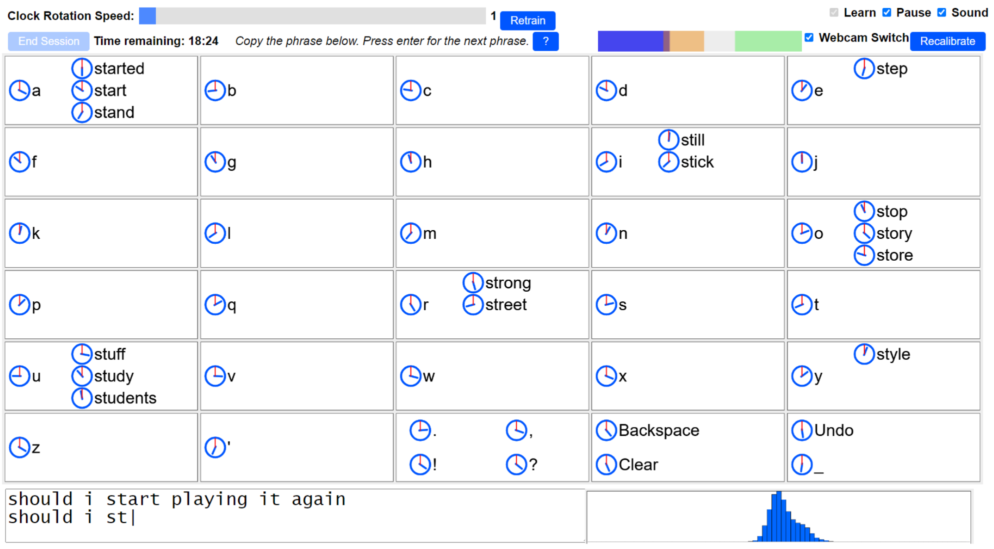
A Performance Evaluation of Nomon: A Flexible Interface for Noisy Single-Switch Users

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



Time remaining: 18:24

Copy the phrase below. Press enter for the next phrase.

☐ Learn ☑ Pause ☑ Sound ☑ Webcam Switch

```
should i start playing it again
should i st|
```

Figure 1: Nomon being used to enter the phrase "should i start playing it again". The user has typed "should i st" thus far. To write the letter "a", the user clicks a switch when the clock near "a" is at noon. Target selection may require multiple clicks depending on the user's switch precision (visualized by the histogram). The user can select word completions to speed writing.

ABSTRACT

Some individuals with motor impairments communicate using a single switch — such as a button click, air puff, or blink. Row-column scanning provides a method for choosing items arranged in a grid using a single switch. An alternative, Nomon, allows potential selections to be arranged arbitrarily rather than requiring a grid (as desired for gaming, drawing, etc.) — and provides an alternative probabilistic selection method. While past results suggest that Nomon may be faster and easier to use than row-column scanning, no work has yet quantified performance of the two methods over longer time periods or in tasks beyond writing. In this paper, we also develop and validate a webcam-based switch that allows a user without a motor impairment to approximate the response times of a motor-impaired single switch user; although the approximation is not a replacement for testing with single-switch users, it allows us to better initialize, calibrate, and evaluate our method. Over 10 sessions with the webcam switch, we found users typed faster and more easily with Nomon than with row-column scanning. The benefits of Nomon were even more pronounced in a picture-selection task. Evaluation and feedback from a motor-impaired switch user further supports the promise of Nomon.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9157-3/22/04.
https://doi.org/10.1145/3491102.3517738

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in accessibility**; **Keyboards**; **Empirical studies in HCI**.

KEYWORDS

Augmentative and alternative communication; accessibility; single-switch scanning systems; text entry;

ACM Reference Format:
Nicholas Bonaker, Emli-Mari Nel, Keith Vertanen, and Tamara Broderick. 2022. A Performance Evaluation of Nomon: A Flexible Interface for Noisy Single-Switch Users. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3491102.3517738

1 INTRODUCTION

Individuals with severe motor impairments, such as cerebral palsy or locked-in syndrome, often communicate via augmentative and alternative communication (AAC) devices with single switch input [13, 30, 42]. Users control the activation time of the switch by, e.g., pressing a button, releasing a puff of air, or blinking [3, 14, 15]. Most commonly, these switch activations (henceforth "clicks") are used as input to a scanning interface [52, 54]. The graphical user interface highlights different options in turn; the interface chooses whichever option is highlighted when the switch is activated. But highlighting every option in sequence can be inefficient for even a moderate number of options. While a popular variant called row-column scanning is more efficient, it requires that options be arranged in a grid. Computer users often need to choose among options not arranged in a grid; e.g. in drawing, gaming, and web browsing.[1]

Nomon [6, 7] offers a more flexible user experience. Nomon places an indicator next to each selection option, and uses a probabilistic selection mechanism to avoid inefficiently visiting each option in turn. Previous research suggests users can type more quickly and easily using Nomon than using row-column scanning [6, 7]. However, there were four main deficiencies in past work:

(A) Previous research tested each user for less than an hour total using Nomon. Performance may differ with more experience.
(B) In past research, non–motor-impaired users triggered Nomon using a joystick button. In this paper, we refer to individuals who do not regularly use AAC switches as "non–switch users." Motor-impaired individuals often exhibit different single-switch reaction times relative to non-switch users.[2]
(C) Previous research had users enter phrases consisting primarily of common words. Such phrases are easier to write in Nomon due to its use of a language model and the interface's word completions. But a text entry method also needs to support the input of uncommon words (e.g., proper names).
(D) Previous research tested only text entry. But Nomon promises to enable efficient input for tasks beyond text entry.

Not only do we address these concerns in the present study, but we also improved the Nomon program to increase accessibility before running our study. To improve the Nomon program, we first consulted with AAC experts[3] and a single-switch user. Based on their feedback, we designed a more accessible interface for Nomon. Moreover, our AAC consultants flagged that the original Nomon initialization was likely to require impractical or costly manual interactions. So we developed a more suitable initialization process. Further, we adapted simulation methods used to optimize internal parameters in scanning systems [20, 40, 54] to design a more efficient keyboard layout for Nomon.

After these updates, we addressed concern (A) by performing a longer user study; we collected data on each study participant's use of both row-column scanning and Nomon across 10 sessions.

We addressed (B) in two parts. First, we tested the performance of a single-switch user[4] with the Nomon interface. Second, recognizing the especially valuable time of motor-impaired users, we focused our larger-scale testing on non–switch users. But crucially we developed and validated the use of a webcam-based switch to allow non–switch users to better approximate reaction times of motor-impaired users. Although our approximation is not fully representative of all single-switch users, our results in the present paper employ click timings that are better aligned with the target population than the original Nomon study. This approximation allowed us to develop and test our new initialization method and the general effect of a noisier switch with non–switch users. Our results were also useful to convince our collaborating charities that a study involving single-switch users would be worthwhile since a practical initialization procedure was deemed critical.

To address (C), we selected phrases so that a third contained a challenging word not in our language model's vocabulary. With this diverse sent of phrase prompts, we could not only measure user performance with relatively simple text, but also explore whether Nomon degrades gracefully in the face of harder-to-predict text.

To address (D), we also compared performance of row-column scanning and Nomon in a task beyond text entry. There are many potential uses of Nomon such as gaming [32–36], drawing ([6], Section 7.1), and general operating system control ([6], Section 7.3, Section 2 of our supplement). But to facilitate comparison, we focus on a task where row-column scanning can still be applied: selection from a large set of pictures. We expect similar behavior when selecting among files on a desktop, selecting a computer application to launch, or selecting products at an online retailer.

Our results demonstrate that, under these conditions, users find it faster and easier to enter text using Nomon than using row-column scanning. In the text-entry task, participants typed 15% faster with Nomon and rated it easier to use. The benefits of Nomon are even more pronounced in the picture-selection task where participants selected targets 36% faster. We make the following contributions:

- An updated and easily available Nomon interface, redesigned to increase accessibility through feedback from switch users and AAC specialists.
- A model of a Nomon user and a subsequent simulation study to optimize the design of the Nomon interface.
- A user study comparing non–switch users' performance with Nomon and row-column scanning in: (1) a text entry task with challenging out-of-vocabulary words and (2) a picture-selection task to simulate applications beyond text entry.

[1] Currently switch users are primarily limited to games and websites that have certain, constrained switch-friendly formats; see, e.g., https://www.bltt.org/switch/activities.htm and https://everydayspeech.com/adventures-switch-accessible-websites/.
[2] See the first row of Figure 16 (Section 5) below, showing data kindly provided by Dr. Heidi Koester and collected in [22, 24].
[3] The AAC experts we consulted include staff at charities SpecialEffect https://www.specialeffect.org.uk/ and the Ace Centre https://acecentre.org.uk/.

[4] The user we consulted about interface design and the user whose performance we tested were two different single-switch users.

- A user trial of Nomon in a text-entry task with a motor-impaired switch user.
- A method for approximating motor-impaired reaction times with non–switch user inputs, and a validation of this method.

The rest of this paper is structured as follows. We survey approaches to single-switch text composition in Section 2. We detail how our two interfaces operate, and justify our interface and study design choices, in Section 3. We describe our user study and picture-selection task in Section 4. We describe our method for approximating reaction times of motor-impaired users in Section 3.5 and formally justify it in Section 5.

2 RELATED WORK

2.1 Input via scanning

Individuals with motor impairments tend to write slowly using row-column scanning (RCS): Koester and Simpson [23] observe entry rates of 0.3–2.9 words per minute (wpm), and Roark et al. [48] observe 1.9 wpm. Researchers have investigated various approaches to speed text entry. Arranging letters cleverly in the grid can speed selection [10, 59]. Carefully configuring the scanning interface can substantially impact performance [2, 22–24, 28, 53]. Instead of scanning rows and columns sequentially, the interface can highlight subsets of cells in some way, e.g. via Huffman coding [4, 47, 48] or a language model [62]. Character or word predictions may speed input [29, 57], but not in all cases [20, 21, 23]. Scanning can be used for applications other than text input, e.g. navigation in virtual environments [12] or playing games [66]. However, despite these efforts, scanning requires choosing either (1) a fast scan rate that risks false selections or (2) precise target selection but with a slow scan rate. A further obstacle is that applications must be designed to fit the scanning paradigm, e.g. by placing options in a grid.

2.2 Selecting a moving target within a time window

Selecting a moving target within a particular time window is a task that arises in some situations, e.g., smartphone games. For this task, researchers have studied error rates and models of timing performance [25, 26] as well as automated game playtesting [27]. Controlling a timed activation in these cases (e.g., via a button or screen press) might be interpreted as an activation of a single switch. As it stands, though, the task remains distinct from either RCS or Nomon. In particular, in the moving target task, there is a continuous movement of some visual element, and the goal is to click when the element is in some spatial range, implying a particular time range for selection. In RCS, there is a fixed time window for making a selection, but there is no continuous visual element; instead the user is shown only the discrete highlighting of rows or columns. Nomon, by contrast, presents a continuous visual element (the rotating clock hand), but there is no fixed time window during which clicking realizes some goal. Rather, in Nomon, changing the timing of a user's click induces a continuous change in the observed likelihood value, and the likelihood shape itself is user-specific. This shape need not be Gaussian or even unimodal, and is learned by the Nomon method. It would be interesting to explore whether using a continuous visual prompt inspired by gameplay might aid, e.g., in the usability of RCS interfaces.

Additional work in this vein models the neuromechanical process of a finger pressing a physical button in a non–switch user [18, 44]. Single-switch users, though, often have specialized switches that may be activated by different body parts. For example, some switches detect puffs of air, electrical muscle activations, or blinking. It remains to be seen if modeling techniques in the same spirit might by usefully applied to these other forms of switch input.

2.3 Input via a noisy switch

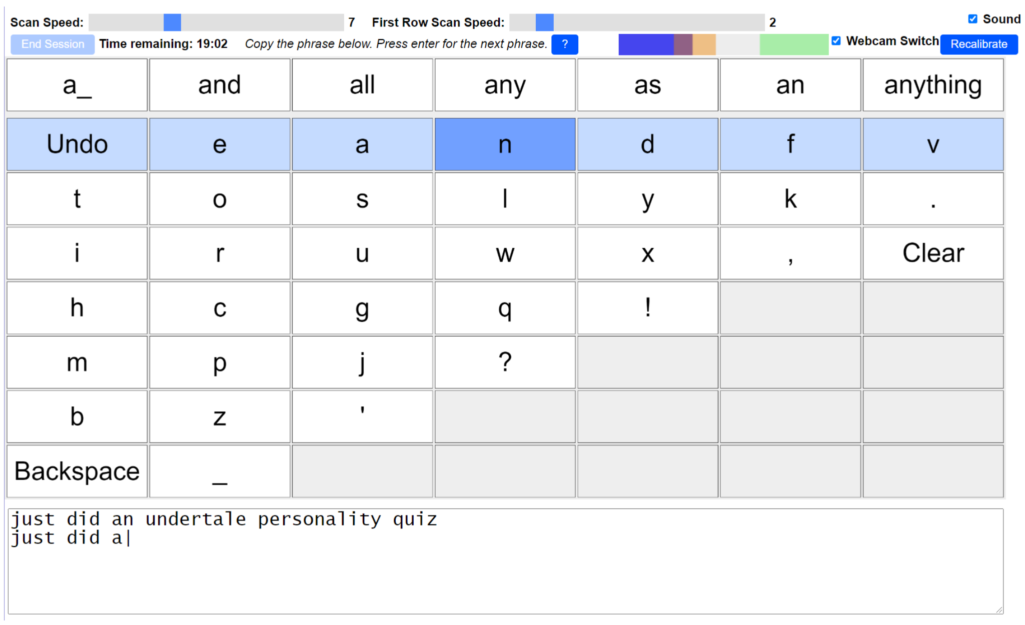
In the Dasher interface, users write by navigating through a world of nested letter boxes [63]. The size of each letter's box is based on language model that adapts as the user writes. Typically Dasher is operated via a pointing device such as a mouse or eye-tracker [49, 58]. Dasher can also be controlled via a single switch [37, 38]. Dasher applies Shannon's noisy-channel coding theorem [51] to facilitate efficient text entry. Dasher's navigation interface allows selection of multiple letters or even entire words with a single click. This mechanism can reduce the physical effort and time required of users. A pilot study showed a non–switch-using expert could write at 10 wpm using only 0.4 clicks per character [37]. To our knowledge, there have been no further user studies of one-button Dasher. MacKay et al. [38] note that the capacity of the channel is substantially reduced by a noisy switch with an erroneous activation a fraction f of the time. Also Dasher requires options be arranged in a strict order (e.g., alphabetically) which can limit applications beyond text entry.

Unlike Dasher, Nomon does not require an ordering on selection options. Nomon also explicitly models, and nonparametrically learns, a distribution describing how a user clicks relative to a baseline time. Dynamically adapting to the user's particular clicking style, represented by this click-time distribution, is a novel aspect of Nomon in the context of single-switch text entry methods. Theoretically, adaptation should result in less error correcting and faster text entry. A game for children with motor impairments [33, 34, 36] demonstrates Nomon's applicability in real life. Nel et al. [43] extended Nomon's noise model to develop a communication method for single-switch users who are also visually-impaired.

Williamson et al. [65] presented a probabilistic user interface for binary input devices with high noise levels when reliability can be ensured only 65%–90% of time (e.g., non-invasive EEG). Like Dasher, the interface works by progressively zooming in and draws heavily from information theory. The authors use Hornstein error correcting [1] to increase noise tolerance. Their method builds up certainty for a sequence of user selections before making a decision on all the selections rather than deciding one target at-a-time.

3 INTERFACE DESIGN

We use two interfaces in our study, Nomon and row-column scanning. Here we describe the interfaces in detail and justify our parameter choices in each case — via both simulation studies and collaboration with AAC users and specialists. We also discuss how the COVID-19 pandemic affected our interface and user study.


Scan Speed: 7
First Row Scan Speed: 2
Time remaining: 19:02
Copy the phrase below. Press enter for the next phrase.
Sound ✓
Webcam Switch ✓

```
just did an undertale personality quiz
just did a|
```

Figure 2: Our row-column scanning keyboard interface. The user is being prompted to write "just did an undertale personality quiz" and has written "just did a" so far. The interface progressively highlights rows until the user clicks their switch, and then progressively highlights columns within the selected row until the user clicks once more. Users can also select word completions that are displayed at the top.

3.1 Row-Column Scanning

3.1.1 Background. A row-column scanning (RCS) interface presents the user with a 2D grid of options. For a text entry task, these options are letters and word completions. The system scans through each row at a constant time interval called the scan delay. When a user clicks their switch, the interface selects the currently highlighted row and proceeds to scan through each column. The user clicks again when the column scan highlights their target. The second click makes a selection.

3.1.2 Our implementation. Figure 2 shows our RCS implementation. While there are research and commercial RCS implementations [9, 50, 55, 56], we implemented our own version since our goal was to compare the RCS and Nomon interfaces as directly and fairly as possible. Having our own implementation allows us to use the same word prediction engine in both interfaces and augment both interfaces with similar logging and experimental controls. As noted in [28, 57], word predictions can profoundly impact the entry rate and click load of switch users. As advised by the AAC consultants, we followed The Grid 3 (a popular commercial scanning software) design for our RCS interface.

In both our RCS and Nomon text entry interfaces, the principal options were: character keys (the letters a–z); space; punctuation keys (comma, period, apostrophe, question mark, and exclamation point); and three correction keys — undo (to revert the latest selection), backspace (to delete the current final character), and clear (to clear all text that currently appears).

In the event a user selects a row in error, we follow the recommendation of [54] and set the maximum number of column scans to two complete cycles. After this point, the interface reverts back to row scanning. This procedure stands in contrast to alternative options, such as requiring the user select an option to stop scanning the columns or to reverse the direction of scanning [54].

3.1.3 Keyboard Layout. Proper RCS configuration is critical for fast writing speeds [2, 21–24, 28, 54, 59]. Therefore, we ran a simulation study to determine optimal interface parameters in our RCS implementation; see the supplemental materials for full details. We then verified our results with the recommendations of the previous literature. Namely, we considered a maximum number of word completions to display at any one time $W_{\max} \in \{1, 2, \ldots, 18\}$. We considered whether to display word completions at the top or bottom of the interface. We also considered whether to sort character options alphabetically or by frequency (with more-common letters in English near the top left of the grid to reduce scan time to reach them). Optimizing text-entry rate in our simulations led us to choose to arrange letters in frequency order and to include seven word completions arranged by decreasing probability in the top row. The frequency arrangement coincides with the recommendation of [54, 59]. Our word completion arrangement matches the recommendation of [24]. The resulting grid was 8×7 in size with at most 42 options.

3.1.4 User-adjustable parameters. As is common in RCS implementations [24], users could control two timing parameters: the scan time and the extra delay. The scan time s is how long the interface highlights an individual row or column. We set $s = 2e^{-j/14}$ seconds for $j \in \{0, 1, \ldots, 20\}$. That is, smaller values of j correspond to longer scanning delays with s ranging between $[0.48, 2.00]$ seconds.

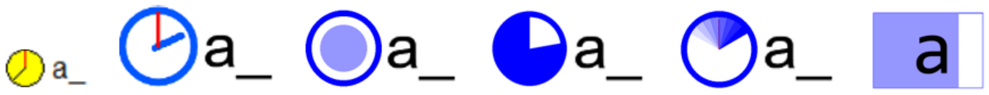


Figure 3: From left to right: (1) The clock used in the original Nomon interface [7], (2) the clock design used in this study, (3) the filling ball clock, (4) the "pac-man" clock, (5) the radar clock, (6) the progress bar. Clock sizes are relative to the clock from the original interface on a 2560×1600 pixel display.

The extra delay d is added to the scan time for the first row and column. We set $d = 0.15(10 - k)$ seconds for $k \in \{0, 1, \dots, 10\}$. Therefore, $d \in [0, 1.5]$ seconds. Smaller values of k correspond to longer extra delays and $k = 10$ corresponds to no extra delay. Participants started with the slowest settings of $j, k = 0$ and were allowed to increase or decrease either j or k or both by 1 between phrases.

3.2 Nomon

3.2.1 Background. In Nomon, every option in the interface has a clock next to it (Figure 1). Each clock has a unique phase, and the minute hands of all clocks rotate at a constant, shared speed. A user needs to look at only one clock to select its corresponding target. RCS and other methods are potentially more taxing in that they require a user to shift visual attention between different parts of the screen. The Nomon user is instructed to click when their target clock's hand passes the red "noon" line. After each click, the clock hands change phase. The phase change is chosen to separate the clock phases of the most probable next targets from one another. The user repeatedly clicks, each time aiming for when the minute hand passes noon, until their target is selected. The number of clicks required to select a target is dependent on the precision of the user and on how probable the target is. In a text entry application that makes use of a language model, an experienced user can select targets in around two clicks [7]. A video demonstration of how typing with the Nomon interface works can be found in our supplemental materials.

3.2.2 End-User and AAC Consultant Involvement in the Design Process. Throughout the process of redesigning the Nomon interface, we consulted with two charities specializing in individuals with severe motor impairments: SpecialEffect and the Ace Centre. We received feedback from ten of the SpecialEffect staff members and one consultant from the Ace Centre. In addition, a single-switch user affiliated with SpecialEffect gave us feedback on usability and accessibility. All the feedback played a major role in our design choices, including: color options (e.g., to help prevent seizures or migraines), clock design, font choice, text contrast, the addition of a tutorial and calibration phase, and improved visual/audio selection feedback.

Our AAC-charity consultants noted that it can be cumbersome and error-prone to initialize parameters before using an AAC interface. In the case of Nomon, the click-time distribution estimate itself requires initialization. And we know that click-time distributions can vary considerably across users; see Section 3.5. So the fixed initialization of the click-time distribution estimate in the original Nomon would generally be misspecified for a new user. This discrepancy could necessitate impractical or costly manual intervention from carers or users. We therefore introduced a calibration phase to initialize the estimated click-time distribution of a user before they start using Nomon.

At our consultants' suggestion, we also considered alternative indicators besides clocks; namely, from right to left in Figure 3, progress bars, clocks with radar trails, a "pac-man" filling clock, and filling circles. Based on the consultants' feedback, we settled on a larger clock design with thicker borders and higher contrast, and a larger, bolder font. These changes are consistent with modifications to Nomon to increase usability reported in [33].

3.2.3 Simulating a Nomon User.

Motivation. In the Nomon keyboard interface, two parameters control the presentation of word completions on the screen: W_c, the number of word completions in each character's box; and $W_{\max}$, the total number of word completions allowed across all characters. In the original study of Nomon, W_c was set to 3 words per character as it was the maximum number that could fit on the screen, and $W_{\max}$ was left uncapped [6]. Given prior success in optimizing parameters in scanning systems via simulation [20, 40, 54], we investigated optimizing these two parameters in Nomon. We developed a model of a switch user that simulates text composition in the Nomon keyboard. We then applied this model to generate synthetic user selection data that could predict performance at various parameter configurations.

We emphasize that Nomon's internal model (i.e., the model controlling when a selection is made, how the clock phases are set, etc.) is distinct from our user simulation. To emphasize the difference, note that Nomon can operate, with its own internal model, without employing a user simulation to optimize these two parameters; for instance, the parameters could instead be given default values. Conversely, we can (and do) employ a user simulation to optimize the parameters of RCS even though RCS does not include an internal model to control its operation like Nomon does.

Input Error Model: The Click-Time Distribution. A click-time distribution is a likelihood estimated by the Nomon interface of when a user clicks relative to noon. The histogram in Figure 4 visualizes this distribution for a particular Nomon user. Nomon estimates this distribution as part of its probabilistic selection mechanism [6]. After a series of clicks from the user, the posterior probability of each clock can be calculated through Bayes' theorem. A clock is selected in Nomon if its posterior probability is sufficiently high [6]. As a consequence, if a user's click-time distribution is narrow, they will generally be able to select a clock with few clicks. If their distribution is wide, it will take more clicks to select an option. Precise users will be able to select clocks quickly, and imprecise users

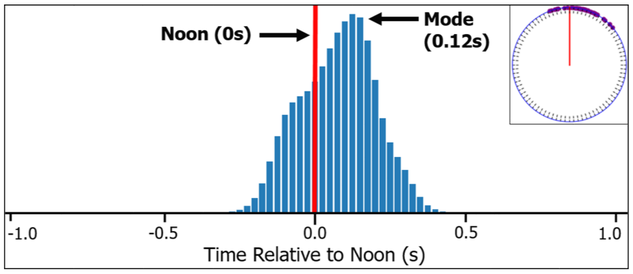


Figure 4: Visualization of a click-time distribution. A click-time distribution is a likelihood estimated by Nomon of when a user clicks relative to noon. The histogram above can be thought of as unraveled from the clock in the top right. The purple points around this clock are the samples (timings of clicks relative to noon) used to construct this likelihood estimate. The mode of this histogram occurs 0.12 seconds later than overlap of the moving hand with noon.

will require more clicks to to reduce the chance of an erroneous selection.

User Model. Our user model interacts directly with a running instance of Nomon, receiving information on the available clocks and their hour-hand locations, and outputting switch activations. Given a phrase, the user model attempts to type the phrase by targeting and selecting character clocks and word completion clocks (if available) in the interface. It selects these target clocks by calculating the time until the target clock is at noon and simulating a switch event at that time.

However, real users rarely click exactly at noon. We added input noise to the user model by sampling from an experienced user's fixed click-time distribution. A click-time distribution sample was added as an offset to when the user model simulated a switch event.

We then generated synthetic user data by running our user model on the corpus of phrases used for the text entry task in our user study (Section 4.3.1). We repeated these simulations while varying the two parameters that control the word prediction layout.

Keyboard Layout and Simulation Results. Figure 1 shows the Nomon interface we used for our text entry study. We divided the layout into a 6×5 grid of principal options, with characters alphabetically arranged. Each character option could have a maximum of 3 word completions displayed next to it in the grid. Thus, there could be up to $3 \cdot 26$ word completions in total.

We ran simulations of our user model to choose the number of word completions per character ($W_c \in \{1, 2, 3\}$) and the number of total word completions ($W_{\max} \in \{1, 2, \ldots, W_c \cdot 26\}$) to display. Our simulations in Figure 5 showed $W_c = 3$ and $W_{\max} = 17$ achieved the maximal text entry rate while keeping click load (number of switch activations per selection) to a minimum. More selection options can increase the click load depending on the width of the user's click-time distribution. The addition of word completions past 17 did not have a noticeable effect on entry rate gains; however, each additional word increased the click load. Therefore, we chose the lowest value of $W_{\max}$ that achieved the maximal entry rate. As a result, there were at most 52 total options present on the screen at any one time.

Note that our results indicate that Nomon can handle more word completions than an RCS interface without seeing a drop in performance. We expected this behavior a priori: adding word completions increases the number of options RCS must cycle through before arriving at other options; Nomon is not so directly affected — though greatly increasing the number of options in Nomon can ultimately require more clicks from the user to disambiguate. Further, our choice of fewer word predictions for the RCS interface was guided by our simulations and corroborated by existing literature on RCS optimization (detailed in Section 3.1.3). However, no such work has determined optimal values for these parameters in Nomon. The original Nomon interface included up to $3 \cdot 26 = 78$ word completions [7] — a number that our simulations suggest is too many.

3.2.4 User-adjustable parameter. Users could set the rotation speed of the clocks T to values of $T = 4e^{-l/10}$ seconds for $l \in \{0, 1 \ldots 20\}$. That is, smaller values of l correspond to slower rotation, with $T \in [0.5, 4]$ seconds. Participants started with the slowest setting of $l = 0$ and were allowed to increase or decrease l by 1 between phrases.

3.3 Open-Source and Browser-based Interface

We originally designed our interfaces to run as standalone applications installed on a user's computer. We had started conducting an in-person lab study using these applications. But due to the COVID-19 pandemic, we suspended this study in March 2020. In the following months, we ported our applications to web interfaces running in a user's browser via HTML and JavaScript. Though it was a setback, we believe our implementations are now much more accessible than before; anyone can run our implementations in a standard browser without the need for local software installation. We encourage the reader to try it out at https://nomon.app and share any feedback. Our code for the Nomon application is open source and can be accessed via the link. At the same link, we also provide the code used to run the text-entry simulations with Nomon described in Section 3.2.3 and the simulations with RCS described in the supplemental materials.

3.4 Word Predictions and Character Probabilities

In text entry applications, Nomon and RCS can make use of word predictions based on the text written so far. If the user has not started typing a word, Nomon and RCS predict the most likely words based on the previous words. If the user has started entering a word, the interfaces predict the most probable words that complete the current word. In calculating the clock phases, Nomon also uses the probability distribution over the next character given the current text.

To take advantage of this language information, our interfaces need to query language models at the word and character levels. Our language models were trained on data from a crawl of the web, social media, and movie subtitles. Our goal was to create models that approximate the sort of text AAC users might need for written

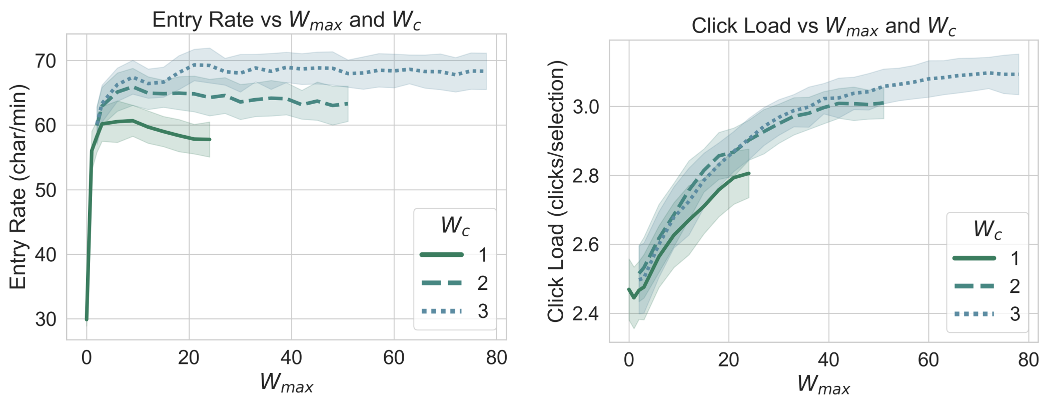


Figure 5: User model simulation results for entry rate (left) and click load (right) for the Nomon keyboard. We ran simulations across values of $W_c \in \{1, 2, 3\}$ (word completions per character) and $W_{\mathbf{max}} \in \{1, 2, \ldots, W_c \cdot 26\}$ (total word completions displayed). Lines show the mean across 150 phrases. Error bands show bootstrapped 95% confidence intervals. Samples were drawn from the click-time distribution of an experienced user — hence the relatively high text-entry rate. Clicks per selection were, as expected, in line with experimental results for the picture selection task in our user study. Note that selections can include a word in this simulation. Participants averaged around 2.5 clicks/selection for the text entry task (with $W_{max} = 17$) and reached 3.5 clicks/selection in the emoji task when there were 60+ options on the screen (which is similar to larger values of W_{max}).

or person-to-person communications. We started with 286 B words of data and used cross-entropy difference selection [41] to filter this down to 8.5 B words of data. Filtering used a set of in-domain language models trained on conversational AAC-like data [61], short email messages, and two-sided dialogues [31]. We trained our 12-gram character model using Witten–Bell smoothing [5] and the 4-gram word model using modified Kneser–Ney smoothing [8]. The character and word language models had a compressed size of 782 MB and 837 MB respectively.[5] We used BerkeleyLM [45] for efficient language model queries.

These language models have a large memory footprint and ranking words can be computationally expensive. Rather than performing these calculations in the browser, we instead built a web API that our interfaces queried. The Nomon and RCS interfaces predict the most likely words from the subset of a 100 K vocabulary that matches the currently entered prefix. Specifically, for this subset of words, we calculated the log probability of the remaining letters of each word, including a trailing space, under the character language model. To each word, we also added the log probability of the word under the word language model. Both language models conditioned on any text written prior to any current partial word.

Since we could not control the latency between users and our language model server, we added caching API endpoints. These caching endpoints allowed the interface to look up all probabilities for all possible next selections by the user, thereby preventing noticeable lag after selecting a character or word. The language model was hosted as an API on a separate Apache Tomcat server with 8 CPUs and 8 GB of RAM. The predictions for a particular API call were computed in parallel to utilize all CPUs. Server load never exceeded 2 participants at a time to prevent lag in presenting predictions.

[5]Specifically we used the large models from https://imagineville.org/software/lm/dec19_char/ and https://imagineville.org/software/lm/dec19/.

3.5 A Single Switch via Webcam Input

Motivation. Individuals with severe motor impairments are challenging to recruit; their time and insight is particularly valuable. We feel ethically that we should thoroughly vet any system before consuming substantial time and effort of single-switch users. In line with this thinking, many preliminary studies on AAC methods include non–switch-using participants. In a survey of 42 studies on AAC software, 21% included non–switch users [19]. Furthermore, when researchers include motor-impaired participants, the sample sizes can be limited. In the same survey, 21% of studies included only a single participant [19]. Non-switch users provide a way to ameliorate noisy data from (often few or one) motor-impaired users and to gain statistical power to distinguish interface performance [17]. Since non–switch users play a prominent role in the study of AAC software, we want to ensure they approximate the target population as well as possible. For our studies, we designed a single switch based on input from a webcam. In particular, we chose our webcam switch over a button press in order to better approximate motor-impaired reaction times, as we describe below.

Switch details. Our webcam switch tracks the movement of a user's face and displays their current face location in the form of an orange box. Users activate the webcam switch by moving their head in and out of two regions in succession: (1) a reset region and (2) a trigger region (Figure 6). To "click" (i.e., to activate the webcam switch) the user first moves their head so that the orange box intersects the blue reset region, and then moves their head into the green trigger region. The reset box activates when the participant is in a neutral position, and the trigger box activates when the participant moves their head to the other side. Triggering our webcam switch requires a wide motion of the user's torso and therefore decreases reaction time relative to a simple button press.

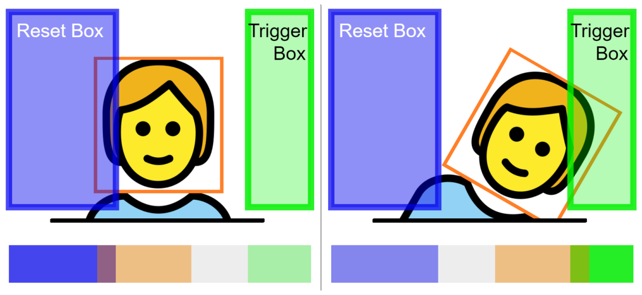


Figure 6: An illustration of our webcam switch, designed to approximate motor-impaired reaction times. To cause a switch click, the user first activates the blue reset box — which is adjusted to their resting position — and then activates the green trigger box — which is adjusted so that they must move their head to the side. The emojis were adapted from [64].

Preliminary justification. The first author made a limited comparison of reaction times using our webcam switch and a tactile button. We found the reaction times using our webcam switch were much closer to observed data of motor-impaired switch operation, collected in [22, 24]. Using data from our own user study, we formally validate this finding; see Section 5.

4 USER STUDY

Here we describe the two tasks in our user study: a text-entry task and a picture-selection task — as well as results for both tasks. We chose an extended study design with 10 sessions to provide insight into experienced use of both interfaces. We conclude that experienced users of Nomon find both tasks faster and easier than row-column scanning.

4.1 Participants

We recruited 13 non–switch-using participants through emails sent to university and community mailing lists. All participants provided written, informed consent. Our experimental protocol was approved by our institutional review board. 8 participants were female and 5 were male. Their ages ranged from 19 to 76 (mean 35, sd 20). 8 were currently attending university, and their locations varied across the United States. None were familiar with either interface or with single-switch text entry software.

In addition, we recruited a single-switch user to trial the Nomon keyboard. They have an advanced form of spinal muscular atrophy and have over 14 years of experience using single-switch scanning. They use an EMG switch and EZ Keys row-column scanning for their daily computer interaction. This switch was used throughout their involvement in this study.

4.2 Procedure

The non–switch participants took part in 10 sessions and paced themselves after the initial session. We instructed them to aim for 1–2 sessions per week, with no more than 1 session per day. Participants took around 8 weeks to finish the study.

In the first session, we explained the purpose of the study and obtained informed consent. We considered this session as practice since participants used both interfaces for less than 5 minutes each. We did not analyze results from this practice session. The first author was present via video conferencing during the first session and second session to introduce the study and answer any questions.

Sessions 2–9 were structured as follows. Participants used the Nomon and RCS interfaces for 20 minutes each to perform the text-entry task described in Section 4.3. We alternated which interface (Nomon or RCS) each participant used first to achieve a near-even split. We had participants alternate which interface they used first from session to session. In the study, we referred to the two interfaces simply as A and B to minimize bias towards Nomon [11].

In sessions 2, 5, and 9, participants completed a questionnaire after using each interface. In sessions 2 and 9, participants also completed a NASA Task Load Index (TLX) [16]. The NASA TLX aims to measure the "load" experienced by a user when performing a task. Sessions 2 and 9 included the sources-of-load section. In session 5, we administered only the magnitude-of-load section. In session 6, participants completed a reaction time task before using either interface; see Section 5 for full details.

In session 10, we had participants perform a picture selection task described in Section 4.4. Participants used each interface for this task for 20 minutes, for a total of 40 minutes. After each method, we administered the NASA TLX (including the sources-of-load) as well as a questionnaire.

4.3 Experiment 1: Text Entry Task

4.3.1 Procedure. In the text-entry task, participants typed as many phrases as possible in a 20-minute time period with each interface. Participants signaled that they were finished transcribing a phrase by pressing the "Enter" key. We drew phrases uniformly at random (without replacement, both within sessions and across sessions) from a set of phrases. Our aim was to choose phrases that were easy to remember and that represent text people might chose to write when not artificially constrained by AAC software. To those ends, we constructed two phrase subsets: (1) an out-of-vocabulary (OOV) phrase subset: a set of phrases containing exactly one word not in the language model (described in Section 3.4) and (2) an in-vocabulary (IV) phrase set: a set of phrases for which all words were in our language model. We derived both phrase subsets from the "challenging phrase set" developed in [60]. These phrases were all manually reviewed in [60] to ensure that they were easy to remember. The IV and OOV subsets had a mean phrase length of 7.15 (sd 1.60) and 7.24 (sd 1.64) words respectively.

Finally, we constructed our full phrase set by mixing the subsets at a ratio of two in-vocabulary phrases for every one out-of-vocabulary phrase. This mixture ensures we test both the word-completion and the general text-entry abilities of both interfaces. While word completions allow much faster text entry in single-switch text-entry systems [23], unusual words can sometimes arise, e.g. individuals' names, places, and abbreviations. The previous study of text entry with Nomon [7] made use of the MacKenzie phrase set [39]. This phrase set has been shown to have a low incidence of OOV words [60]. Further, while the MacKenzie phrases may have contained some OOV words, the study did not explicitly

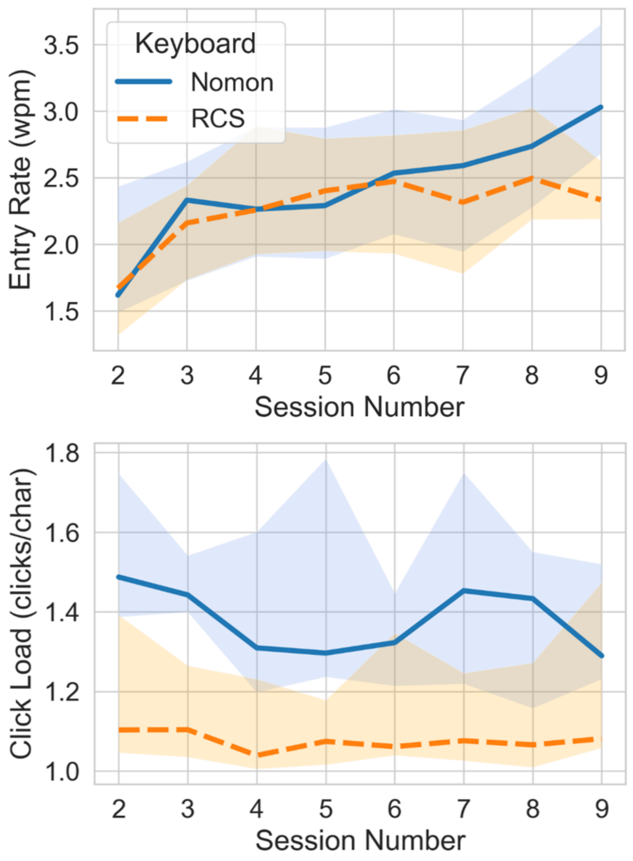


Figure 7: Median text entry rate (*top*) and click load (*bottom*) across all 8 sessions of the text entry task. Error bands show the first and third quartiles of the distributions. The upper curve corresponds to Nomon in both plots.

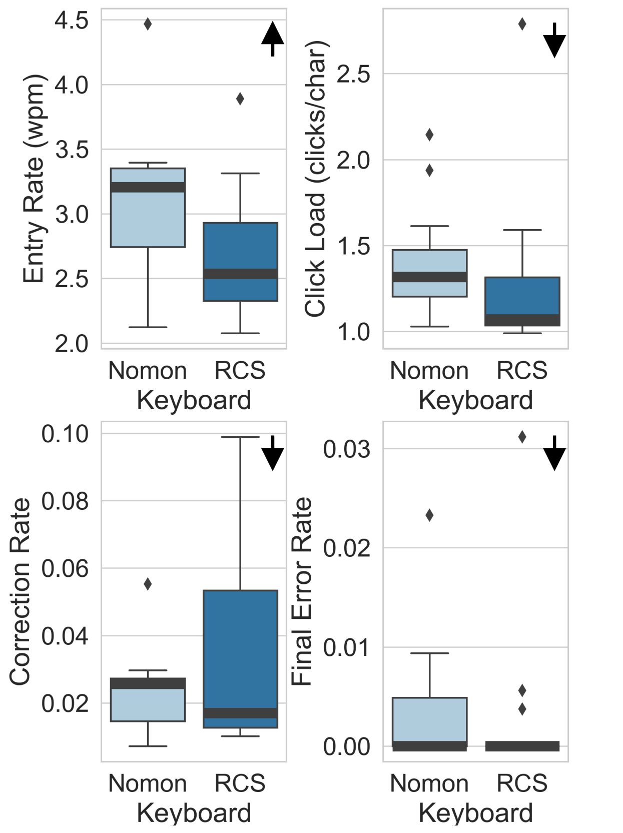


Figure 8: Metrics for sessions 8 and 9. The colored regions are the first to third quartiles of the distributions. The whiskers show the 5th and 95th percentiles of the distributions. Arrows in the top right show the direction of better performance.

examine the effects of these OOV words on text entry performance [6] — whereas we are able to separately examine IV and OOV performance in the present study.

4.3.2 Performance metrics. We calculate *text-entry rate* in words per minute (wpm). We define a word as 5 characters including space. We include only characters present in the final output in our count (i.e., no corrected or undone text). We measured the time interval from the first switch input in each phrase up until the participant signaled they were finished with a phrase.

We define *click load* as clicks per character (cpc) in the final output of a phrase (excluding corrected characters). Activating a switch is often an arduous task for individuals with severe motor impairments; therefore, it is important to consider this metric and not merely the text-entry rate when assessing effectiveness of a single-switch method.

We define *correction rate* as the number of corrections divided by the total number of selections a user required to type a phrase. A correction is a selection of any of the Undo, Backspace, or Clear options. The correction rate gives a measure of how often a user made a mistake when typing.

We define *final error rate* as the Levenshtein distance between the target phrase and a participant's final text output divided by the length of the target phrase. The Levenshtein distance measures how many character insertions, deletions, or substitutions are required to go from one string to another.

4.3.3 Results.

Expert Performance. We are interested primarily in comparing the performance of expert users; therefore, we restrict our analyses to data aggregated over the final two sessions (eight and nine). We performed a Shapiro–Wilk test for normality in the paired samples across the two interfaces. We found the normality assumption was violated for click load ($W = 0.667, p < 0.001$) and final error rate ($W = 0.479, p < 0.001$). Where normality could be assumed, we used a dependent t-test (denoted as t); otherwise we used a Wilcoxon signed-rank test. Table 1 shows numerical results and the corresponding significance tests.

Figure 8 displays the aggregate text entry metrics for sessions 8 and 9, for all participants. Participants typed 1.15 times faster using Nomon over RCS; however, they had a slightly higher click

Metric	Nomon		RCS		Statistical Test		
	mean	median	mean	median			
Entry Rate (wpm)	3.10	3.21	2.69	2.53	$t(12) = 2.88$	$r = 0.639$	$\boldsymbol{p} = \mathbf{0.014}$
Click Load (cpc)	1.39	1.32	1.27	1.07	Wilcoxon	$r = 0.187$	$\boldsymbol{p} = \mathbf{0.046}$
Correction Rate	0.0215	0.0257	0.0354	0.0170	$t(12) = -1.76$	$r = 0.452$	$p = 0.104$
Final Error Rate	0.0038	0.000	0.0031	0.000	Wilcoxon	$r = 0.11$	$p = 0.499$
IV Entry Rate (wpm)	3.48	3.37	3.07	2.92	$t(12) = 2.25$	$r = 0.410$	$\boldsymbol{p} = \mathbf{0.033}$
OOV Entry Rate (wpm)	2.32	2.43	1.90	1.81	$t(12) = 3.94$	$r = 0.620$	$\boldsymbol{p} < \mathbf{0.001}$
IV Correction Rate	0.019	0.016	0.029	0.016	$t(12) = 2.25$	$r = 0.319$	$p = 0.174$
OOV Correction Rate	0.026	0.022	0.051	0.035	$t(12) = -2.41$	$r = 0.435$	$\boldsymbol{p} = \mathbf{0.023}$
NASA TLX, session 2	38.3	41.4	35.3	34.6	$t(12) = 0.65$	$r = 0.161$	$p = 0.525$
NASA TLX, session 5	32.9	33.4	32.5	33.9	$t(12) = 0.65$	$r = 0.160$	$p = 0.905$
NASA TLX, session 9	27.0	27.0	33.4	33.4	$t(12) = 0.12$	$r = 0.575$	$\boldsymbol{p} = \mathbf{0.032}$

Table 1: Mean and median result values and statistical tests for the text-entry task in the user study. Results are for sessions 8 and 9. Metrics in bold were significant.

load using Nomon compared to RCS. The first published Nomon study [7] found that participants typed 1.35 times faster using Nomon over RCS. The discrepancy in results might be attributed to the noise we have introduced via the webcam switch, as we observed larger error bars compared to [7]. As in Figure 4 in [7], the RCS entry rate here seemed to plateau before the entry rate of Nomon. In our study, the RCS plateau is reached in a later session, which might be expected due to the learning curve associated with a more noisy switch. We found no significant difference in correction rates or final error rates between the interfaces.

Switch User Performance. We recruited a single-switch user to complete the text entry task using the Nomon interface. We do not compare their performance between Nomon and RCS directly, as they use an RCS system daily and it would not lend a fair comparison. Rather, we compare their performance with Nomon to that of the hindered, non–switch users.

The participant regularly uses the RCS software EZ Keys with a 100 millisecond scan speed. They have abbreviation expansion and custom, task-specific word completions to speed text entry. Utilizing this optimized setup, they have self-reported to type at an impressive 13 wpm. We note the fast scan speed at which this switch user regularly uses an RCS interface. The switch user's proficiency with their switch allowed them to use Nomon with a rotation period of 0.76 seconds — a considerably faster period than the average 3.35 seconds of the non–switch-using participants in this study. While this level of switch accuracy and speed may not be representative of a majority of single-switch users, this particular switch user's proficiency and associated quick communication speed was why we felt comfortable having this user pilot test our study methods. The switch user has provided us with insights into our study and software design that will prove invaluable in our following work with more diverse members of the target population.

We show the switch user's results alongside those of the non–switch users from session 6 (after an equivalent 80 minutes of practice) in Figure 9. The switch user's sessions ran identically to the text-entry-task sessions for the non–switch-users. The correction rate and final error rate of the switch user both fell within those of the non–switch users. However, the switch user had a considerably

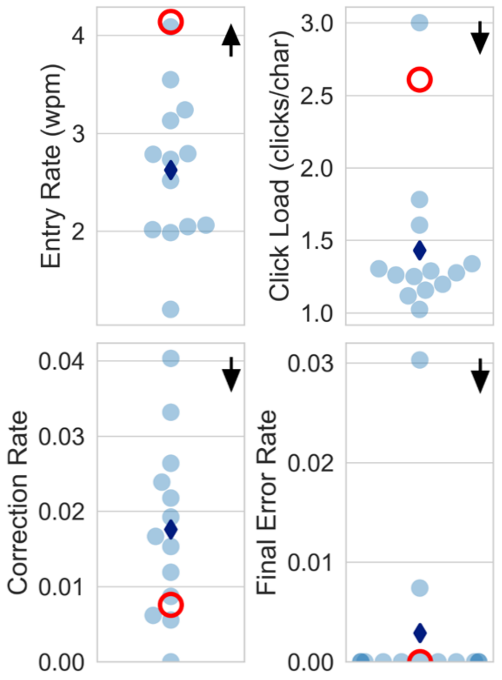


Figure 9: Comparison of text entry metrics for the non–switch users and the motor-impaired user after 80 minutes of prior practice (equivalent to session 6 for the non–switch users). Each filled light-blue circle represents a single non–switch user, and the population mean is given by a filled dark-blue diamond. Red circles represent the motor impaired user. Arrows in the top right show the direction of better performance; e.g. we prefer a higher text-entry rate.

higher text entry rate (1.5 times faster; 4.14 wpm) and click load (1.8 times larger; 2.61 cpc). The switch user's shorter rotation period (4.4 times faster) than the non–switch users may account for this increase in both entry rate and click load. While a shorter rotation

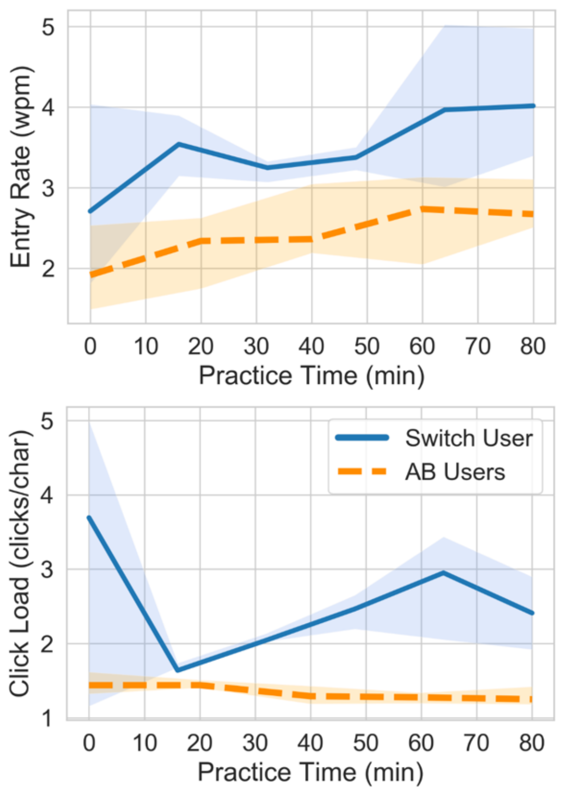


Figure 10: Median text entry rate (*top*) and click load (*bottom*) for the switch user (*blue, solid line*) and hindered, non–switch users (*orange, dashed line*) in Nomon. The x-axis shows how long users practiced with Nomon. At any time point, we plot a summary of the switch user's distribution over performance on individual phrases, while we plot a summary of the non–switch users' performances across the 13 participants. The error bands show the first and third quartiles of the relevant distribution.

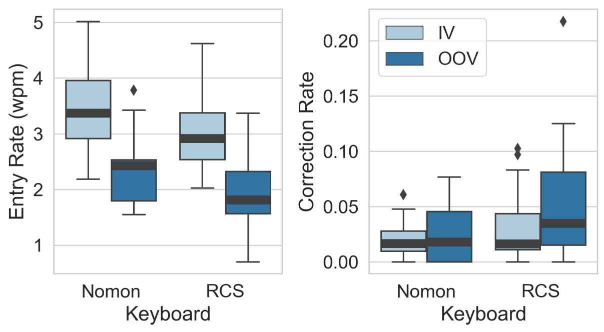


Figure 11: Entry rates and error rates for Nomon and RCS for in-vocabulary (IV) and out-of-vocabulary (OOV) phrases. In each paired comparison, IV appears to the left of OOV. Results are from sessions 8 and 9. The colored regions are the first to third quartiles of the distributions. The whiskers show the 5th and 95th percentiles of the distributions.

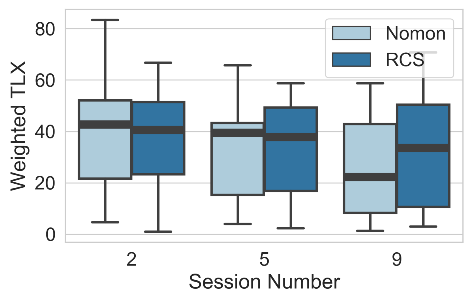


Figure 12: Results of the NASA Task Load Index administered following session 2, 5, and 9. In each paired comparison (per session), Nomon appears to the left of RCS.

period may have allowed the switch user to minimize dead-time and thus increase their entry rate, the shorter period may have caused them to be less precise and require more clicks per selection.

Further, we compare the learning curves of the switch user and the non–switch users with Nomon in Figure 10. The switch user had a consistently higher entry rate compared to the non–switch users at identical practice times with Nomon. The switch user's performance also increased with practice, much like the non–switch-using participants. However, the click load of the switch user varied much more throughout their practice sessions. After around 20 minutes of practice, the switch user reached their lowest click load of 1.6 cpc. The click load then continued to increase throughout the remaining sessions. This minimal click load occurred when the switch user had a rotation period of 1.62 seconds, with the larger click loads occurring as the user progressively shortened the rotation period.

Challenging Text Entry. The combination of IV and OOV phrases allows us to test both the word completion and general text entry abilities of the interfaces. As evident in Figure 11, we found that the addition of a single OOV word in a phrase can considerably lower text entry rates in both interfaces. This result is consistent with work investigating the effect of OOV words in mobile text entry in [60]. Users were able to better handle these OOV words using Nomon. They typed OOV phrases 1.22 times faster and with with half as many corrections using Nomon over RCS. This difference suggests Nomon may be better suited to less predictable text composition than RCS. Indeed, Nomon's probabilistic selection mechanism does not seem to favor word completions for quick selection as dramatically as RCS (which dedicates the first scan row to word predictions that are useless for OOV words). Furthermore, users also performed better with Nomon on IV phrases, though to a lesser extent; they typed 1.13 times faster using Nomon, but had no significant difference in correction rate.

4.3.4 Subjective Feedback. We assessed user experience with questionnaires for each interface in the second, fifth, and ninth sessions. Participants indicated their agreement with a series of statements on a scale from 1 to 5, with 1 indicating "strongly disagree" and 5 indicating "strongly agree." The distribution of responses across the sessions appears in Figure 13.

As evident in Figure 13, participants increasingly felt they typed faster, more accurately, and with fewer errors as they used Nomon more. Conversely, participants generally rated RCS the same in these three areas throughout the study. Figure 12 shows no notable difference in the overall NASA TLX scores between RCS and Nomon in sessions two and five. However, in the final session, participants rated Nomon as having a lower task load ($t(12) = 0.12, p < 0.032$). This result further indicates that participants increasingly found Nomon easier to use with practice.

At the conclusion of the text entry task, we asked participants to choose between the two interfaces. 12 out of 13 participants indicated that they preferred typing with Nomon over RCS. Common reasons for this choice were that Nomon is "more forgiving with errors," there is "more flexibility" and "agency" in the selection process, and "less downtime waiting for scanning."

We also received feedback from the switch user on their experience using Nomon. They noted, "I observed more word predictions showing up as choices. This is where I see some real potential for increased typing rate (in terms of words per minute). Nomon is distinctly different from traditional scanning and may offer an easier path to higher text entry rates." The full responses from our participants can be found in the supplemental materials.

4.4 Experiment 2: Picture Selection Task

Text entry is a particularly important task for AAC users, so our user study focused on this task for most sessions. But there are many tasks of interest beyond text entry. Nomon has the advantage over RCS of being adaptable to tasks that need not fit into a grid. However, there exist tasks beyond text entry for which the two interfaces can be compared. In particular, when users choose among a large set of files on their computer, photos on a photo-sharing website, or products at an online vendor, these items can be arranged in a grid. Our aim was to encapsulate such a task and compare Nomon and RCS. We chose emojis as our set of options since we thought they would be easily recognizable by users and engaging for our participants.

For this experiment, we adapted the Nomon and RCS interfaces to include 60 emojis (Figure 14). The core functionality behind both interfaces remained the same. The interfaces highlighted the current target to avoid participants spending time searching through the options. This search time varies widely depending on how quickly a participant can find the next target; therefore including it in entry-rate calculations would introduce unnecessary variance. We chose 60 emojis because 60 was close to the maximum number of objects that could fit on the screens of both interfaces.

We expect Nomon to excel at this task. Under an uninformed prior (as in this task), previous work has shown that the number of switch clicks required to select a target in Nomon scales logarithmically with the number of options [6]. With a constant rotation speed, the time required for selection (excluding reaction time and the time spent searching for the desired option) should scale similarly. By contrast, the mean number of scans to select an option in an RCS interface scales with the square root of the number of options n; the user must make an average of $\sqrt{n}/2$ row scans and then $\sqrt{n}/2$ column scans (if options are arranged in a square grid).

4.4.1 *Procedure and Performance Metrics.*

We used the final session to test this alternative task. We expected users would have ample experience with both interfaces by the final session and therefore would not require multiple sessions to adjust to the picture-selection task. In lieu of English phrases, we asked participants to write sequences of five emojis at a time. We computed four metrics: entry rate (selections per minute), click load (clicks per selection), correction rate, and final error rate.

4.4.2 *Results.*

A Shapiro–Wilk test for normality in the paired samples found this assumption was violated for final error rate ($W = 0.479, p < 0.001$). We used a dependent t-test (denoted as t) where normality could be assumed; otherwise we used a Wilcoxon signed-rank test. Table 2 shows numerical results and the corresponding significance tests.

Figure 15 shows user performance in the picture selection task in session 10. The benefits of Nomon were even more pronounced in picture selection compared to text entry. Participants selected targets substantially and significantly faster using Nomon — an average of 36% faster. This increase in entry rate comes with a trade-off in click load. Participants had a higher click load of 3.50 clicks per selection using Nomon, compared to 2.23 clicks per selection using RCS. However, we expected this increase given the conjectured logarithmic scaling in the number of required switch clicks [6]. Participants also made fewer corrections per selection using Nomon — 1.1% with Nomon versus 2.6% with RCS. We found no significant difference in final error rates between the interfaces.

5 REACTION TIME STUDY

In Section 3.5, we described the webcam switch we employ in our user study. In this section, we validate our claim that this switch yields a useful approximation of motor-impaired single-switch reaction times with non–switch user inputs.

Quantities to approximate. There are two key quantities [53] for single-switch operation that we aim to approximate:

- Simple reaction time (SRT) — SRT is the time difference between the introduction of a stimulus to a user and their subsequent response.
- Double click time (DCT) — DCT is the amount of time between a user's successive switch activations. DCT measures how quickly a user can click their switch again after they have just clicked it.

SRT and DCT dictate how quickly users can operate single-switch software. E.g., if the scan delay or rotation time is too fast compared to a user's typical SRT, they may find the software unusable [53]. RCS requires users to click their switch twice in immediate succession to select targets in the first column; if the scan delay is too fast compared to a user's typical DCT, they will be unable to select these targets.

Single-switch user and non–switch user data. Dr. Heidi Koester graciously provided data on the SRTs and DCTs of non–switch-using and single-switch-using individuals that she and her colleagues collected — namely, 10 motor-impaired users in [24] and

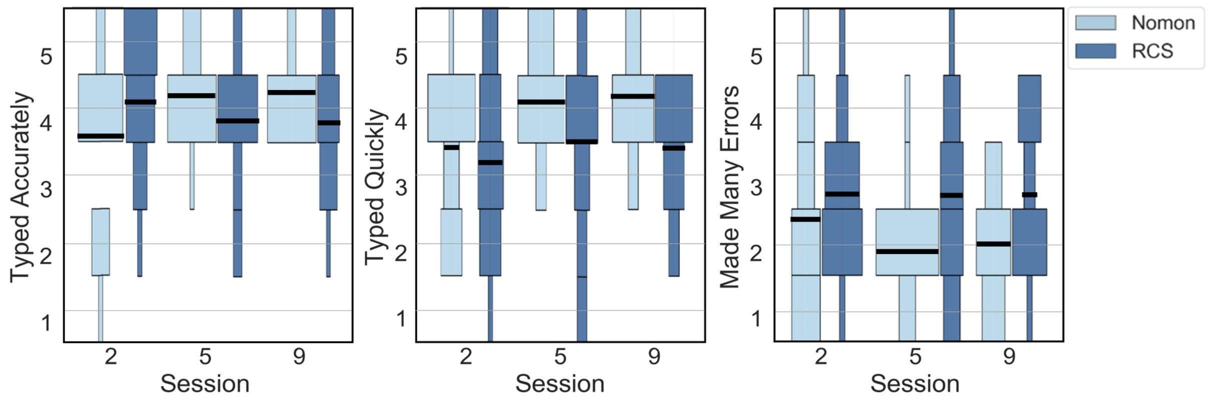


Figure 13: Letter-value plot of the results of questionnaires administered in the beginning, middle, and end of the user study. The relative width of the color regions denote how many participants gave a statement that Likert score. Statements were presented in the form "In this part of the study, I felt that I typed quickly." Participants responded on a scale from 1 (strongly disagree) to 5 (strongly agree). Means are represented by horizontal black lines. In each paired comparison (per session), RCS appears to the right of Nomon. Arrows show the direction of better ratings for each prompt.

Metric	Nomon		RCS		Statistical Test		
	mean	median	mean	median			
Entry Rate (selections/min)	6.64	6.65	4.88	4.59	$t(12) = 5.24$	$r = 0.834$	$\boldsymbol{p < 0.001}$
Click Load (cicks/selection)	3.50	3.25	2.22	2.23	$t(12) = 7.73$	$r = 0.912$	$\boldsymbol{p < 0.001}$
Correction Rate	0.011	0.0095	0.026	0.0238	$t(12) = -2.45$	$r = 0.577$	$\boldsymbol{p = 0.031}$
Final Error Rate	0.0025	0.000	0.0043	0.000	Wilcoxon	$r = 0.072$	$p = 0.893$
NASA TLX, session 10	27.2	27.1	27.7	27.7	$t(12) = -0.22$	$r = 0.064$	$p = 0.893$

Table 2: Mean and median result values and statistical tests for the picture-selection task. Metrics in bold were significant.

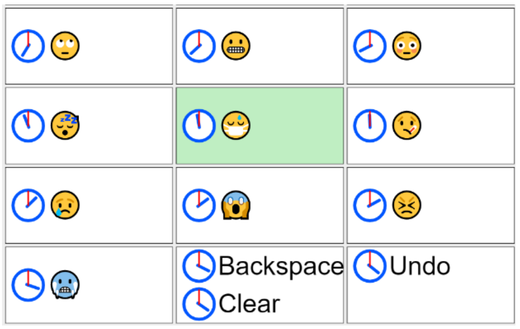


Figure 14: A portion of the Nomon interface for the picture-selection task. We adapted the RCS interface for the picture-selection task in a similar way.

10 motor-impaired users and 8 non–switch users in [23]. While this data may not fully represent the diversity of motor-impaired switch users, it provides insight into the extent to which unhindered, non–switch-using participants can be unrepresentative of the motor-impaired population. Further, the data shows that by hindering non–switch users with our webcam switch method, we can better represent some subset of the motor-impaired population in two key metrics related to single-switch use (SRT and DCT).

Procedure. We collected our data as an additional task added before the start of the sixth session of our user study. Participants used a web interface that first had them use our webcam switch and, secondly, their keyboard spacebar as a switch. Following [24], for each switch, we had the screen flash 30 different times at random intervals. We instructed participants to click their switch twice in quick succession after they saw the screen flash. For each switch method, we recorded 30 trials and calculated the participant's average SRT and DCT. These averages are visualized in the histograms in the bottom row of Figure 16.

5.1 Results

In the top row of Figure 16, we see that single-switch users with severe motor impairments generally have an SRT and DCT much longer than non–switch users. There is also a wide variance among the motor-impaired population, with some individuals much faster than the mean, and some much slower.

Figure 16 shows that our webcam switch yields SRT and DCT values that are considerably more in line with those of the motor-impaired target population — as compared to a spacebar switch. The webcam switch lowers the SRT of the participants from 350 ms (with the spacebar) to 1050 ms. By comparison, the mean SRTs for the non–switch using and single-switch using populations are 350 ms and 820 ms, respectively. Similarly, the webcam switch lowers

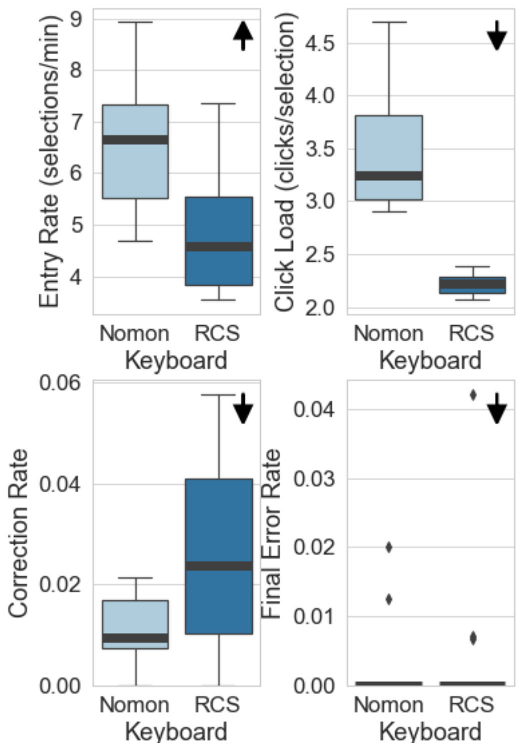


Figure 15: Performance metrics for the picture selection task in session 10. The colored regions are the first to third quartiles of the distributions. The whiskers show the 5th and 95th percentiles of the distributions. Arrows in the top right show the direction of better performance; e.g., we prefer a higher entry rate.

the participants' mean DCT from 180 ms to 1400 ms. These DCTs are consistent with those from the non–switch using and single-switch using populations of 290 ms and 1460 ms. We conclude that our webcam switch technique substantially lowered both SRT and DCT to levels consistent with data from single-switch users with motor-impairments.

6 DISCUSSION

We investigated the effectiveness of Nomon as a method of single-switch communication. We evaluated the performance of Nomon over multiple sessions compared to the widely used row-column scanning method. In a text-entry task, participants typed 15% faster using Nomon. However, they experienced a 10% higher click load with Nomon. This higher click load could be problematic for users where switch activation is tiring.

We are exploring ideas to mitigate this higher click load. One such idea is to use information from an eye gaze tracker, as users will undoubtedly be gazing towards the clock they are trying to select. Interestingly, the switch user who trialed Nomon commented that they "notice[d] a sense of direct selection [with Nomon] (though technically it is not) akin to eye gaze interfaces. One important difference is that I did NOT experience the same eye strain/fatigue often associated with eye gaze mouse pointer navigation."

Separately, we posit that it may be possible to allow just one click per letter for predictable words. Currently Nomon requires each individual character to pass a probability threshold before committing to that character. We believe we could postpone committing to any text until the end of a word (similar to how auto-correction works on a touchscreen keyboard). With only a noisy switch as input, designing how users signal the end of a word, correct errors, and enter difficult words would be challenging — but would constitute interesting future work.

Participants continued improving with Nomon even in the final session, while they appear to plateau with RCS after session 5. Furthermore, participants found typing easier and faster using Nomon in the final sessions. 12 out of 13 participants indicated that Nomon was their preferred method of text entry. We had hoped eight text-entry sessions would be enough for Nomon performance to plateau, but users continued to improve even in our final session. Our results suggests a longer study may be necessary to fully explore Nomon's potential, especially when evaluating with motor-impaired users.

To our knowledge, our study is the first to investigate single-switch input of text containing difficult out-of-vocabulary (OOV) words. When selecting OOV words, the word language model is not active but the character language model still provides a non-trivial prior over common sequences of characters. We found Nomon significantly reduced the need to perform corrections and significantly increased entry rate on OOV phrases. This advantage is important since error correction can be a frustrating process, especially using a single switch. Our interfaces limited word predictions to a vocabulary of 100 K words. We think further improvements in Nomon's efficacy for OOV words may be possible by expanding the prediction engine's vocabulary to a larger word list when the set of predictions becomes sparse or empty. This word list could be created from timely online data sources (e.g., Twitter) and predictions ranked via a language model with a subword vocabulary and trained on enormous amounts of data (e.g., GPT-2 [46]). Our participants also suggested other improvements such as increasing the probability of the undo clock, and removing word predictions that were not selected to free up space for other words.

We explored applications beyond text entry with a picture selection task. The picture-selection task gives the user a large number of options with a uniform prior. Here, the benefits of Nomon were more pronounced as participants selected options 35% faster and with 63% fewer errors. On the other hand, participants had a 53% higher click load; this increase in click load seems to be fundamental to Nomon's flexible selection scheme, where the number of switch clicks required for selection should scale logarithmically with the number of options. These results are promising for future work using Nomon in applications beyond text entry. In particular, it would be interesting to explore tasks that can leverage a prior over targets learned from individual users (e.g., the sequence of links clicked in an application or the control of home IoT devices).

To aid our studies above, we designed and validated a webcam-based switch technique for better approximating motor-impaired operation of a single switch with non–switch-using participants. We found that the simple reaction times (SRTs) and double click times (DCTs) of non–switch users with a physical button were

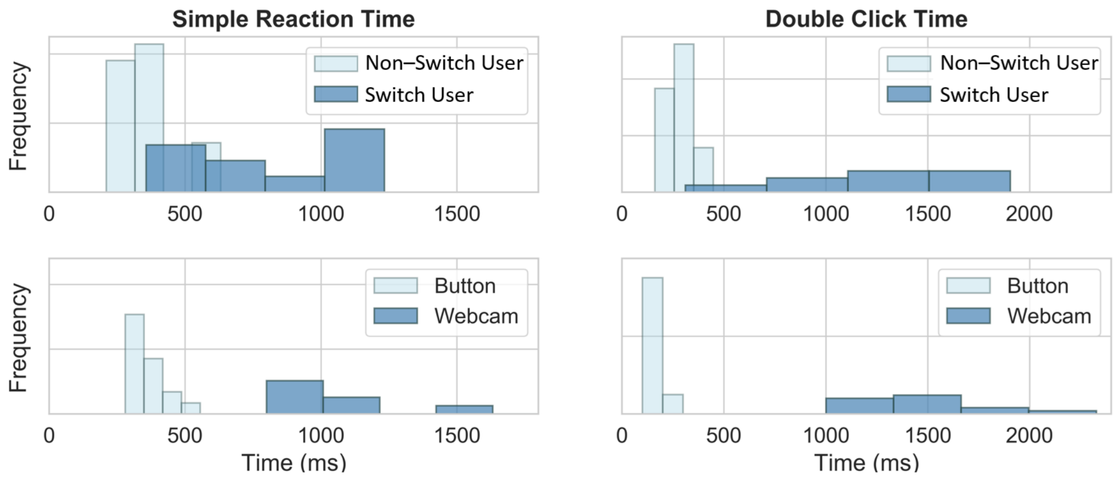


Figure 16: Comparison of the SRTs and DCTs of the two switch methods to those of the non–switch-using and single-switch-using populations. On the top row is the data from [22, 24], with the non–switch users in light blue (appearing left in each plot) and the single-switch users in dark blue (appearing right in each plot). The bottom row contains the data we collected with the button and webcam switches from our non–switch-using participants. The button histogram is in light blue (appears left in each plot), and the webcam histogram is in dark blue (appears right in each plot). SRTs are in the left column and DCTs in the right. Each data point is the average value across switch clicks of a particular participant using a particular switch method; we show a histogram over these data points.

unrepresentative of SRTs and DCTs of single-switch users with motor impairments. Our webcam method artificially lowers a non–switch user's reaction times to more closely resemble the reaction times of single-switch users with motor impairments. Using this technique with our participants allowed us to collect data that more closely resembles that of our target population while recruiting non–switch users as participants.

To further evaluate Nomon, we are planning a similar user study to the one reported here but with a group of motor-impaired users.

7 CONCLUSION

To conclude, we made the Nomon interface more accessible through collaboration with switch users and AAC specialists. We further optimized the design of the Nomon interface via computational simulations. We developed a webcam-based technique to simulate the click timing of motor-impaired users. Our user study results alongside our initial trial with a switch user show that Nomon may currently provide accelerated text input for single-switch AAC users. In their final session (after 2.5 hours of practice), users wrote 15% faster using Nomon than with conventional row-column scanning. We found this speedup was even more pronounced when composing challenging text containing out-of-vocabulary words, and when Nomon was used in a picture selection task. Overall, our results show that Nomon may provide a more efficient, and more flexible, method for rate-limited users to control their computer via a single switch.

ACKNOWLEDGMENTS

We thank the two switch users, who provided important feedback and evaluated the Nomon interface. We are grateful to Bill Donegan and Mick Donegan for organizing several sessions with the staff of SpecialEffect, who provided invaluable feedback towards the final design of Nomon. We thank Heidi Koester for sharing data and useful comments. We thank the Ace Centre – including Will Wade – for helpful conversations. This work was supported in part by the Seth Teller Memorial Fund to Advance Technology for People with Disabilities, a Peter J. Eloranta Summer Undergraduate Research Fellowship, the MIT Intelligence Quest, and the NSF under Grant No. IIS-1750193.

REFERENCES

[1] Minkyu Ahn and Sung Chan Jun. 2015. Performance variation in motor imagery brain–computer interface: A brief review. *Journal of Neuroscience Methods* 243 (2015), 103 – 110. https://doi.org/10.1016/j.jneumeth.2015.01.033
[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] Melanie Baljko and Andrew Tam. 2006. Indirect Text Entry Using One or Two Keys. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility* (Portland, Oregon, USA) *(Assets '06)*. Association for Computing Machinery, New York, NY, USA, 18–25. https://doi.org/10.1145/1168987.1168992
[5] Timothy C. Bell, John G. Cleary, and Ian H. Witten. 1990. *Text Compression.* Prentice Hall, NJ.
[6] Tamara Broderick. 2009. Nomon: Efficient communication with a single switch. University of Cambridge. Cambridge, United Kingdom.
[7] Tamara Broderick and David J. C. MacKay. 2009. Fast and Flexible Selection with a Single Switch. *PLoS ONE* 4, 10 (2009).
[8] Stanley F. Chen and Joshua T. Goodman. 1998. *An Empirical Study of Smoothing Techniques for Language Modeling.* Technical Report. Computer Science Group,

Harvard University.

[9] CoughDrop. Accessed September 2020. CoughDrop. https://coughdrop.zendesk.com/hc/en-us/articles/201366669-How-do-I-set-up-scanning-options-in-CoughDrop-

[10] 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.

[11] Nicola Dell, Vidya Vaidyanathan, Indrani Medhi, Edward Cutrell, and William Thies. 2012. "Yours is Better!": Participant Response Bias in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 1321–1330. https://doi.org/10.1145/2207676.2208589

[12] 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

[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] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *P. A. Hancock and N. Meshkati (Eds.) Human Mental Workload* (1988).

[17] Jeffery Higginbotham. 1995. Use of nondisabled subjects in AAC research: Confessions of a research infidel. *Augmentative and Alternative Communication* 11, 1 (1995), 2–5.

[18] Sunjun Kim, Byungjoo Lee, and Antti Oulasvirta. 2018. *Impact Activation Improves Rapid Button Pressing*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3173574.3174145

[19] Heidi H. Koester and Sajay Arthanat. 2018. The design, conduct, and reporting of research on text entry with alternative access interfaces: Recommendations from a systematic review. *Technology and Disability* 30 (2018), 83–95.

[20] 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.

[21] Heidi H. Koester and Simon P. Levine. 1996. Effect of a Word Prediction Feature on User Performance. *Augmentative and Alternative Communication* 12, 3 (1996), 155–168.

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

[23] 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.

[24] 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.

[25] Byungjoo Lee, Sunjun Kim, Antti Oulasvirta, Jong-In Lee, and Eunji Park. 2018. *Moving Target Selection: A Cue Integration Model*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173804

[26] Byungjoo Lee and Antti Oulasvirta. 2016. Modelling Error Rates in Temporal Pointing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1857–1868. https://doi.org/10.1145/2858036.2858143

[27] Injung Lee, Hyunchul Kim, and Byungjoo Lee. 2021. *Automated Playtesting with a Cognitive Model of Sensorimotor Coordination*. Association for Computing Machinery, New York, NY, USA, 4920–4929. https://doi.org/10.1145/3474085.3475429

[28] 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

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

[30] 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.

[31] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

[32] Sebastián Aced López, Fulvio Corno, and Luigi De Russis. 2015. Can We Make Dynamic, Accessible and Fun One-Switch Video Games?. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. 421–422.

[33] 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.

[34] 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.

[35] Sebastián Aced López, Fulvio Corno, and Luigi De Russis. 2016. Clocks, bars and balls: Design and evaluation of alternative gnomon widgets for children with disabilities. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1654–1660.

[36] 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.

[37] 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.

[38] David J. C. MacKay, Chris J. Ball, and Mick Donegan. 2004. Efficient communication with one or two buttons. In *Maximum Entropy and Bayesian Methods (AIP Conference Proceedings, Vol. 735)*. 207–218.

[39] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI EA '03)*. ACM, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971

[40] 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.

[41] Robert C. Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers* (Uppsala, Sweden) *(ACLShort '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 220–224. http://dl.acm.org/citation.cfm?id=1858842.1858883

[42] 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.

[43] 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.

[44] Antti Oulasvirta, Sunjun Kim, and Byungjoo Lee. 2018. *Neuromechanics of a Button Press*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174082

[45] Adam Pauls and Dan Klein. 2011. Faster and Smaller N-gram Language Models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (Portland, Oregon) *(HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 258–267. http://dl.acm.org/citation.cfm?id=2002472.2002506

[46] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language Models are Unsupervised Multitask Learners. (2018). https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[47] Brian Roark, Russell Beckley, Chris Gibbons, and Melanie Fried-Oken. 2013. Huffman scanning: Using language models within fixed-grid keyboard emulation. *Computer Speech & Language* 27, 6 (2013), 1212 – 1234. https://doi.org/10.1016/j.csl.2012.10.006 Special Issue on Speech and Language Processing for Assistive Technology.

[48] Brian Roark, Melanie Fried-Oken, and Chris Gibbons. 2015. Huffman and linear scanning methods with statistical language models. *Augmentative and Alternative Communication* 31, 1 (2015), 37–50.

[49] Daniel Rough, Keith Vertanen, and Per Ola Kristensson. 2014. An Evaluation of Dasher with a High-Performance Language Model as a Gaze Communication Method. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (Como, Italy) *(AVI '14)*. ACM, New York, NY, USA, 169–176.

[50] Saltillo. Accessed September 2020. NovaChat. https://saltillo.com/support/article/scanning-patterns

[51] Claude Shannon. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27 (July, October 1948), 379–423, 623–656.

[52] 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.

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

[54] 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).

[55] Talk To Me Technologies. Accessed September 2020. Proloquo2go. https://www.talktometechnologies.com/pages/proloquo2go

[56] Smartbox Assistive Technology. Accessed September 2020. Grid 3. https://thinksmartbox.com/product/grid-3/

[57] 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.

[58] Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash away! Longitudinal study of fast text entry by Eye Gaze. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (Savannah, Georgia). 19–26.

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

[60] Keith Vertanen, Dylan Gaines, Crystal Fletcher, Alex M. Stanage, Robbie Watling, and Per Ola Kristensson. 2019. VelociWatch: Designing and Evaluating a Virtual Keyboard for the Input of Challenging Text. In *CHI '19: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland).

[61] Keith Vertanen and Per Ola Kristensson. 2011. The Imagination of Crowds: Conversational AAC Language Modeling using Crowdsourcing and Large Data Sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 700–711.

[62] Tonio Wandmacher, Jean-Yves Antoine, Franck Poirier, and Jean-Paul Départe. 2008. SIBYLLE, An Assistive Communication System Adapting to the Context and Its User. *ACM Transactions on Accessible Computing* 1, Article 6, 6:1–6:30 pages. Issue 1.

[63] 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.

[64] Johanna Wellnitz. 2020. Person [svg]. OpenMoji. https://openmoji.org/data/color/svg/1F9D1.svg.

[65] 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.

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