *click load* as clicks per selection. We counted only those selections present in the final composition; i.e., we excluded corrected selections.

We defined *correction rate* as the percent of selections that were a corrective action (Undo, Backspace, Clear) relative to the total number of selections used to type a phrase.

We defined *final error rate* (in percent) as the edit distance between the target sequence of selections and a participant's final output sequence, divided by the length of the target sequence. We calculated the edit distance as the Levenshtein distance which allows for insertions, deletions, and substitutions. Though communication board pictures have an associated text component, this text was purely for user experience and was not used for error rate calculations.

To generate Figures 5 and 7, we first compute each metric for each phrase typed by a user. Then, for a given metric, we collect the computed values across all the phrases for a given user. We plot the empirical median and quartiles of this collection of values.

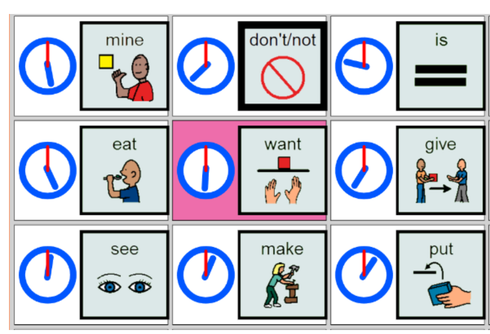

Figure 4: A zoomed section of the Nomon interface as we adapted it for the picture-selection task. We adapted the RCS interface in a similar manner; we replaced word predictions and characters with a grid of pictures.

4.4 Text Entry Task

4.4.1 Procedure. For the text-entry task, participants copied as many target phrases as possible in ten minutes. The target clock was not highlighted in this task (as it was in the picture selection task) as participants could choose to use character or word completion options. Participants typed two periods ".." to signal they had completed copying a phrase. Phrases were drawn uniformly at random (without replacement) from a pool of phrases designed to represent real text that people would compose in everyday life. We used the same phrase set as Bonaker et al. [6] in their evaluation of Nomon and RCS with non–motor-impaired participants.

This phrase set consisted of two subsets: the first set (IV) contained only words that were in the vocabulary of the language model used for our Nomon and RCS implementations, and the second set (OOV) contained phrases with exactly one word that was not in the language model's vocabulary. The OOV phrase set accounted for real world text entry where certain proper nouns or abbreviations might not appear in a language model. We mixed the phrase sets so that participants would type two IV phrases for every OOV phrase. The IV and OOV subsets had a mean phrase length of 7.15 (sd 1.60) and 7.24 (sd 1.64) words respectively [6].

4.4.2 Performance metrics. We calculated the *text-entry rate* in words per minute (wpm), where a word is defined as five characters including space. We counted only characters present in the final composition; i.e., we excluded corrected text. We measured the time spent on a phrase from the first switch activation up until the participant typed the last period to signal they were finished with a phrase.

We defined *click load* as clicks per character; we counted only characters present in the final composition (excluding corrected characters).

We defined *correction rate* as the percent of selections that were a corrective action (Undo, Clear) relative to the total number of selections used to type a phrase.

We defined *final error rate* (in percent) as the edit distance between the target phrase and a participant's final text output, divided by the length of the target sequence. We calculated the edit distance as the Levenshtein distance which allows for insertions, deletions, and substitutions.

To generate Figures 6, 8, and 9, we made analogous calculations to those for the picture-selection task, described in Section 4.3.2.

4.5 Individual Observations

We next describe our study participants in detail (see Table 1 for an overview) . For each participant, we compute the standard deviation of their click times relative to noon. We order the participants, and assign letter labels, according to this standard deviation. The smallest standard deviation, or highest precision, comes first (participant A). We use the same ordering in our figures.

4.5.1 Participant A. Participant A experienced a complete spinal cord injury that paralyzed him below the neck except for slight control of a finger on his right hand. He often used this finger to control his power wheelchair. He had full control of his computer cursor via a Natural Point Smart Mouse, and he often played intensely interactive video games using a Quadstick multichannel sip-puff switch. The participant also had a Buddy Button which he used as a single-switch for this study. He had no experience with scanning systems, but he considered himself extremely accurate with a single switch. The participant preferred to communicate verbally and enter text with his cursor and the Windows on-screen keyboard. The participant self-reported that he has minor dyslexia. Although it was not a problem for him during the study, he suggested adding an option to reduce the number of word predictions in Nomon or a dyslexia-friendly font option.

Participant A completed both the picture-selection and text-entry tasks. He was fastest of all participants using Nomon in both text entry and picture selection. His entry rate was slightly higher using Nomon for both tasks as shown in Figures 5 and 6. At the end of the study, the participant said he preferred using Nomon "by far—it was quicker and more accurate in the way it worked ... (it felt like predictive text)."

4.5.2 Participant B. Participant B had an advanced muscular dystrophy and used multiple switches to control his computer, smartphone, and power wheelchair. He most commonly used a gooseneck-mounted SCATIR switch and an EMG switch that detected small facial movements. His method of choice for text entry and text-to-speech communication was EZ Keys—a scanning software outfitted with custom, task-specific language models to speed text entry—which he operated at a fast 100ms scan speed.

Participant B was extremely proficient with single-switch scanning methods. He was particularly interested in testing Nomon as a text-entry method; as such, his sessions consisted entirely of the text-entry task. In lieu of practice sessions with the picture-selection task, the participant completed his practice sessions with the text-entry task. We evaluated his performance in the final three sessions.

In the text entry task, participant B's entry rates for both interfaces were among the highest out of the group we tested (with his RCS entry rate being considerably higher than the other participants). On average, he typed faster with RCS while his entry rate with Nomon was more consistent, as visible in Figure 6. In a final survey on his experience using both interfaces, participant B noted he preferred typing with Nomon; he felt it was "much easier to locate word predictions because the choices are adjacent to the next letter selection." He further mentioned that he felt his performance with RCS increased throughout the study as he rememorized the character layout (which differed from his day-to-day scanning system).

4.5.3 Participant C. Participant C had quadriplegic cerebral palsy. She generally used an eye-gaze setup for communication and control of her computer—specifically, the on-screen keyboards TD-Snap, TDControl, and Optikey. However, as this eye-gaze setup was vulnerable to infrared light from the sun, she often used a scanning system while in her outside wheelchair setup (or in a sunny room). When eye-gaze tracking is unusable, the participant has three Buddy buttons positioned around her headrest. The participant self-reported having dyslexia and that she found the many, similar word prefixes in the text-keyboard version of Nomon hard to read.

Participant C completed both the picture-selection and text-entry portions of the study. In the picture-selection task, her entry rate

ID	Sex	Diagnosis	Primary Single Switch (used for study)	Primary Text Entry Method	Tasks Completed
A	M	"Complete" spinal cord injury	Buddy Button positioned on shoulder	Windows on-screen keyboard	Picture, Text
B	M	Advanced muscular dystrophy	EMG switch	EZ Keys	Text
C	F	Quadriplegic cerebral palsy	Jelly Bean switch positioned on headrest	TDsnap, TDcontrol, and Optikey	Picture, Text
D	F	Ehlers–Danlos syndrome	Eye-gaze with blink detection	Dwell and Blink	Picture, Text
E	M	Spinal muscular atrophy	Laser beam switch	Windows on-screen keyboard	Picture
F	M	Stroke, quadriplegic	Quha Zono air mouse with thumb switch	Grid 3	Picture, Text
G	F	Cerebral palsy	Buddy Button positioned by left hand	Grid 2 and Dasher	Picture, Text

Table 1: Overview of participants and their primary text entry methods

was higher using Nomon (Figure 5). In the text-entry task, her entry rates were similar for both interfaces, if not slightly faster using RCS (Figure 6). However, her correction rate in the text-entry task was considerably higher using RCS (Figure 9). The participant preferred using Nomon at the end of the study; she reported that it felt faster to select options and easier to correct errors using Nomon.

4.5.4 Participant D. Participant D had Ehlers–Danlos Syndrome and used eye gaze and blink detection software to control her computer setup. Specifically, she used the Blink software with her eye gaze tracker for computer control and the Dwell eye gaze keyboard for text entry. She had very limited experience using single-switch software and scanning systems. For the purposes of this study, she emulated using a single switch by triggering a left-mouse press with her blink detection software. Further, the participant self-reported having difficulty seeing small icons on the computer screen such as the indicator clocks in the full-keyboard version of Nomon. She suggested adding an option to Nomon to reduce the number of clocks on the screen.

Participant D elected to complete both the picture-selection and text-entry portions of the study. We note that the participant had a month-long gap between when she was able to complete the picture-selection task and the text-entry task due to health complications. We asked her to complete a quick refresher session to re-familiarize herself prior to starting the text-entry task.

In both the picture-selection task and the text-entry task, her entry rate was considerably higher using Nomon as compared to using RCS (Figure 5). She was the only participant to have a lower click load using Nomon in the text-entry task (Figure 7). We believe there were two factors at play here: (1) the participant was able to utilize word completions on more occasions when using Nomon, and (2) her correction rate was considerably higher using RCS. Using word completions can dramatically reduce the click load in text-entry applications, and high levels of correction can inflate the click load. At the end of the study, Participant D stated that she preferred using Nomon because "it felt faster."

4.5.5 Participant E. Participant E was diagnosed with spinal muscular atrophy that greatly limited his motion below the neck. He communicated verbally and operated a smart head-mouse for full cursor control along with a laser beam switch, Buddy Button, and other micro-switches to simulate mouse and key presses on his computer. The participant regularly entered text using his cursor and an on-screen keyboard; however, he had often used scanning interfaces to control his smart home environment and TV.

Participant E completed five practice sessions with the Nomon interface. As he already felt proficient with scanning interfaces, Participant E completed a single practice session using RCS. He elected not to complete the text-entry task due to outside constraints.

On average, his entry rates in the picture-selection task were similar between Nomon and RCS while the variance for RCS was much larger (see Figure 5). He had the highest click load for Nomon out of the group we tested. Closer inspection of his selections revealed that this higher click load in Nomon was often the effect of mistakes and corrective actions. As such, his correction rate for Nomon was larger for more phrases than for RCS. However, his final error rate for Nomon was much lower than RCS. For two phrases using RCS, his final error rate was greater than 80%.

In all, the participant stated that he preferred using the RCS interface, mainly because he "found it frustrating to click so many times [with Nomon]." Despite it being his preference, the participant expressed that the downtime waiting for the rows and columns to scan in RCS was "boring."

4.5.6 *Participant F.*

Participant F experienced a stroke that completely paralysed him from the neck down (quadriplegic) and left him non-verbal. His main method of communication was a Grid Pad 12 from Smartbox paired with the Grid 3 software. He had a Quha Zono air mouse with a very sensitive thumb switch that provided a left click function for use in this study. He considered himself very experienced with single-switch scanning interfaces.

Participant F completed both the picture-selection and text-entry portions of the study. His entry rates in the picture selection task were similar for both Nomon and RCS, while his entry rate in the text-entry task was higher with RCS (Figures 5 and 6). He was the only participant where his correction rate was higher using Nomon in the text-entry task. Interestingly, he felt it was easier to correct errors in Nomon than RCS. After the study, he said he preferred using Nomon because he felt it was faster than RCS.

4.5.7 *Participant G.*

Participant G was diagnosed with cerebral palsy and used a joystick and multiple Buddy Buttons operated by her hands for computer control. She communicated verbally through an interpreter and regularly used Grid 2 and Dasher for text entry. In her early life, she primarily used a single switch and had considerable experience with single-switch communication methods.

Participant G reported that she can develop acid reflux when focusing on clicking precisely (as is required in the operation of Nomon and scanning systems). This acid reflux can lead to an increase in erroneous switch events when she uses her switch for long periods of time. To mitigate this effect and discomfort, the participant interspersed the short sessions of the study with regular rest breaks. Participant G completed both the picture-selection and text-entry portions of the study. She felt comfortable using the RCS interface as she had prior experience with scanning systems.

As noted above, this participant has a medical condition which can increase her tendency to click erroneously after using single-switch methods for longer periods of time. These erroneous activations were evident in higher final error rates for both Nomon and RCS. She acknowledged in a follow-up after the completion of the picture-selection task that she often chose not to correct errors in this task. We asked her to attempt more error correction in the following text-entry task.

Participant G operated Nomon and RCS with the slowest speed setting of the participants to mitigate the effects of her less accurate click timings. Her entry rates were comparatively lower as a result of these speed settings. In the picture-selection task, her entry rate was higher using Nomon. Her entry rates were similar for both interfaces in the text-entry task. At the end of the study, she preferred using Nomon "due to the increased predictive power it has over Row Column Scanning." However, she noted that Nomon initially took "a higher level of concentration compared to Grid 2 (RCS)." The participant further expressed frustration at the multiple clicks required to select options in Nomon and at the difficulty she experienced correcting errors in both interfaces.

4.6 Key Trends and Results

4.6.1 *Entry rates were higher with Nomon in the picture-selection task for the majority of participants.*

Figure 5 shows the entry rates for the six participants that completed the picture-selection task (A, C, D, E, F, and G). Four of these participants selected pictures faster using Nomon (A, C, D, and G), and two had entry rates that were similar for Nomon and RCS (E and F).

4.6.2 *Text-entry rates varied substantially by participant; some participants typed faster with Nomon, some with RCS, and some achieved similar performances with both interfaces.*

Text-entry rates for the six participants that completed the text-entry task (A, B, C, D, F, and G) are shown in Figure 6. Participant D typed faster with Nomon; two participants typed faster with RCS (B and F); and three had similar text-entry rates for Nomon and RCS (A, C, and G).

4.6.3 *Click loads were consistently higher for Nomon in both text entry and picture selection tasks.*

Picture Selection Task. Click loads from the picture selection task are shown in Figure 7. We found that all participants had higher click loads in this task—needing 1.18 more clicks per selection on average relative to RCS. This result is in line with the study of non–motor-impaired participants, where the click loads for Nomon were higher than RCS in a similar picture-selection task [6]. Some participants expressed frustration with the number of clicks needed to make a selection in this task. Certainly, clicking a switch can be taxing for some switch users—as was the case for participant G in our study. We explore possible methods for reducing this click load in the Discussion.

Text Entry Task. Figure 8 shows click loads from the text entry task. Five of the six participants that completed this task had a higher click load using Nomon, while participant D had a higher click load using RCS. This increase in click load relative to RCS was less pronounced than in the picture-selection task, likely due to a few factors. (1) The text-entry task includes word completions, which are known to reduce click load in single-switch interfaces [21]. (2) In the text-entry task, Nomon's selection mechanism is able to use prior information from the language model, which allows more likely characters and words to be selected in fewer clicks. And (3) the text-entry task has fewer selectable options than the picture-selection task; in Nomon, when the clock period is kept constant, we generally expect the number of clicks necessary to make a selection to increase with the number of options on the screen, whereas in RCS the number of clicks to make a selection is always two.

Interestingly, the degree to which Nomon's click load was increased relative to RCS varied substantially between users. For instance, participant G had one of the highest click loads for both tasks. As mentioned above, she had a tendency to make erroneous clicks, which both required more clicks for Nomon's selection mechanism to build confidence in the target clock and caused her to spend more time correcting errors. Similarly, participant E had a high click load for Nomon as well as a higher rate of errors and error correction.

4.6.4 *Correction rates were lower with Nomon for most users in text entry tasks.*

Figure 9 shows correction rates for participants in the

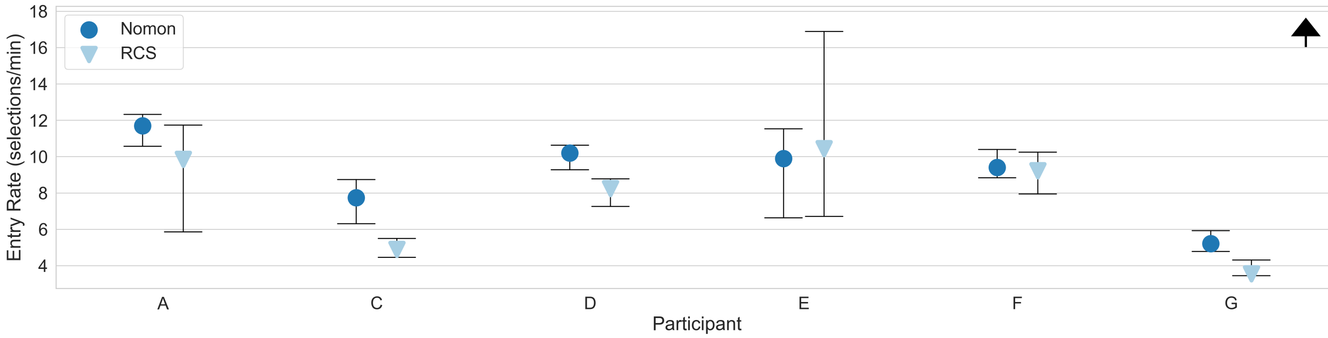

Figure 5: Entry rate (number of selections per minute) for the picture selection task across participants. Participants are arranged from left to right in order of decreasing click precision. The colored markers (a darker circle for Nomon; a lighter triangle for RCS) denote the median values for each interface and participant. Whiskers show the first and third quartiles. An arrow in the top right shows the direction of better performance.

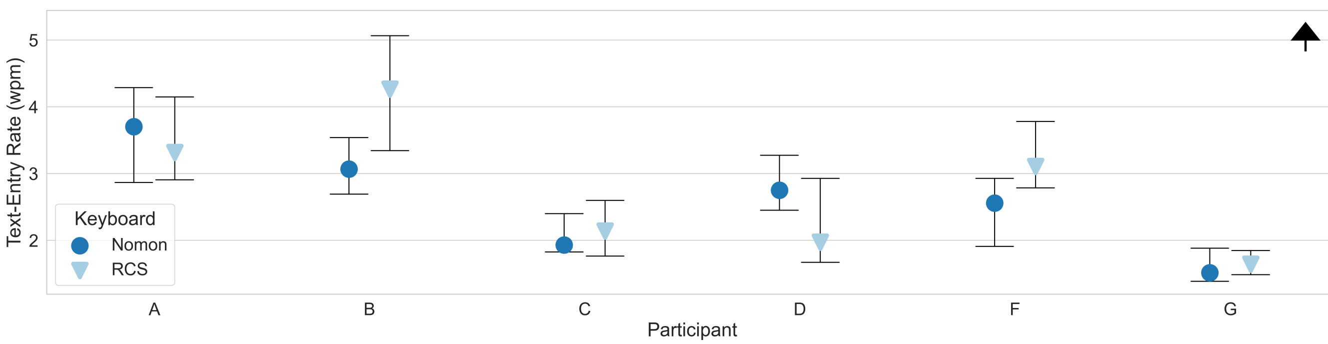

Figure 6: Text-entry rate (number of words per minute) for the text-entry task across participants. Participants are arranged from left to right in order of decreasing click precision. The colored markers (a darker circle for Nomon; a lighter triangle for RCS) denote the median values for each interface and participant. Whiskers show the first and third quartiles. An arrow in the top right shows the direction of better performance.

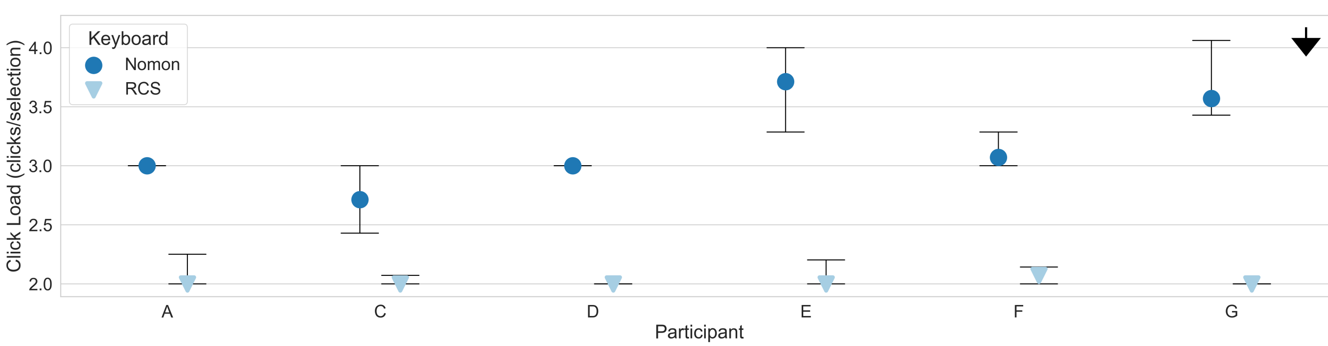

Figure 7: Click load (number of clicks per selection) for the picture selection task across participants. Participants are arranged from left to right in order of decreasing click precision. The colored markers (a darker circle for Nomon; a lighter triangle for RCS) denote the median values for each interface and participant. Whiskers show the first and third quartiles. An arrow in the top right shows the direction of better performance.

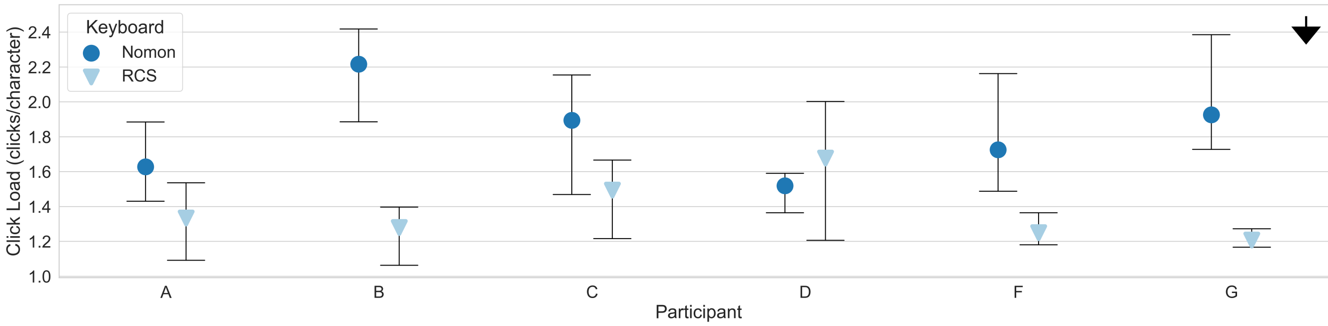

Figure 8: Click load (number of clicks per character) for the text-entry task across participants. Participants are arranged from left to right in order of decreasing click precision. The colored markers (a darker circle for Nomon; a lighter triangle for RCS) denote the median values for each interface and participant. Whiskers show the first and third quartiles. An arrow in the top right shows the direction of better performance.

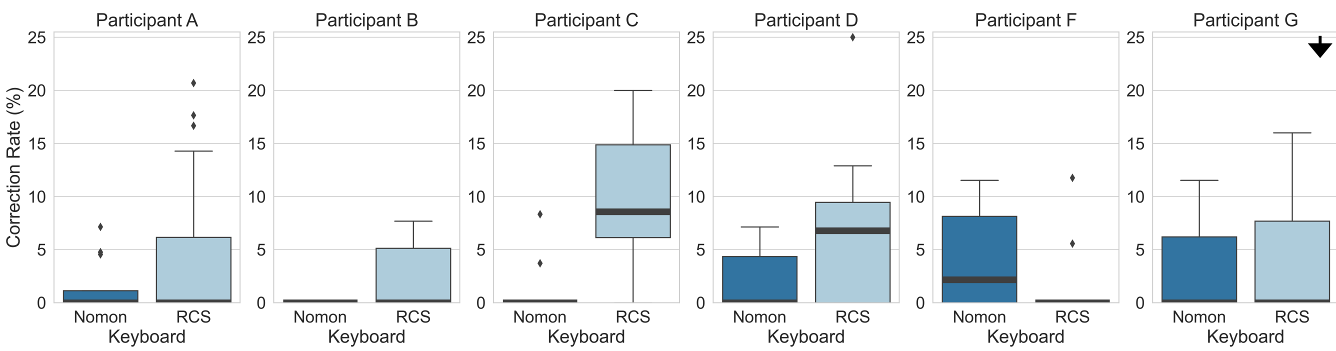

Figure 9: Correction Rate (% of selections that were a corrective action) for the text entry task across participants. Participants are arranged from left to right in order of decreasing click precision. The thick, black lines represent the medians values for each interface and participant. The colored regions are the first to third quartiles. Whiskers show the 5th and 95th percentiles. An arrow in the top right shows the direction of better performance.

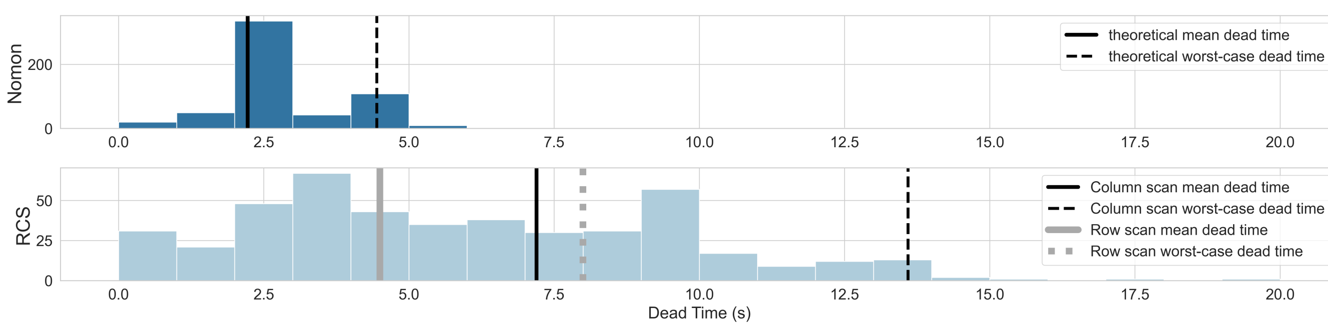

Figure 10: Distribution of dead-time between clicks for Participant C. The top histogram shows the participant's data for Nomon. The solid black line shows our theoretical estimate for the mean dead-time (2.22s), and the dashed black line shows our theoretical estimate for the worst-case dead-time (4.45s). The bottom histogram shows the participant's data for RCS. The solid grey line shows our theoretical estimate for the mean row dead-time (4.50s), and the dotted grey line shows our theoretical estimate for the worst-case row dead-time (8.00s). The solid black line shows our theoretical estimate for the mean column dead-time (7.20s), and the dashed black line shows our hypothesis for the worst-case column dead-time (13.59s). For this figure, we cut off any dead-times greater than 30s; there were two such dead-times for RCS and zero for Nomon.

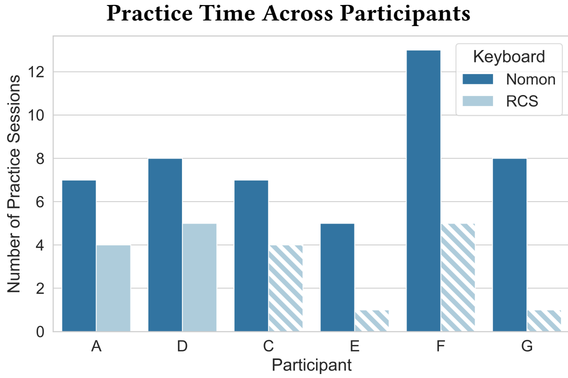

Figure 11: Comparison of practice time taken with Nomon (dark blue) and RCS (light blue) for each participant. Diagonal hatches on the RCS bars denote participants with prior experience using RCS. Only participants A and D had no prior experience using RCS.

text entry task. Nearly all participants had a higher correction rate using RCS—except for participant F, who made more corrections using Nomon. Further, participants C and F felt that it was easier to correct errors in Nomon, while the remaining participants expressed no preference for correcting errors with either interface.

4.6.5 Final error rates were near zero for both Nomon and RCS for a majority of participants. We instructed participants to copy phrases in both tasks as "quickly and accurately as possible," and we encouraged them to correct any errors. Resulting final error rates for most participants were low in both tasks.

In the text-entry task, the mean error rate for both interfaces for participants A, B, C, D, and F ranged from 0.00% to 0.93%. Participant G had slightly higher mean error rates of 3.89% with Nomon and 5.68% with RCS, perhaps an effect of her tendency to click erroneously as discussed in Section 4.5.7.

In the picture selection task, the mean error rate for participants A, C, D, and F ranged from 0.29% to 2.30%. Participant E had a much higher mean error rate of 14.2% with RCS, compared to 1.89% with Nomon. We note in Section 4.5.5 that this increase is primarily due to two phrases with a final error rate greater than 80%. Similar to the picture-selection task, participant G had relatively high mean error rates of 7.37% for Nomon and 10.1% for RCS.

4.6.6 Participants with no prior switch-scanning experience needed more practice time to learn to use Nomon than to learn RCS. Figure 11 shows the number of practice sessions completed with Nomon and RCS for each interface. The study protocol allowed participants to choose how much practice they felt they needed with each interface before they were comfortable to begin the evaluation phase. Participants A and D were the only two with no prior experience using RCS or switch-scanning methods. Though they were learning to use both Nomon and RCS for the first time, both participants used two fewer practice sessions with RCS than with Nomon. In initial sessions, participants often expressed that Nomon was more complicated to learn up-front, but become easier with practice. The non-fixed number of clicks required to make a selection in Nomon (as compared to the fixed, two clicks for RCS) was one of the most common sources of misunderstandings when using Nomon for the first time. On average, participants needed nine practice sessions (around ninety minutes total) to feel comfortable using Nomon.

4.6.7 All but one participant preferred typing with Nomon; most participants reported they felt Nomon was faster and had more predictive power. Six participants expressed that Nomon was their preferred interface at the end of the study. Of these six, three attributed their choice to increased predictive power with Nomon (A, B, and G). Though both our Nomon and RCS implementations used identical language models, Nomon affords far more use of the language model in the selection process. For one, unlike RCS, Nomon's selection mechanism allows characters that are more probable according to the language model to be selected faster. Second, including additional options in Nomon does not incur as large of a entry-rate cost as in RCS. So Nomon essentially allows a larger number of word predictions to be presented to the user.

Participant B further mentioned that he preferred the location of word predictions in Nomon, as their location adjacent to the next letter selection made them easier to find.

Four participants (A, C, D, and F) expressed a feeling that Nomon was faster than RCS. Interestingly, this was the case for two participants that had a slightly slower entry rate with Nomon in the text entry task (C, F). Participants also identified an increased dead-time when using RCS while waiting for rows and columns to highlight. Perhaps this dead-time added to their perception that RCS was slower; we explore the difference in dead-times between the interfaces below.

Conversely, participant E preferred using RCS to Nomon. He expressed frustration with the higher number of clicks required to use Nomon more often than other users. Click load was the primary reason he cited for his preferred interface. Participant E had the highest click load for Nomon; he averaged almost a full click more per selection (four clicks per selection) than most other participants in the picture-selection task.

4.6.8 Dead-time between clicks was considerably longer with RCS than with Nomon. We hypothesize that users may have generally perceived Nomon to be faster than RCS because they experienced more extremes of dead-time (when they are waiting to be able to click) with RCS than with Nomon. To help us test this hypothesis, we used a model to estimate the dead-time that we expected users to experience with Nomon and RCS as a function of the clock speed and scan speed. In what follows, we find that both our model as well as empirical observations from users support that the dead-time in RCS is substantially higher than in Nomon. In our derivations below, we recall notation from Sections 3.1.4 and 3.2.2.

Theoretical dead-time in Nomon. After each click in Nomon, the clock hands change phase. To make our calculations easier, we assume the hand on the user's target clock has a uniform probability of changing to any phase $p \in [0, T]$, where T is the rotation period of each clock in seconds. The user would then need to wait $T - p$ seconds for their target clock to reach noon; so $T - p$ is the user's dead-time between clicks (ignoring the case where the user pauses for multiple clock rotations). Under this model, the user's expected dead-time between clicks is given by $\mathbb{E}[T - p] = T - \mathbb{E}[p] = T/2$

seconds. The worst-case dead time under this model occurs when the clock phase initializes to $p = 0$ seconds, leading to a dead-time of T seconds.

Theoretical dead-time in RCS. In RCS, the dead-time a user experiences between clicks is completely determined by the option's location in the grid. The user experiences two distinct dead-times: one waiting for the scan to reach their desired row, and one for the desired column. In the picture-selection task, the target option has an equal chance of being in any of the six rows and any of the ten columns in the randomly selected row.

For an option in the first row, the user experiences 0 seconds of dead-time. For the second row, the user experiences one scan delay S plus the additional extra delay D, for total of $S + D$ seconds of dead-time. The third row has $2S + D$ seconds of dead time, and so forth. The theoretical dead-time for an option in the n^{th} row is then given by:

$$\text{deadtime}(n) = \begin{cases} 0 & n = 1 \\ (n-1) \cdot S + D & n \geq 2 \end{cases}$$

Under this model, the overall expected row dead-time for an interface with R rows is then:

$$\begin{aligned} \mathbb{E}\left[\text{deadtime}(n)\right] &= \frac{1}{R}\left(0 + \sum_{n=2}^{R}\left((n-1) \cdot S + D\right)\right) \\ &= \frac{1}{R} \cdot \left(\frac{R \cdot (R-1)}{2} \cdot S + (R-1) \cdot D\right) \end{aligned}$$

In fact, the same formula can be used for the column dead-time by substituting the number of columns for the number of rows. Using this formula for our interface with $R = 6$ rows, we see the expected dead-time for a row scan is $(15S + 5D)/6$ seconds. The expected dead-time for a column scan with 10 columns is likewise $(45S + 9D)/10$ seconds. The worst-case dead time under this model occurs when the desired option is in the sixth row and tenth column. In this worst case, we expect a dead-time of $5S + D$ seconds, and a column dead-time of $9S + D$ seconds.

We validated our theoretical dead-times by comparing them to the observed dead-times between clicks with both Nomon and RCS in the picture-selection task. Figure 10 shows the distribution of dead-times from participant C, but similar trends appeared for all other participants. Participant C's clock period in Nomon was $T = 4.45$ seconds. She used a scan delay of $S = 1.40$ seconds and an extra delay of $D = 1.00$ seconds in RCS.

For Nomon, the theoretical mean dead-time was 2.22 seconds, which is right in the middle of the participant's dead-time distribution. Further, the theoretical worst-case dead-time was 4.45 seconds, and nearly all of her dead-time distribution was less than this worst-case estimate. Dead-times greater than the worst-case estimate were likely due to waiting for more than one clock rotation. We note that the participant's dead-time distribution does not appear to follow our assumption that the phase initializes over a uniform range $p \in [0, T]$. However, our theoretical mean and worst-case estimates still align well with the data.

For RCS, her dead-time distribution was considerably more spread out compared to Nomon. The theoretical mean dead-time for row scans was 4.50 seconds, and the worst-case estimate was 8.00 seconds. The estimates for column scans were larger as there were more columns than rows. The theoretical mean dead-time for column scans was 7.20 seconds, and the worst-case estimate was 13.59 seconds. Again, nearly all of the participant's dead-time distribution falls left of the larger worst-case estimate.

Overall, participant C had a mean dead-time of 2.82 seconds with Nomon and 6.02 seconds with RCS—more than twice as long. Further, the worst case dead-time with RCS (13.59 seconds) was more than three times as long as with Nomon (4.45 seconds).

4.6.9 Click time distributions varied widely between participants, influencing their choice of clock period for Nomon. A detailed definition of click-time distributions can be found in the background information for Nomon, Section 3.1.1. We discuss how participants found an appropriate clock period in Section 4.2.2.

Figure 12 plots an estimate of the click-time distribution for each participant and their final, chosen clock period used in the evaluation phase of the study. In general, participants with wider (or less precise) click-time distributions chose slower clock periods. Participant C breaks from this pattern and elected to use a much slower clock period than her narrower click-time distribution could facilitate. She was reminded that she had the option to shorten the clock period, but felt the longer clock period made using Nomon easier.

5 DATASET OF NOMON INTERACTIONS

We developed a dataset of switch-user interactions with the Nomon interface from our user study. This dataset has potential applications for simulations of text entry with Nomon. Such simulations have successfully been used to optimize parameters and settings in scanning systems [4, 17, 28]. Initial simulations of Nomon have already been used to optimize parameters controlling the display of word predictions [6]; however, these simulations were based on data from interactions of non–motor-impaired users with Nomon. We believe the dataset we present here could be used to provide a more representative sample of switch users for use in simulations of Nomon. Here, we describe the the format of our dataset, which is available on our OSF repository (https://osf.io/9nx48).

5.1 Dataset Structure.

We structured our dataset on user interactions as follows. Data from each switch user in our study (Table 1) is presented in a unique CSV data table. Each row of these CSV tables represents a single click sent into the Nomon Keyboard. The columns in the dataset describe the context for each click and are detailed below:

- **Session Num** — The session number the data was drawn from. Sessions lasted 10 minutes each, though earlier sessions may be shorter.
- **Phrase Num** — Index of the click's phrase in the current session.
- **Selection Num** — Index of the click's selection towards typing the current phrase.
- **Click Num** — Index of the current switch press needed to make the current selection.
- **Phrase Text** — The target phrase presented to the user.

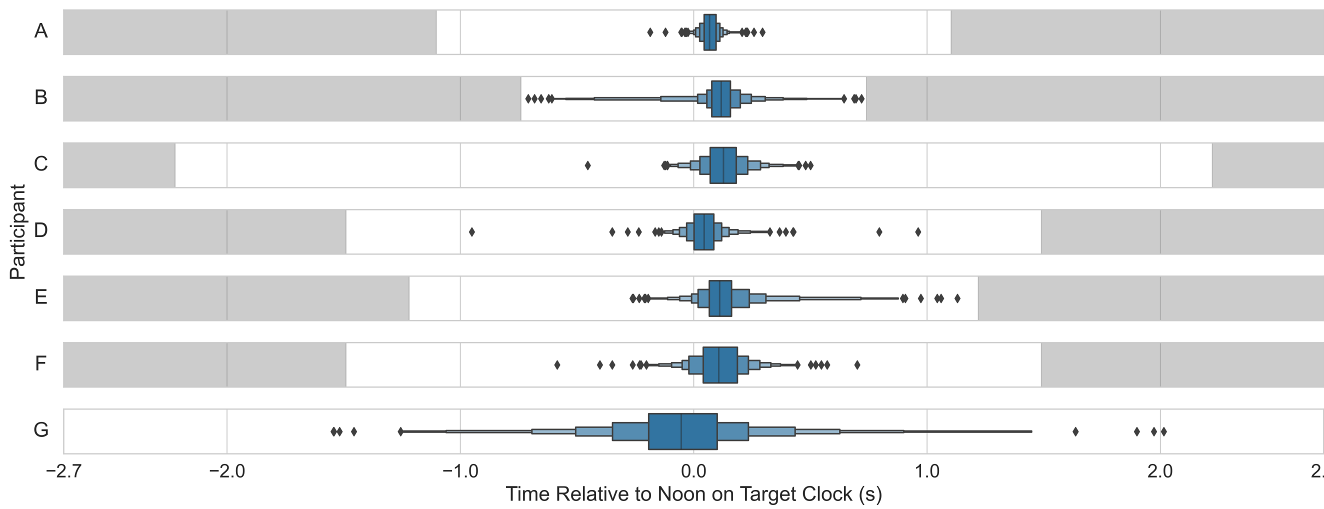

Figure 12: A visual representation of how participants clicked relative to Noon on their target clock. This visualization estimates the click-time distribution for each participant from the data we collected. Participants appear in order of decreasing click precision (top to bottom). The relative height of the blue boxes denote the number of clicks that fall in that range. Each participant's final, chosen clock period in Nomon (used for the evaluation phase of the study) is shown by the white area on the graph, centered at 0s (noon). Participant G used Nomon at the longest clock period: 5.4s.

- **Typed Text** — The text currently typed by the user on a given phrase. Note this may differ from the phrase text if the user made an error.
- **Target** — The target word/character highlighted for the user to select.
- **Selection** — The word/character/corrective option ultimately selected by the user.
- **Clock Period (s)** — The time in seconds it takes the clocks to make a full rotation.
- **Click Time Relative (s)** — The time that the user clicked their switch minus the time at which the clock they ultimately selected was at Noon, modulo the Clock Period. The time is recorded in seconds and takes values within $[-\text{Clock Period}/2, \text{Clock Period}/2]$.
- **Click Time Absolute (s)** — The global timestamp measured in seconds since epoch (Unix time) that the user clicked their switch.
- **Dead Time (s)** — The time in seconds since the last time the user clicked. Equal to the difference between the current and previous Click Time Absolute values. This value is set to Null for the first click in a phrase.

6 DISCUSSION

In this work, we evaluated the usability of Nomon as a communication method for single-switch users. We performed a user study with seven switch-users with motor impairments; we tested how Nomon compares to RCS in both a picture-selection task and a text-entry task. Crucially, these switch users represented a wide diversity of precision, ability, and experience with single-switch methods. In the text-entry task, we found entry rates varied substantially by participant; some typed faster with Nomon, some with RCS, and some performed similarly with both. All but one participant preferred typing with Nomon; most participants reported that Nomon felt faster (even if they had a slightly higher entry rate with RCS). In the picture-selection task, we found that the majority of participants selected pictures faster with Nomon. Click loads were consistently higher with Nomon in both tasks, with a larger difference in the picture-selection task.

We believe that investigating methods of reducing click load will prove a crucial aspect of Nomon's continued development. We see two directions for future work in this area. (1) Several aspects of Nomon's probabilistic selection mechanism remain untested since Nomon's introduction in [8]. Specifically, parameters controlling the learning and update of the user's click-time likelihood distribution could be candidates for optimization. The simulations conducted in [6] could be adapted with our new dataset to investigate the effects of these parameters on click load. (2) Given recent interest in multi-modal interfaces combining eye-gaze and switch scanning [5, 10], we hypothesize that a similar integration could reduce click loads in Nomon and speed entry rates. Nomon's probabilistic selection mechanism affords the addition of information sources beyond just the user's click-time distribution and language model priors. We theorize that even noisy information on where a user is looking on the screen could help Nomon's mechanism converge faster and with fewer clicks.

Limitations of the participant pool. To our knowledge, the present paper is the first work to trial Nomon with a variety of switch users. We note that our target population of single-switch users is a diverse group with varying abilities. It would not be feasible to capture a fully representative sample of this population with only seven participants. To reasonably run a well-powered statistical analysis or aggregate across participants, we would want a larger participant group that could be said to represent a uniform random sample of a meaningful population. Further, the participants who were able to complete our study were predisposed towards particular fluency with computer interaction and communication. In fact, we had to drop one participant in an introductory session as it was too difficult to effectively communicate with them over video conference. Therefore, we decided that presenting the results for each participant in their own context would best represent the data we collected. Seeing Nomon's potential for many of our participants, future work could expand upon our study and trial Nomon with a more expansive and varied set of switch users. These new users could provide informative experiences that are not captured in the results and dataset that we present here.

We now discuss implications for the future design of Nomon from our participants' feedback, experiences, and suggestions:

A more useful Nomon. Nearly all participants preferred using Nomon, and many asked if there was a version they could use in their AAC setup. As development of Nomon has primarily been focused in a research setting, we asked participants what features they would like to see that would make Nomon a more useful interface for real-world communication. Most crucially, participants suggested adding the ability to output text written in Nomon to other programs, or to use Nomon as an on-screen keyboard. Participants further expressed interest in a text-to-speech option for Nomon, similar to those available in other AAC interfaces.

We see a couple hurdles that future development of Nomon will need to overcome to allow for these features. (1) The Nomon code will need to be adapted to run on a user's computer as a local web application. Restrictions in the permissions of websites to interact with local files and programs will limit the ability to use the current web-based version of Nomon as an on-screen keyboard. Part of this work will include the addition of a language model on the user's local computer that does not require an internet connection. A local language model has the added benefit of allowing adaptation to a user's previously written text without the privacy implications of uploading their text to a cloud server. Adapting a user's language model has been shown to substantially improve performance on noisy text input [1, 12]. (2) The keyboard interface will need to be adapted to include punctuation, numbers, and special characters. Perhaps these options could be accessed by a special clock that toggles between the current character and word completion view and a numerals and symbols view.

Increasing accessibility. We asked participants for feedback on aspects of Nomon that bothered them or could be adjusted to further enhance accessibility for users. Two participants (C and D) recommended an option to reduce the number of word completions in the text entry interface to aid visibility, while participant B recommended an option to increase the number to speed text entry. One reason for allowing an option for fewer word predictions was to make room for larger clocks, giving users with vision problems an easier-to-see indicator. Having three word completion clocks in each character box was the main bottleneck on clock size in our Nomon keyboard implementation. Another reason for allowing a reduced number of clocks was to make the word completions more friendly to people with dyslexia. Participant C (who had dyslexia) had difficulty parsing the stacked word completions when the beginning of words were similar (for example, "young," "younger," and "young's" in the character box for the letter "n" in Figure 1). She further suggested an option for using a dyslexia-friendly font that could also help address this issue. Future implementations of Nomon could benefit from allowing users to choose between having fewer or more word completions in the interface as well as allowing users to select between different fonts.

In a similar vein, one participant expressed that the picture selection interface started to become disorienting towards the end of each session. She asked if there was a way to reduce the number of clocks on the screen for this task. This feedback suggests that, while Nomon's selection mechanism can handle large numbers of clocks, there may be an upper limit on the number of clocks that is comfortable for some users. In such cases, it may be possible to group some of the selectable targets together and have the user first select the group's clock followed by another selection of their desired target in that group (e.g., the user might first select a clock for writing numeric digits followed by selecting their actual desired digit). Future implementations of Nomon should keep this in mind when designing interfaces that involve a very large number of clocks. Further, it has occurred to us that a large number of clocks can be problematic if screen resolution is decreased to the point that some minute hands are effectively discretized, creating more competing targets. This problem has become more pronounced in the picture selection task. For this (and similar) tasks we hope to investigate other clock designs to enhance visible precision and timing.

7 CONCLUSION

We performed the first user study comparing the performance of Nomon and RCS with a diverse group of users with motor impairments. We examined performance in both a text-entry and a picture-selection task. Our results showed that most participants were faster with Nomon in the picture-selection task, while entry rates in the text-entry task varied more by user. Results also showed that participants had to make more clicks per selection with Nomon in both tasks. Overall, most participants expressed that they preferred typing with Nomon because it felt faster. In addition to the user study, we updated the Nomon interface with feedback from AAC charities and a switch user to make it fully accessible via a single switch. We provided a new tutorial for Nomon, targeted to teaching potential switch users how to use Nomon. Finally, we provided the first dataset of single-switch users' interactions with the Nomon interface. In summary, this work shows that Nomon is currently an effective method of communication for some single-switch users, and—with future work improving the click-time modeling—it has the potential to improve communication for a greater proportion of users.

ACKNOWLEDGMENTS

We are extremely grateful to the seven switch users that provided their time, efforts, and expertise throughout the study. Their feedback and suggestions will be instrumental in the continued development of Nomon. We are indebted to the staff of SpecialEffect and the Ace Centre for connecting us with switch users interested in our study, and for testing and feedback before we started the study. We especially thank Mick Donegan of SpecialEffect and Will Wade of the Ace Centre for serving as our principal contacts with these organizations and for their invaluable feedback on our study. And we are grateful for Will Wade's help with the NHS Research Ethics Committee review process. This work was supported in part by the NSF under Grant No. IIS-1750193.

REFERENCES

[1] Jiban Adhikary and Keith Vertanen. 2023. Language Model Personalization for Improved Touchscreen Typing. In *Proceedings of the International Conference on Spoken Language Processing* (Dublin, Ireland).

[2] Jennifer Angelo. 1992. Comparison of three computer scanning modes as an interface method for persons with cerebral palsy. *The American journal of occupational therapy* 46, 3 (1992), 217–222. https://doi.org/10.5014/ajot.46.3.21

[3] Jennifer Angelo. 2000. Factors affecting the use of a single switch with assistive technology devices. *Journal of Rehabilitation Research & Development* 37, 5 (2000).

[4] Samit Bhattacharya, Debasis Samanta, and Anupam Basu. 2008. User errors on scanning keyboards: Empirical study, model and design principles. *Interacting with Computers* 20, 3 (2008), 406–418. https://doi.org/10.1016/j.intcom.2008.03.002 Special Issue: On the Abuse and Misuse of Social Agents.

[5] Pradipta Biswas and Pat Langdon. 2013. A New Interaction Technique Involving Eye Gaze Tracker and Scanning System. In *Proceedings of the 2013 Conference on Eye Tracking South Africa* (Cape Town, South Africa) *(ETSA '13)*. Association for Computing Machinery, New York, NY, USA, 67–70. https://doi.org/10.1145/2509315.2509322

[6] Nicholas Ryan Bonaker, Emli-Mari Nel, Keith Vertanen, and Tamara Broderick. 2022. A Performance Evaluation of Nomon: A Flexible Interface for Noisy Single-Switch Users. 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 495, 17 pages. https://doi.org/10.1145/3491102.3517738

[7] Tamara Broderick. 2009. Nomon: Efficient communication with a single switch. University of Cambridge. Cambridge, United Kingdom.

[8] Tamara Broderick and David J. C. MacKay. 2009. Fast and Flexible Selection with a Single Switch. *PLoS ONE* 4, 10 (2009).

[9] W Crochetiere, R Foulds, and R Sterne. 1974. Computer aided motor communication. In *Proceedings of the 1974 Conference on Engineering Devices in Rehabilitation*. 1–8.

[10] S. K. Fager, T. Sorenson, E. Jakobs, H. Koester, T. Jakobs, and D. R. Beukelman. 2022. Multi-modal access method (eye-tracking + switch-scanning) for individuals with severe motor impairment: A preliminary investigation. *Assistive Technology* 0, 0 (2022), 1–9. https://doi.org/10.1080/10400435.2022.2053895 PMID: 35298355.

[11] Eelke Folmer, Fangzhou Liu, and Barrie Ellis. 2011. Navigating a 3D Avatar Using a Single Switch. In *Proceedings of the 6th International Conference on Foundations of Digital Games* (Bordeaux, France) *(FDG '11)*. Association for Computing Machinery, New York, NY, USA, 154–160. https://doi.org/10.1145/2159365.2159386

[12] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and Its Personalization on Touchscreen Typing Performance. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 649–658. https://doi.org/10.1145/2702123.2702503

[13] Chris Gibbons and Erin Beneteau. 2010. Functional performance using eye control and single switch scanning by people with ALS. *Perspectives on Augmentative and Alternative Communication* 19, 3 (2010), 64–69.

[14] Scott T. Grafton. 2010. Unlocking communication with the nose. *Proceedings of the National Academy of Sciences* 107, 32, 13979–13980.

[15] Kristen Grauman, Margrit Betke, Jonathan Lombardi, James Gips, and Gary R. Bradski. 2003. Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Universal Access in the Information Society* 2, 4 (2003), 359–373.

[16] Heidi Horstmann Koester and Sajay Arthanat. 2018. Text entry rate of access interfaces used by people with physical disabilities: A systematic review. *Assistive Technology* 30, 3 (2018), 151–163. https://doi.org/10.1080/10400435.2017.1291544 PMID: 28368689.

[17] Heidi H. Koester and Simon P. Levine. 1994. Modeling the speed of text entry with a word prediction interface. *IEEE transactions on rehabilitation engineering* 2, 3, 177–187.

[18] Heidi H. Koester and Jennifer Mankowski. 2015. Automatic Adjustment of Keyboard Settings Can Enhance Typing. *Assistive Technology* 27, 3 (2015), 136–146.

[19] Heidi H. Koester and Richard C. Simpson. 2014. Method for enhancing text entry rate with single-switch scanning. *Journal of Rehabilitation Research and Development* 51, 6 (2014), 995–1012.

[20] Heidi H. Koester and Richard C. Simpson. 2017. Effectiveness and usability of Scanning Wizard software: a tool for enhancing switch scanning. *Disability and Rehabilitation: Assistive Technology* 14(2) (2017), 161–171.

[21] Gregory Lesher, Bryan Moulton, and D Jeffery Higginbotham. 1998. Techniques for augmenting scanning communication. *Augmentative and Alternative Communication* 14, 2 (1998), 81–101.

[22] Gregory Lesher, Bryan Moulton, and D. Jeffery Higginbotham. 1998. Techniques for augmenting scanning communication. *Augmentative and Alternative Communication* 14, 2 (1998), 81–101. https://doi.org/10.1080/07434619812331278236 arXiv:https://doi.org/10.1080/07434619812331278236

[23] Brian Leung and Tom Chau. 2014. Autonomic responses to correct outcomes and interaction errors during single-switch scanning among children with severe spastic quadriplegic cerebral palsy. *Journal of neuroengineering and rehabilitation* 11, 1 (2014), 34.

[24] Sebastián Aced López, Fulvio Corno, and Luigi De Russis. 2015. Gnomon: Enabling dynamic one-switch games for children with severe motor disabilities. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. 995–1000.

[25] Sebastián Aced López, Fulvio Corno, and Luigi De Russis. 2015. Playable one-switch video games for children with severe motor disabilities based on GNomon. In *2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*. IEEE, 176–185.

[26] Sebastián Aced López, Fulvio Corno, and Luigi De Russis. 2017. Design and development of one-switch video games for children with severe motor disabilities. *ACM Transactions on Accessible Computing (TACCESS)* 10, 4 (2017), 1–42.

[27] David J. C. MacKay and Chris J. Ball. 2006. *Dasher's One-button Dynamic Mode – Theory and Preliminary Results*. Technical Report. Cavendish Laboratory, University of Cambridge.

[28] Robert Mankowski, Richard C. Simpson, and Heidi H. Koester. 2013. Validating a model of row–column scanning. *Disability and Rehabilitation: Assistive Technology* 8, 3 (2013), 321–329.

[29] D.T.G. Mariano, Y. Morére, P. Pino, A.A.R. de Sá, and E.L.M. Naves. 2022. Recovery Delay with Visual Feedback to Reduce Errors in Single-Switch Scanning Keyboards. *IRBM* 43, 1 (2022), 32–38. https://doi.org/10.1016/j.irbm.2020.10.003

[30] Gernot R. Müller-Putz, Christoph Pokorny, Daniela S. Klobassa, and Petar Horki. 2013. A single-switch BCI based on passive and imagined movements: toward restoring communication in minimally conscious patients. *International journal of neural systems* 23, 02 (2013), 1250037.

[31] Emli-Mari Nel, Per Ola Kristensson, and David J. C. MacKay. 2019. Ticker: An Adaptive Single-Switch Text Entry Method for Visually Impaired Users. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 11 (2019), 2756–2769.

[32] Mathieu Raynal and I. Scott" MacKenzie. 2022. TBS3: Two-Bar Single-Switch Scanning for Target Selection. In *Computers Helping People with Special Needs*, Klaus Miesenberger, Georgios Kouroupetroglou, Katerina Mavrou, Roberto Manduchi, Mario Covarrubias Rodriguez, and Petr Penáz (Eds.). Springer International Publishing, Cham, 338–346.

[33] Richard C. Simpson, Heidi H. Koester, and Ed LoPresti. 2006. Evaluation of an adaptive row/column scanning system. *Technology and disability* 18, 3 (2006), 127–138.

[34] Richard C. Simpson, Heidi H. Koester, and Ed LoPresti. 2007. Selecting an Appropriate Scan Rate: The .65 Rule. *Assistive Technology* 19 (2007), 51–58.

[35] Richard C. Simpson, Robert Mankowski, and Heidi H. Koester. 2011. Modeling one-switch row-column scanning with errors and error correction methods. *The open rehabilitation journal* 4, 1 (2011).

[36] Laurianne Sitbon, Oscar Wong, and Margot Brereton. 2014. Efficient Web Browsing with a Single-Switch. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design* (Sydney, New South Wales, Australia) *(OzCHI '14)*. Association for Computing Machinery, New York, NY, USA, 515–518. https://doi.org/10.1145/2686612.2686693

[37] Keith Trnka, John McCaw, Debra Yarrington, Kathleen F. McCoy, and Christopher Pennington. 2009. User Interaction with Word Prediction: The Effects of Prediction Quality. *ACM Transactions on Accessible Computing* 1, 17:1–17:34. Issue 3.

[38] Horabail Venkatagiri. 1999. Efficient keyboard layouts for sequential access in augmentative and alternative communication. *Augmentative and Alternative Communication* 15, 2 (1999), 126–134.

[39] David J. Ward, Alan F. Blackwell, and David J. C. MacKay. 2000. Dasher - a Data Entry Interface using Continuous Gestures and Language Models. ACM Press, 129–137.

[40] John H. Williamson, Melissa Quek, Iulia Popescu, Andrew Ramsay, and Roderick Murray-Smith. 2020. Efficient human-machine control with asymmetric marginal reliability input devices. *Plos one* 15, 6 (2020), e0233603.

[41] Bei Yuan, Eelke Folmer, and Frederick C. Harris. 2011. Game accessibility: a survey. *Universal Access in the information Society* 10, 1 (2011), 81–100.