Optimizing keyboard layouts for typing comfortably: The Engram Study

Arno Klein[1], PhD

[1]Child Mind Institute, New York, NY, United States

Corresponding Author:
Arno Klein, PhD
MATTER Lab
Child Mind Institute
215 E 50th St
NY, NY 10022
United States
Phone: 1-347-577-2091
Email: arno.klein@childmind.org
ORCID ID: 0000-0002-0707-2889

Word count (Introduction through Discussion): 10,198 words

Abstract
Computer keyboard layouts have traditionally been based on legacy technologies, presumed but untested ergonomics principles, or the desire to increase typing speed, not on typing comfort. We crowdsourced typing preference data, determined that typing speed is a poor proxy for comfort, and derived ergonomics objectives and scoring criteria informed by what people prefer to type. We then applied these to a new approach to optimizing keyboard layouts called Engram, based on language-dependent n-gram frequencies and language-independent typing preferences. As a demonstration, we used multi-objective optimization to create keyboard layouts for typing in English (Engram-en) and Spanish (Engram-es). Engram-en performed well overall in a comparison with 31 other layouts. The approach and results raise questions about the field's focus on metrics that may not prioritize typing comfort. All data, software, documentation, and layouts are openly and freely available, and we encourage its use in optimizing layouts for different languages.

Keywords
optimized keyboard layout, touch typing, n-gram frequencies, bigram frequencies, typing comfort

1. Introduction

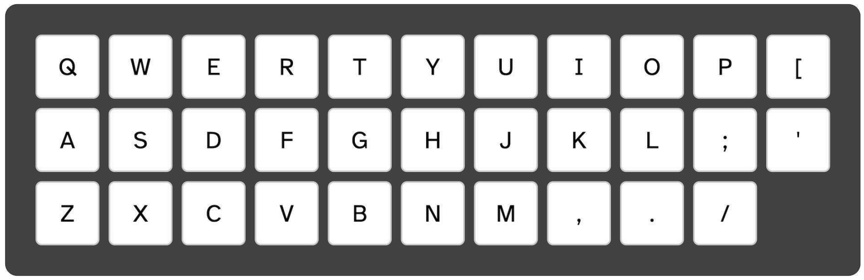
Since the invention of the computer, researchers, engineers, artists, and designers have invented and refined many ways to facilitate human interaction with the computer, such as manipulation of peripheral devices, motion and gesture tracking, voice commands, speech-to-text, and even brain-computer interfaces. Through it all the humble keyboard remains the principal human-computer interface, dominated in form by the conventional, flat, diagonal array of keys that come with computers, laptops, and accessed virtually on mobile device screens. These conventional keyboards are not optimized to accommodate the ergodynamics of the human hand or upper body (Ferguson and Duncan, 1974; Serina, et al., 1999; Amell and Kumar, 2000; Fagarasanu and Kumar, 2003; Garza et al., 2012). Less conventional physical keyboard designs reposition the keys or are shaped in ways intended to reduce the strain of repetitive typing. For example, some keyboards are split into left- and right-hand sides and angled to reduce bending of the wrists (Zipp et al., 1983; Çakir, 1995; Zecevic et al., 2000; Rempel et al., 2007; Rempel et al., 2009), are rounded to conform to the shape of the hand (Gerard et al., 1994; McLoone et al., 2009), position high-access keys in the middle for easy reach by the thumbs (Gerard et al., 1994), and arrange keys into perpendicular rows and columns to reduce diagonal finger movements (see Figure 1). Some also permit a choice of key switches, whose force–displacement characteristics can impact strain and fatigue (Rose, 1991; Rempel et al., 1997; Radwin et al., 1999; Bufton et al., 2006; Lee et al., 2009). The Kinesis Advantage (Lopez 1993; Gerard et al., 1994) and the Ergodox are two examples of commercial keyboard designs that also permit remapping of characters to individual keys, and therefore enable completely customizable *keyboard layouts*.

Since the vast majority of people simply use the keyboard bundled with their computer or physically integrated into their laptop, adopting a better keyboard layout has the greatest potential to significantly improve comfort and reduce strain for the greatest number of people who do touch typing. And to counter the resigned statement “It’s too difficult for people to switch keyboard layouts,” it is important to recognize: (1) there are hundreds of millions of people for whom it would not be a switch, including every new generation, (2) many languages do not yet have a well-established keyboard layout, and (3) people who suffer or do not wish to suffer repetitive strain injuries from typing but need to type have vested interest in improving the ergonomics in their lives. Free and open-source software such as Keyman (keyman.com) make it easy to switch the arrangement of characters on a given computer keyboard to over 2,000 different languages.

Developing an optimal keyboard layout for a given language is another challenge altogether. There are over four hundred septillion, or four hundred trillion trillion (26! = 403,291,461,126,605,635,584,000,000, or 4.03 E+26) possible ways to arrange a sequence of 26 letters, an NP-complete problem that is currently computationally intractable to optimize. Attempts to solve the “keyboard arrangement problem” have been ongoing (Ladany, 1975), with more recent contenders applying simulated annealing (Light and Anderson, 1993; Krzywinski, 2015; Salvo et al. 2016), ant colony optimization (Eggers et al., 2003; Wagner et al., 2003), Cyber Swarm (Yin and Su, 2011), and genetic algorithms (Deshwal and Deb, 2003; Walker, 2003; Goettl et al., 2005, Deshwal and Deb, 2006; Malas et al., 2008; Khorshid et al. 2010; Onsorodi and Korhan, 2020; Pacheco et al. 2020).

In this work, we outline a systematic approach to designing optimal keyboard layouts based on typing preferences and n-gram frequencies (where an n-gram is a string of n adjacent letters), and provide open source software for generating optimal keyboard layouts based on this systematic approach. To demonstrate the versatility of our approach, we created the “Engram-en” layout optimized for the English language (Figures 1 and 2), as well as the “Engram-es” layout optimized for Spanish (Appendix 2). Both of these layouts are freely available for anyone to use on multiple platforms. For evaluation, we compare the Engram-en layout with the layouts in Table 1 using a variety of scoring methods, including one we developed based on Dvorak’s original ergonomics principles.

Table 1. 31 keyboard layouts for the English language.

Layout	Year	Website
QWERTY	1873	 https://en.wikipedia.org/wiki/QWERTY
Dvorak	1936	https://en.wikipedia.org/wiki/Dvorak_keyboard_layout
Capewell-Dvorak	2005	http://www.michaelcapewell.com/projects/keyboard/index.htm
Colemak	2006	https://colemak.com/
Asset	2006	http://millikeys.sourceforge.net/asset/
QGMLWB	2009	http://mk.bcgsc.ca/carpalx/?full_optimization
Workman	2010	https://workmanlayout.org/
MTGAP	2010	https://mathematicalmulticore.wordpress.com/the-keyboard-layout-project/
Norman	2013	https://normanlayout.info/
Colemak-DH	2014	https://colemakmods.github.io/mod-dh/
Hieamtsrn	2014	https://mathematicalmulticore.wordpress.com/the-keyboard-layout-project/#comment-4976
Halmak	2016	https://github.com/MadRabbit/halmak
Boo	2021	https://ballerboo.github.io/boolayout/
Colemak Qi;x	2021	https://github.com/DreymaR/BigBagKbdTrixPKL/tree/master/Layouts/Colemak/Cmk-Qmods#colemak-qix-by-nyfee-2021-03
Engram	2021	https://engram-layouts.xyz/engram-2021
ISRT	2021	https://web.archive.org/web/20210210152903/https://notgate.github.io/layout/
Semimak	2021	https://semilin.github.io/
APTv3	2021	https://github.com/Apsu/APT#aptv3-layout
Sturdy	2021	https://o-x-e-y.github.io/layouts/sturdy/index.html
Whorf	2021	https://layouts.wiki/layouts/2021/whorf/#_top
Nerps	2022	https://www.reddit.com/r/KeyboardLayouts/comments/tpwyjc/sertain_nerts_nerps_low_redirect_low_sfb_low/
CTGAP	2022	https://github.com/CTGAP/ctgap-keyboard-layout
Canary	2022	https://github.com/Apsu/Canary
Octa8	2022	https://github.com/OctahedronV2/Octa8/blob/main/README.md#octa8
Seht Drai	2022	https://github.com/samuelxyz/layouts#seht-drai
BEAKL43	2023	http://ieants.cc/beakl/
Gallium	2023	https://github.com/GalileoBlues/Gallium
Graphite	2023	https://github.com/rdavison/graphite-layout
Recurva	2023	https://github.com/GalileoBlues/Recurva
Hanster-23	2024	https://commons.wikimedia.org/wiki/User:VTSGsRock/the_Hanster_Keyboard_Layout
Focal	2024	https://github.com/Keyhabit/Focal-keyboard-layout/

The rest of the Introduction is devoted to (1.1) briefly describing prominent keyboard layouts, (1.2) frequencies of letters, letter sequences, and punctuation marks, (1.3) typing speed and finger strength, (1.4) typing comfort, and (1.5) musculoskeletal disorders and hand kinematics.

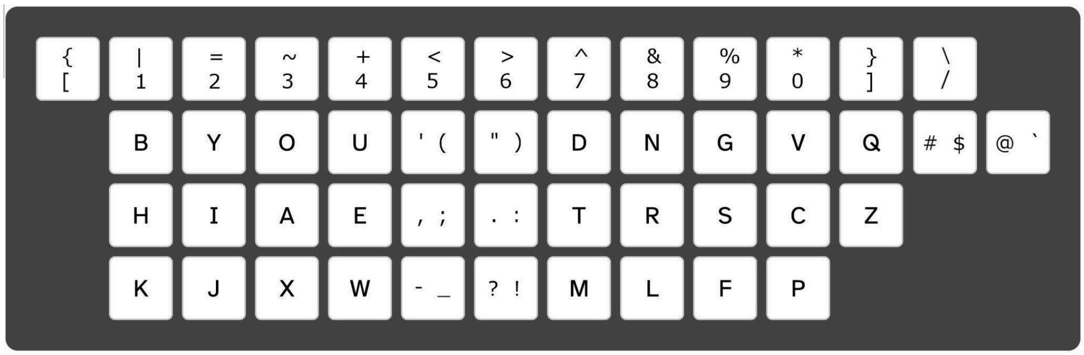


Figure 1. Complete Engram-en layout on an ortholinear keyboard.

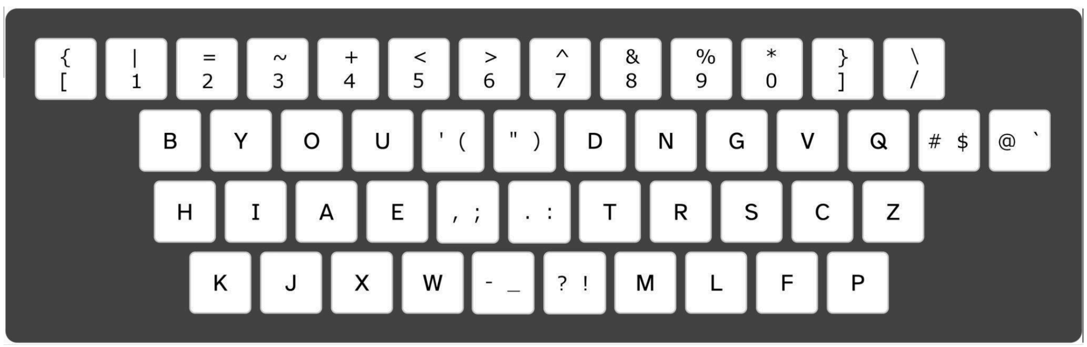


Figure 2. Engram-en layout on a conventional staggered keyboard.

1.1. QWERTY and other prominent keyboard layouts

Since the invention of the typewriter, the Sholes ("QWERTY") keyboard layout has been the preeminent layout, despite being generally acknowledged as inferior to most of its competitors (Amell and Kumar, 2000). Just consider that the third-least frequent letter in English, *J*, is assigned to what many consider the strongest (right index, or first) finger. For a thorough account of the history of keyboard layouts, see (Martin, 1981; Noyes, 1983; Koichi and Motoko, 2011), and for a debunking of myths about QWERTY origins, see Koichi and Motoko (2011):

> The development of QWERTY was a winding road, first by Sholes and others, second by Harrington and Craig, then by Jenne and Clough, again by Sholes, and at last by Wyckoff, Seamans & Benedict. There was no consistent policy towards QWERTY. The keyboard arrangement was incidentally changed into QWERTY, first to receive telegraphs, then to thrash out a compromise between inventors and producers, and at last to evade old patents. We have also argued in this paper that operators of Type-Writers in the 1870's were telegraphers and shorthanders. There were no professional typists at that time, and the operators typewrote with contemporary typing methods that were different from modern ones.

Over the last century, layouts have been introduced to allow for more efficient touch typing in English, but most did not attempt to optimize the arrangement of letters. Instead, most retain elements of QWERTY to make it easier to learn for QWERTY users (most notably Colemak, as well as direct offshoots of QWERTY such as QWERTZ and AZERTY). Exceptions include layouts generated by the CarpalX's simulated annealing of typing effort models (Krzywinski, 2015) and Halmak's genetic algorithm (Nemshilov, 2016), both of which are included in our analysis. Back in the 1930s, Dvorak (Dvorak et al. 1936; Dvorak and Dealey, 1936) undertook one of the most comprehensive reevaluations of the keyboard layout. See Appendix 1: "Dvorak scoring system" for information about Dvorak's criteria for the design and evaluation of keyboard layouts.

Most keyboard layouts that purport to improve typing efficiency, including Dvorak's, demand undue strain on tendons, particularly lateral extension of the first and fourth fingers. The first fingers (closest to the thumbs) must reach to the center columns to access letters, and the right fourth finger (farthest from the thumb) must reach to the periphery of the keyboard to access various punctuation. Following Dvorak, many layouts over-emphasize alternation between hands and under-emphasize same-hand, different-finger transitions. However, same-row, adjacent-finger transitions are arguably easier and faster than alternating hands to type keys that are far removed from one another. Many layouts also ignore the ergonomics of the human hand: different finger lengths and strengths, roundedness of the hand, relative ease of fourth-to-first finger roll-ins vs. first-to-fourth finger roll-outs, etc. Of great concern is that most proposed layouts are not based on established open access data, do not provide reproducible evidence for their superiority, and are not published in a peer-reviewed journal for scrutiny by the scientific community. When evaluations are conducted, they are often restricted to comparisons against the QWERTY layout and not to layouts that are known to be more efficient or designed based on ergonomics principles. Table 1 lists prominent layouts created over the course of the history of typing, including our contribution.

1.2. Frequencies of letters, letter sequences, and punctuation marks

Layouts after QWERTY have attempted to arrange frequently typed letters in easily accessed keys. Perhaps the most extensive analysis to date of n-gram frequencies was conducted by Peter Norvig (2012) on the largest general English text corpus. Norvig generated tables of counts, first for words, then for letters and letter sequences, from words with at least 100,000 instances in the English Version 20120701 of the Google books Ngrams dataset. From the resulting 3.564 trillion letters (26 unique) and 2.819 trillion letter-pairs (669 unique), the order of letters by frequency is: *E, T, A, O, I, N, S, R, H, L, D, C, U, M, F, P, G, W, Y, B, V, K, X, J, Q, Z,* and the 10 most commonly used letter-pairs in English are: *TH, HE, IN, ER, AN, RE, ON, AT, EN,* and *ND.* Only the first 8 letters (E to R) are needed to exceed 90% of the total key-pair frequency, 10 to exceed 95%, and 13 – just half the English alphabet (*E* to *U*) – to round to 99% (see Figure 3 and Table 2, with more detail in Supplement 1: Typing Study Input "n-gram frequencies" and "cumulative bigram frequencies" tables).

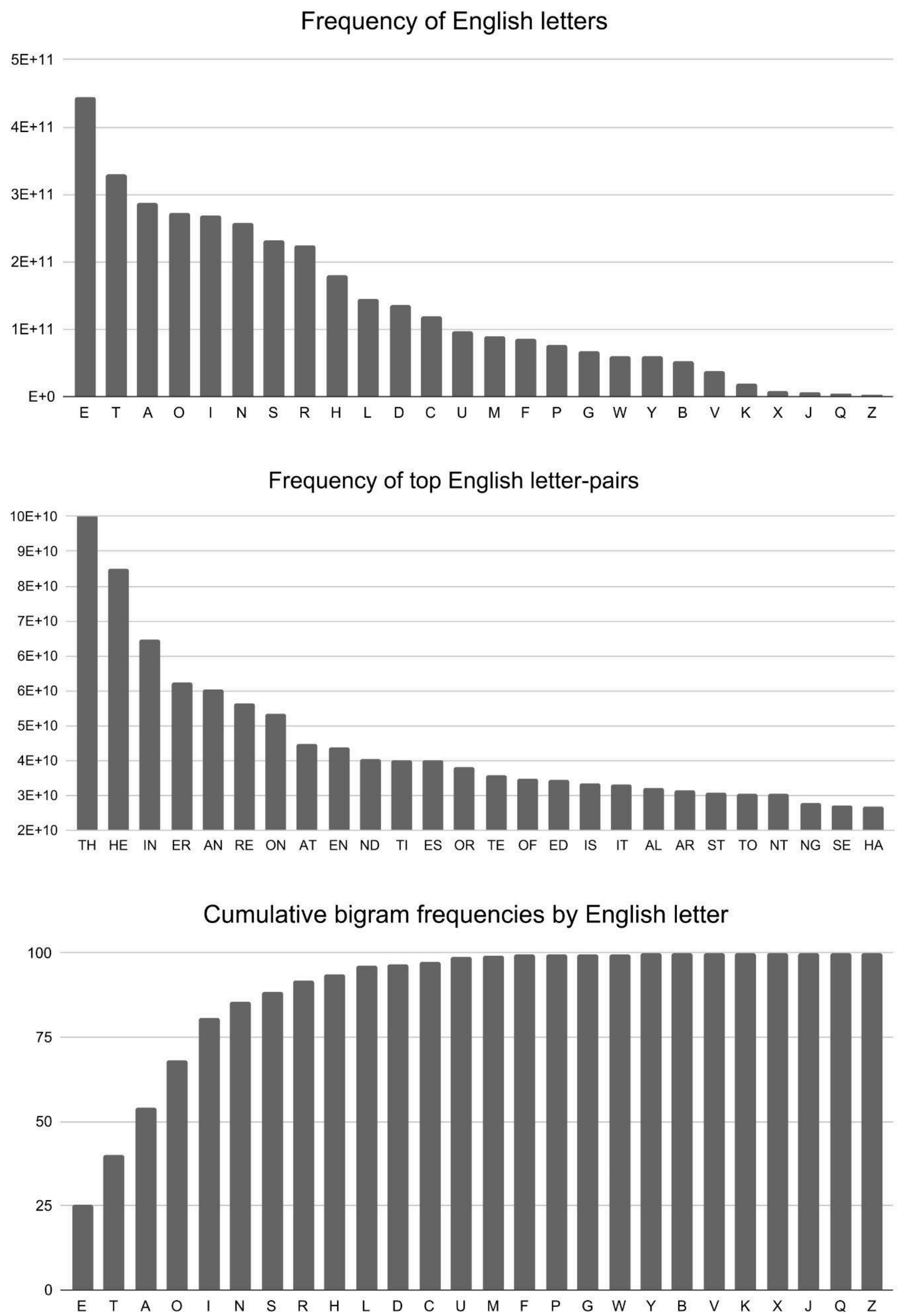


Figure 3. Frequency of English (top) letters and (middle) letter-pairs are combined to show (bottom) how much each letter cumulatively contributes to 100% of the bigram frequency in the Google books Ngrams dataset.

Table 2. Letter frequency and contribution to cumulative bigram frequency (Google books Ngrams dataset).

Letter	Frequency (%)	Cumulative bigrams	Cumulative (%)
E	12.48	51	25.15
T	9.29	100	40.22
A	8.05	147	54.00
O	7.63	192	68.33
I	7.57	235	80.76
N	7.24	276	85.39
S	6.51	315	88.64
R	6.28	352	92.02
H	5.05	387	93.51
L	4.07	420	96.28
D	3.82	451	96.73
C	3.34	480	97.38
U	2.73	507	98.72
M	2.51	532	99.27
F	2.40	555	99.44
P	2.14	576	99.68
G	1.87	594	99.74
W	1.68	609	99.75
Y	1.66	623	99.95
B	1.48	636	99.99
V	1.05	647	99.99
K	0.54	655	99.99
X	0.23	662	100.00
J	0.16	666	100.00
Q	0.12	668	100.00
Z	0.09	669	100

Table 3 aggregates the results of different studies that measured the frequencies of punctuation marks in publications and communications in English and in software programs. The table includes the 12 most frequent punctuation marks from each of the studies described below. We use this data to inform our placement of punctuation after assigning all letters to keys.

Table 3. Punctuation frequency according to different studies.

	Sun, 2018 (per 1M)	Malik, 2013 (% N-gram)	Malik, 2013 (%Twitter)	Cook, 2013 (per 1K)	Ruhlen, 1924 (%)	Xah, 2013 (%)	Xah, 2013 (% JavaScript)	Xah, 2013 (% Python)
,	44189.96		0.401	61.6	556	5.8	8.9	7.5
.	42840.02	1.151	1.694	65.3	535	6.6	9.4	10.3
-	9529.78	0.217	0.185	15.3	21	4.1	1.9	3.0
”		2.284	0.205	26.7	44	3.9	1.6	6.2
()	4500.81	0.140	0.089, 0.181		7	7.4	9.8	8.1
z		0.090	0.270					
?	4154.78	0.032	0.338	5.6	14	0.3		
:	3221.82	0.087	0.381	3.4	11	3.5	2.8	4.7
‘	2980.35	0.200	0.550	24.3	40	4.4	4.0	8.6
!	2057.22	0.013	0.940	3.3	3	0.4		
;	1355.22	0.096	0.048	3.2	22	3.8	8.6	
/		0.019	0.042			4.0	4.9	1.1
_		0.001	0.527			11.0	2.9	10.5
@		0.000	1.221			0.8		
#		0.000	0.337			2.2		

Sun, et al. (2018) published statistical values of punctuation frequency in 20 English-speaking countries from large-scale text corpora. The data were acquired through GloWbE, “a large English corpus collecting international English from the internet, containing about 1.9 billion words of text from twenty different countries. For further information on the corpora used, see https://corpus.byu.edu/.” Table entries are average frequencies per one million characters.

Malik and Findlater (2013) analyzed the frequency of punctuation input on touchscreen keyboards using Twitter as compared against Google N-grams. They found that punctuation in mobile tweets comprised 7.5% of characters versus only 4.4% in the Google corpus. Only six punctuation symbols appeared more frequently than the letter *Q* in the Google corpus (the comma was not included in the analysis). Table entries are in percentage of total characters from the Google N-gram corpus (version 1 containing 472,764,897 characters in English books published between the years 1538 and 2008), and from 173,876 mobile tweets (uniformly sampling 1% of the public tweet stream from June 2012).

Cook (2013) published on the frequencies of English punctuation based on a corpus of about 459,000 words, including three novels (276,000 words), selections of articles from two newspapers (55,000 words), one bureaucratic report (94,000 words), and assorted academic papers on language (34,000 words). Table entries are average frequencies per 1,000 characters.

Ruhlen and Pressey (1924) published a statistical study of what was then current usage in punctuation. Table entries are in frequencies per 10,000 words, drawn from 38,638 words from 100 business letters, 50 professional letters, and excerpts from one issue each of several newspapers and magazines. While it is limited in scope, it provides confirmation of later and larger studies.

Lee (2013) published online an analysis of the frequencies of punctuation in different software programming languages. Table entries are percent frequencies across all programming languages they included in their study, as well as for the C and Python programming languages. While some of the punctuation is biased toward C (19.8%) and Python (18.5%), which make considerable use of the underscore, there is some consistency across languages for the most frequent punctuation.

1.3. Typing speed and finger strength

Almost all prior work on optimizing keyboard layouts has focused on typing speed/efficiency, and has even assumed typing speed is a reasonable proxy for typing comfort. Perhaps the most relevant study regarding typing speed for keyboard layout optimization was conducted by İşeri and Ekşioğlu (2015). While the study generated data on key-pair typing speed, care must be taken when using these measurements, as they were collected from right-handed typists on a conventional QWERTY keyboard, without regard to the order in which the keys were typed.

Perhaps the largest open access dataset related to typing speed is the 136M Keystrokes dataset (Dhakal, V., et al., 2018). It contains keystroke data of over 168,000 users typing 15 sentences each. The data was collected via an online typing test published at a free typing speed assessment webpage. We use these data to evaluate the correlation of typing speed with Dvorak's principles as implemented by our Dvorak scoring algorithm (see Appendix 1).

Keyboard layouts universally increase load on the stronger fingers, but rarely are based on direct measures of individual finger strength. The only example known to the author is an article describing the design of a keyboard layout for the Filipino language (Salvo et al. 2016), which presents the "average finger strength of Filipinos [n=30, ages 16-36] measured in pounds": fingers 1–4 on the right hand were measured to produce 6.09, 6.37, 5.08, and 4.27 pounds of force, respectively, and on the left hand 6.57, 5.65, 4.54, and 3.77 pounds. However, these measurements probably don't represent relative strength relevant for typing: "Respondents were asked to sit in upright position, with their wrists resting on a flat surface. A pinch gauge was placed within each finger's reach. The respondents were asked to exert maximum pressure on the device." Another study of finger strength measured peak keyboard reaction forces by using finger flexor electromyograms, with 2.26, 2.36, 2.02, and 1.84 Newtons for fingers 1–4, respectively (Martin et al.

1996). These values corroborate other studies' measurements of average peak key strike, such as 2.2 + 0.7 N in 25 typists (Sommerich 1996), and 2.0 + 0.6 N in 10 typists (Armstrong 1994).

In the present study, we incorporate finger typing preference (rather than finger strength) as well as typing speed data after regressing out language-specific n-gram frequency effects, and we test the assumption others have made that typing speed is a reasonable proxy for typing comfort.

1.4. Typing comfort

While it is straightforward to directly quantify n-gram frequencies from appropriately curated text corpora (Douglas 2021a), and to objectively, quantitatively, and precisely measure typing behaviors such as speed, strength, and accuracy, there is no straightforward way to directly measure how comfortable it is to type individual keys or transitions between keys. Physiological sensors are used to infer stress and fatigue, but the author knows of no systematic attempt to directly ask participants to report what is easier or more difficult to type. This is understandable, given that participants would have a difficult time trying to give an absolute comfort score to typing individual keys or to typing each of hundreds of key-pairs. In the present study, we overcome this difficulty by having participants repeatedly type two different key-pairs and select which key-pair they prefer to type. We then use these typing preferences to statistically inform an ergonomics-based keyboard layout scoring system.

1.5. Musculoskeletal disorders and hand kinematics

While no study known to the author has directly evaluated injuries related to specific keyboard layouts, it may be assumed that the greater the discomfort of highly repetitive typing movements the greater the probability of typing-related injuries. It is certainly the personal experience of this author, and general knowledge amongst musculoskeletal disorder (MSD) specialists, that highly repetitive movements performed without regard to best practices in ergonomics can lead to injuries.

Upper extremity MSDs such as tendinitis, tenosynovitis, nerve entrapments, and carpal tunnel syndrome, are a significant work-related health problem (CDC 2021, Hopkins 1990). MSDs from overuse of the hands account for more than 50% of occupational injuries (Rempel et al 1992), with more than 50% of individuals in one study reporting symptoms during the first year of their job (Gerr et al 2002). The National Institute for Occupational Safety and Health created the most comprehensive compilation of epidemiologic research on the relation between selected MSDs and work-related factors (NIOSH 1997). The report summarizes results from the U.S. Department of Labor's Annual Survey of Occupational Injuries and Illnesses, the only routinely collected information about occupational injuries and illnesses of U.S. workers. According to this survey, 92,576 injuries or illnesses were due to repetitive motion, including typing or key entry, repetitive use of tools, etc. According to an updated version of the report from 2019 (Dept. of Labor 2019), there were around 35,000 reported MSDs in the hands and wrists alone.

Several studies intended to improve our understanding of MSD risk factors in typing have measured metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joint postures, velocities, and accelerations, as well as tendon travel, hand movement, and key strike force (Long 1994, Sommerich 1994, Sommerich 1996, Baker 2007). While fingertip angles at the point of contact with keys differ between males and females (due to shorter fingers on average in females) and differ between the first finger and the longer fingers (Long 1994), it is not clear what if any anthropometric measures of the hand might predict typing-related MSDs, since hand length doesn't predict MCP or PIP postures, velocities, or accelerations (Sommerich 1994, Sommerich 1996). However, MCP and PIP joint postures have been shown to affect carpal tunnel pressure; and frictional work, the summation over time of the product of tendon force and incremental tendon travel, is directly influenced by repetition, and is thought to be a risk factor for MSDs (Sommerich 1996). In a study that made use of a slightly modified key layout that assigns more common keys to the home row than in the standard QWERTY layout, tendon travel, velocity, and acceleration decreased for all study participants, indicating a relationship between keyboard layout and reduction of MSD risk factors (Sommerich 1994, pages 138, 159, and 212).

2. Methods

In this section, we describe our methods for: (2.1) crowdsourcing key-pair typing preferences, (2.2) converting preferences to ergonomics objectives, (2.3) scoring keyboard layouts, (2.4-5) optimizing keyboard layouts, and (2.6) evaluating and comparing keyboard layouts.

2.1. Crowdsourcing key-pair typing preferences

To determine which key-pairs people prefer to type on a QWERTY computer keyboard, we built an online web app to collect timing and preference data (see Data and Software Availability; Figure 4 and Supplement 2 contain screen captures of the jspsych web app interface). After a participant agrees to an electronic consent and reads the study instructions, the web app presents a screen with random text interspersed with multiple instances of two different space-delimited key-pairs. The participant types the text, then selects which key-pair is easier to type by moving a slider bar. This is repeated 60-100 times depending on the stimulus set, with new text on each screen. The web app stores the timing and preference data in a repository on Open Science Framework (osf.io). Python scripts process the collected data and measure correlations between typing preference, speed, accuracy, etc.

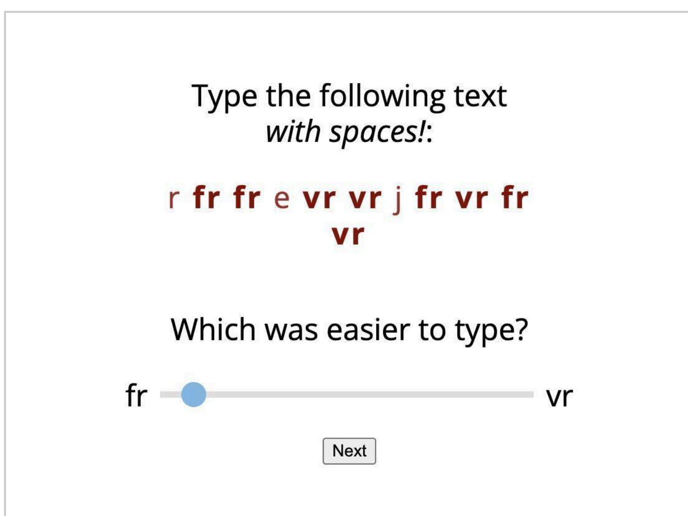


Figure 4. Screenshot of the Bigram Typing Preference Study web app.

Whereas the random text strings presented in the web app require the participant to type with both hands, the space-delimited key-pairs are constrained to letters typed by the left hand, for three reasons. First, assuming bilateral symmetry of kinematics and ergonomics across the left and right hands drastically reduces the amount of data to collect and analyze. Second, typing excessively with one hand increases the probability of fatigue and strain, which would help participants make their preference decisions. Third, typing with the non-dominant hand likely requires greater attention paid to the task, and only one of the non-Prolific participants reported being left-handed (see below). The latter two reasons were borne out by participant comments.

We conducted ten data collection sessions, with different participants, key-pairs, and text lengths. Supplement 1: Typing study input “participants (Prolific)” and “participants (non-Prolific)” tables contain information about the participants, and the “bigram pairs” table contains the key-pair choices presented to the participants. Over 450 unknown participants were recruited through the Prolific online platform (www.prolific.com) to take part in one of seven sessions, and 44 known (non-Prolific) participants were recruited through email and mailing lists to take part in one or more of four sessions, for a total of 62 contributions. Prolific participant metadata includes: fluent languages, sex (female: 278, male: 223), age (mean: 41.3, SD: 15.0 years), ethnicity, and student/employment status. Non-Prolific participant metadata includes: handedness (only one reported being left-handed), primary language (25% reported a language other than English), sex (22 reported female, 19 reported male, 3 unreported), age (mean: 41.6, SD: 16.7, min: 18, max: 85 years), height in cm (mean: 172.3, SD: 9.6, min: 152.4, max: 190.5), and comments regarding how easy it was to type the key-pairs and to make a choice between the key-pairs. All participants used their own keyboard and all filled out the same electronic consent form. Because all data were collected anonymously, no IRB oversight was required. Exemption from IRB oversight was approved by the Advarra Institutional Review Board. Participants using the Prolific website are required to agree to their Terms of Service notification (https://prolific.ac/assets/docs/Participant_Terms.pdf) before being allowed to complete surveys. Per the IRB exemption, no additional informed consent was required.

The key-pair typing preference studies include a diverse set of key-pairs. Key-pairs were selected for each of the studies as follows (the number in parentheses indicates the number of unique key-pairs that were presented twice to each participant, once with the first key-pair on the left side of the slider bar, and another time on the right side; comfort scores in Study 8 were derived from a Bayesian preference learning model trained on feature data from Studies 1-5):

- Studies 1-5 (Prolific: 407): Manually selected pairs
- Study 6 (non-Prolific: 50): Pairs maximally distributed across PCA feature space
- Study 7 (non/Prolific: 50): Pairs maximally distributed to complement pairs in Study 6
- Study 8 (non-Prolific: 37): Pairs with similar comfort scores
    - Rank comfort scores for key-pairs previously chosen by Prolific participants
    - Select top 29 pairs of ranked neighbors
- Study 9 (non-Prolific: 34): Same-key pairs

2.2. Converting typing preferences to ergonomics objectives

The goal of analyzing the above study data was to determine what finger placement and movements touch typists prefer, to establish ergonomics objectives for touch typing that can guide optimization and evaluation of keyboard layouts. Our general approach emphasizes effect sizes and confidence intervals to quantify practical differences, with statistical tests to confirm that preferences are detectable above chance.

Study 9's data allows us to determine which keys people prefer to type, and all other data allow us to test which key-pair transitions people prefer to type. We tested the following transitions: row separation (how many rows apart the two keys are), column separation (how many columns apart they are), lateral stretch (whether keys are to the right or left of the eight "finger-columns" that extend up and down from the home keys), and finger sequence (order that fingers type, either inward toward the thumb or outward away from the thumb). Given traditional physical keyboard layouts arrayed as three rows of letters, the only transition category we can fully control is lateral extension, at least for 24 of 26 letters, by organizing 24 letters in the eight finger-columns. Among the tests, we evaluated whether typists prefer to type within, or laterally extend outside of, these finger-columns.

We collected data for the left hand typing on the left side of a keyboard, and assumed bilateral symmetry of hand kinematics and typing preferences. The questions listed below outline our tests for establishing typing ergonomics objectives for evaluation and optimization of keyboard layouts:

1. Key preference: Which same-key key-pairs do people prefer to type?
2. Row separation: Do typists prefer to type on the same row rather than reach to an adjacent row? Do they prefer to reach than to hurdle (type keys that straddle the home row)?
3. Column separation: Do typists prefer to type in the same column vs. in different columns? Do they prefer to type in adjacent columns or distant columns, when typing in the same row, reaching, or hurdling?
4. Lateral stretch: Do typists prefer to type within the eight finger-columns vs. laterally stretching the left index finger rightwards (staying within the same row)?
5. Finger sequence: Do typists prefer to type fingers in a sequence toward the thumb, when typing in the same row, reaching, or hurdling?

2.3. Scoring keyboard layouts

To optimize and evaluate keyboard layouts, we define a layout scoring system based on six ergonomics objectives (see 3.3, Table 4, and Supplement 4: "Empirical Basis for Engram Scoring"). Each bigram (letter-pair assigned to a key-pair) receives a frequency-weighted score for each objective, where the weight is the frequency of that letter-pair in a given language. These are averaged across all (N!) possible bigrams to give an overall layout score.

The following equation defines a layout score as a function of letters-to-keys assignment *p*, where *L* is the set of all *N* letters, *i,j* is a letter-pair, $w_{i,j}$ is the frequency of the letter-pair, *p(i)*,*p(j)* is the letter-pair mapped to a key-pair (bigram), and $S_c(w_{i,j}p(i), p(j))$ is the bigram's weighted score for objective *obj*:

$$S_{obj}(p) = 1/N! \sum_{i,j \in L} S(w_{i,j}, p(i), p(j))$$

Multi-objective optimization (MOO) optimizes across *M* multiple objectives without having to combine them. We will use a MOO approach, because it offers several advantages over traditional weighted-sum methods. There are no arbitrary weighting decisions, which are often difficult to determine and can bias results. Instead, each objective contributes equally to the optimization process, preventing any single objective from dominating the search. Also, MOO can reveal trade-offs between different objectives, for example showing when improving finger position scores comes at the cost of finger transition optimization. Rather than producing a single "optimal" layout, MOO generates a Pareto front of non-dominated solutions, providing multiple layout alternatives with different objective trade-offs. A layout is considered dominated if at least one of the current solutions has objective scores that are all at least equal to, and at least one greater than, the contender's scores. A multi-objective framework ensures that our optimization explores trade-offs between different scoring objectives.

2.4. Optimizing keyboard layouts

To optimize a keyboard layout in a given language, we implemented a general-purpose layout optimization software in Python (see Data and Software Availability). The software optimizes an arrangement of items in positions based on item-pair (*i,j* and *j,i*) and position-pair scores (and optionally on item-triple and position-triple scores). In our use-case, n-grams (letter-pairs and letter-triples) correspond to ordered sequences of items, and key permutations (key-pairs and key-triples) correspond to ordered sequences of positions. N-gram scores are (normalized) frequencies of n-grams in a given language, and key permutation scores are computed on multiple objectives that are independent of language, as usage frequencies are regressed out.

2.4.1. Confine letters to keys that minimize lateral finger movements

Before addressing the arrangement of characters on a keyboard, it is important to define the boundary containing letters so that finger movements are constrained in beneficial ways. Given that a standard (non-split) computer keyboard is narrower than the average adult shoulder width, wrists have to bend to place fingers in alignment with the keyboard. And given that a standard keyboard has diagonally staggered keys, fingers must make lateral movements when typing keys in a different row. However, a finger moves much less laterally when typing within a finger-column than within a row. We mentioned above that lateral stretch is the only key-pair transition category that we can fully control. To do so, we assign the most frequent 24 letters to the 24 keys in, above, and below each home row (home block), and reserve the two center columns for punctuation. These eight finger-columns require no lateral finger movements when touch typing on a split, ortholinear keyboard, and minimal lateral movements on a standard, diagonal keyboard, since each column is accessed by one finger. The Results section will show statistical evidence to justify this strategy.

2.4.2. Iteratively assign the most frequent letters to the most comfortable keys

There are over 600 sextillion (24! = 6.20 E+23) possible arrangements of 24 letters, which is currently computationally intractable. In experiments on various high-end personal computers and on the Pittsburgh Supercomputer Center's Extreme-Memory nodes (Brown et al., 2021), arranging 10–11 letters in the former and 11–12 letters in the latter appears to be the practical limit for memory and time constraints using our code. However, there are factors that predispose this seemingly intractable problem to be fragmentable. As detailed in 1.2 above, it only takes 8 to 10 of the most frequent letters to cover most of the total bigram frequency; the least frequent letters have minimal interaction with the most frequent letters. And as will be shown in the Results section, keys may be grouped by (Bradley-Terry tiers representing) statistically significantly different degrees of preference, with the greatest differences between the top four keys and those below in preference, and the bottom six keys and the 18 above in preference. Taken together, these justify breaking the problem up into iterative steps that focus on these different preference groups. Of course, as computational resources permit, and better-informed data-driven objective, cost, and search functions are devised, constraints can be relaxed and search spaces expanded.

The entire approach can be summarized in the following steps (with details below, and examples and figures in 3.4):

1. **10 letters**: Assign 10 letters to 10 keys (constraining 2 letters to 4 keys).
2. **18 letters**: Remove 2 letters in each layout from Step 1, and assign 10 unassigned letters to fill 18 available keys. Filter the resulting layouts.
3. **24 letters**: Remove 4 of the 18 letters in each layout from Step 2, and arrange the 10 remaining of the 24 letters. Filter the resulting layouts.
4. **26 letters**: Arrange the final 2 of 26 letters. Select the layout with the highest average score.
5. **Punctuation and symbols**: Systematically assign punctuation to the two middle columns between the home blocks, and symbols to the Shift-number keys.

In Steps 1–3, MOO optimally assigns the highest-frequency unassigned letters to the same number of highest-preference available keys, resulting in hundreds of Pareto front solutions. In Steps 2–3, prior to assigning letters to each of the Pareto solutions, some letters assigned to the lowest-preference keys are unassigned, to explore more of the solution space. In Step 1, since typists prefer typing within the home row (see Results), we optimally arrange the 10 highest-frequency letters within 10 keys: the eight highest-preference keys containing six home keys, plus the two remaining home keys. Steps 2, 3, and 4 fill 18, 24, then all 26 keys. The 26-letter layout with the highest average score is selected as the winning layout.

To reduce computation, we apply constraints to Steps 1 and 3. In Step 1, we constrain the two highest-frequency letters to the four highest-preference keys. We retain only those layouts with the highest-frequency letter on the left side to remove mirror redundancy in the following steps, and to balance the load incurred by the extra keys to the right of the finger-columns when the layout is complete. In Step 3, since the home-block arrangement forces otherwise laterally displaced letters to be vertically stacked in finger-columns, and given that from a MOO perspective all solutions are equally valid, we remove Step 2 layouts that have common bigrams (that contribute to 50% of the total cumulative bigram frequency) requiring one to type with either one finger or a hurdle.

Processing time is dictated by the processing resources available, of course, but also by the number of letters to assign, the number of preassigned letters, and the size of the n-grams under comparison (bigrams vs. trigrams). We implemented a branch-and-bound, depth-first search algorithm with a tight upper bound calculation, but the overhead to run these searches took longer than simply conducting an exhaustive search, at least up to the 10 letter assignments in the steps above, so we opted to conduct exhaustive searches. Step 1 takes a Macbook Pro laptop (Apple M3 Max with 36GB Memory running Sequoia 15.4.1 and the current code) two minutes to exhaustively search for optimal arrangements of 10 letters without any preassigned letters. Step 2 takes approximately 100 minutes to assign the same number of letters (to each of the layouts from Step 1), since there are an additional 10 preassigned letters to factor into the scoring and many more bigrams and trigrams to score. Step 3 takes over 220 minutes to assign the same number of letters again (to each of the layouts from Step 2), since there are now 14 preassigned letters. We run all software using Python 3.13 with the following libraries (versions are minimal requirements): scipy 1.15.3, scikit-learn 1.7.0, pandas 2.2.3, statsmodels 0.14.4, numba 0.61.0, factorial 1.0.0, openpyxl 3.1.5, pyyaml 6.0.2, tqdm 4.67.1, psutil 6.1.1, matplotlib 3.10.0, and seaborn 0.13.2.

2.5. Optimizing keyboard layouts for English and for Spanish
To optimize a layout for English, we use our layout scoring system (2.3) derived from the crowdsourced typing data (2.1), and the English n-gram frequencies from the Google books Ngrams dataset described in 1.2 above (Norvig, 2012). Creating an optimized layout for a different language requires letter and letter-pair frequency data extracted from a representative text corpus for that language, and the layout would need to accommodate any additional characters not commonly used in English. For a Spanish version, for example, we would need to include additional punctuation as well as diacritical marks. Douglas (2021b) computed Spanish letter and letter bigram frequency data after preparing the Leipzig Spanish corpus (https://wortschatz.uni-leipzig.de/en/download/Spanish) for this study. He selected the largest file from each row of the corpus, except for rows specified as not from Spain, after removing lines containing non-Spanish names and words. A previous version of the English and Spanish layouts developed by a preliminary version of the Engram approach have been tested and in use by typists for a few years.

2.6. Comparing keyboard layouts
We compared the Engram-en English keyboard layout with 31 other English language keyboard layouts listed in Table 1. This list includes most of the layouts listed in Getreuer's statistics table comparing alternative keyboard layouts (Getreuer 2025a), excluding layouts that require special keys or have letters outside of the 32 standard keys. When multiple versions of a layout were available, we selected the angle-mod or ANSI staggered-row version.

Recognizing that (i) there are many keyboard layout scoring systems (see Keyboard Layout Analyzers, 2025), (ii) there is considerable disagreement about which metrics are important (see discussions among the very active and informative "Alt Keyboard Layouts" Discord server community, 2025), and (iii) no single score in isolation could possibly reflect all of the qualities of a given layout, we availed ourselves of multiple scoring systems, either implemented in Python or accessible online. The online layout analysis tools provide considerable detail, with statistics about individual finger, row, and column use. We selected their highest-level metrics that we considered to be interpretable (such as percentages of single-finger bigrams, and excluding metrics such as "effort").

The Engram, Dvorak-4, and Comfort Scorers all use the English n-gram frequencies from the Google books Ngrams dataset. The online scorers require a text corpus. The Layouts Wiki uses the “Reddit Corpus (small)” (2025) from ConvoKit, which contains approximately 54 million characters: “This corpus is a collection of nearly 300,000 real conversations from Reddit, gathered from 100 popular communities in September 2018. It contains posts and comments from over 100,000 different users discussing everything from technology to hobbies to current events. What makes this corpus particularly valuable for keyboard layout analysis is that it represents how people actually communicate online—complete with informal language, abbreviations, slang, and the natural flow of digital conversation.” For KLAnext, we used a modified version of the largest corpus it offers, the Chained English Bigrams 9. We removed all punctuation from the corpus, because some online scorers do not permit full configurability, and because we are principally interested in the optimal arrangement of letters in this study, and less interested in optimizing other characters for specialized tasks such as programming, text formatting, etc.

Engram Scorer. We score each layout using the same six objective scores used to optimize layouts in this study (software implementation of the layout scoring system described in 2.3). Note that it is by no means a foregone conclusion that layouts we generate using our objectives will score higher than other layouts, for the following reasons. First, our layouts were optimized within the restricted confines of the home blocks, whereas all other layouts assign letters inside and outside of these blocks. Second, we use multiple independent objectives, so our layouts will by design permit tradeoffs between different objectives; layouts derived elsewhere can very well exceed our layouts in any of these objectives. Third, our approach is based on typing preferences, which are a function of biomechanical factors that could be optimized more directly by other layout systems. Indeed, the Results section will show that some of the layouts in Table 1 perform better on some of the Engram Scorer objectives.

Dvorak-4 Scorer. We have derived and validated perhaps the first quantitative scoring system from the original Dvorak, et al. 1936 “Typing Behavior” book and the “Dvorak-Dealey ‘Simplified’ Typewriter Keyboard” patent. See Appendix 1 (and 3.1.2) for background, description, and analysis of this scorer.

Comfort Scorer. We trained a Bayesian preference learning model on the original bigram typing preference data to estimate latent comfort scores for every possible bigram. Layout scores are the average across all possible bigrams’ comfort scores multiplied by their frequencies in the Google books Ngrams dataset. These scores are independent of the Engram scoring method.

Layouts Wiki (2025) implementation of the Keygen Pro analysis tool offers the most detailed set of measures, from which we selected the same-finger and lateral-stretch bigram and skipgram metrics. All metrics are described in detail in the Keyboard Layouts doc (Getreuer 2025b) and in the Layouts Wiki website.

KLAnext (Gillespie and Douglas, 2025) is based on the original Keyboard Layout Analyzer created by Patrick Gillespie in 2008, and has undergone various modifications, most recently by Ian Douglas. We used its distance measure, as none of the other scorers offer this measure.

3. Results

The results below are organized as follows: (3.1) data quality, (3.2) typing speed, preference, and letter-pair frequency, (3.3) objectives results and scoring criteria, (3.4) optimized keyboard layout, and (3.5) comparison of keyboard layouts.

3.1. Data quality

3.1.1. Prolific vs. non-Prolific data

Extensive tests revealed that the Prolific participant preference data is of lower quality than the non-Prolific data. A participant has two attempts to choose between the same two key-pairs. Prolific participants were less consistent when choosing key-pairs than were non-Prolific participants, more of them exhibited statistically improbable choice patterns, and more were exposed as inattentive by failing attention checks. Attention checks included picking clearly less comfortable key-pairs, or selecting the same side of the slider bar at least 20 times in a row (only one of the non-Prolific participants was removed due to the latter check). Bayesian models trained on Prolific data also led to worse predictions than models trained on non-Prolific data or on both non-Prolific and Prolific data. Non-Prolific models predicted preferences in held-out non-Prolific data better than Prolific models predicted preferences in held-out Prolific data. Non-Prolific models also generalized better: they performed at least as well as most Prolific and non-Prolific + Prolific models at predicting preferences in held-out Prolific data. For both datasets, we therefore removed all trials with inconsistent choices. After applying this filter, the non-Prolific dataset (43 unique participants volunteering for a total of 61 sessions and 5,236 trials) contains 4,490 trials representing 165 unique bigram pairs.

3.1.2. Dvorak's principles and typing speed

An empirical analysis using 19.4 million correctly-typed bigrams from the 136M Keystrokes Dataset (Dhakal et al., 2018) to determine which of Dvorak's theoretical principles actually correlate with typing speed in practice revealed that Dvorak's criteria show very weak correlations with typing speed. A 4-criterion model (hand/finger alternation, row separation behavior, finger-column adherence, and non-adjacent finger usage) achieved the strongest correlation with actual typing speed ($r = -0.1912$, $p < 0.001$), explaining 3.7% of speed variance. This 4-criterion model significantly outperformed the full 7-criterion model ($r = -0.1564$), so we used these four criteria as the Dvorak-4 Scorer in our analysis.

3.2. Relationships between typing speed, preference, and letter-pair frequency

3.2.1. Speed and frequency

We conducted an analysis of key-pair typing preferences, speed, and accuracy (see Supplement 3: Typing study output “typing data”, “typing speed”, and “typing accuracy” tables). To separate the effects of familiarity from inherent biomechanical ease or comfort, we examined whether more frequent letter-pairs are typed faster, which would suggest that practice and exposure influence typing performance. As expected, more frequent letter-pairs are indeed typed more quickly (Figure 5). The correlation between typing time and frequency was -0.401 (p=2.20e-5, R-squared: 0.161) for typing both keys and -0.315 (p=0.001, R-squared: 0.099) for transitioning from the first to the second key. These correlations underscore the need to control for frequency.

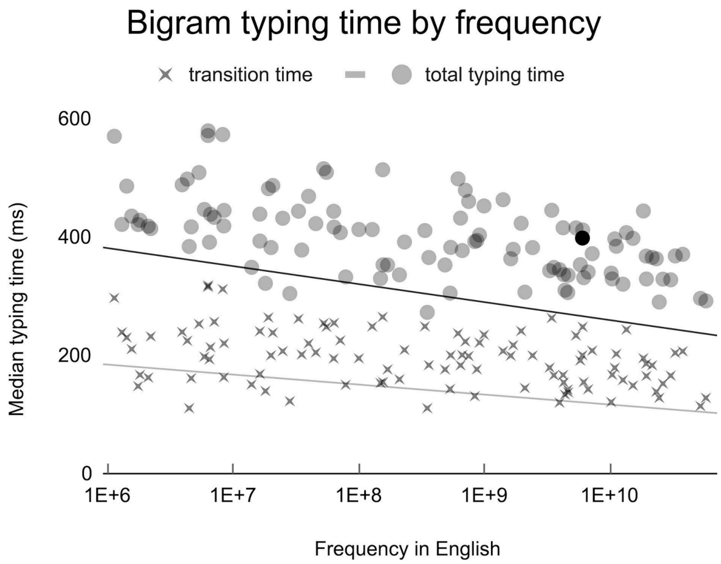


Figure 5. Bigram typing time (either transitioning between keys (x) or typing both keys (o)) decreases as a function of frequency (shown here on a logarithmic scale).

3.2.2. Speed and preference

Since participants are instructed to select the key-pair that is easier or more comfortable to type, we should expect that ease would be correlated with speed, so next we examined the relationship between typing speed and preference. When a key-pair is chosen, it is generally typed more quickly than when it is not chosen, with a correlation of 0.551 (p=1.76e-14) between their median typing times (see Figure 6). Participants chose the key-pair they typed faster 66.9% of the time, and the variance explained by speed difference is 5.7%. If we treat speed as a proxy for preference, then we get a mean prediction accuracy of 66.4% (SD: 12.9%; range: 39.5–89.0%). If it were the case that people type preferred key-pairs faster, or they prefer key-pairs they type faster, and they were consistent in their preferences, then typing speed could serve as an objective measure in place of, or as a proxy for, preference. However, these results are not compelling enough to warrant substituting speed for preference, so we continue with our approach below.

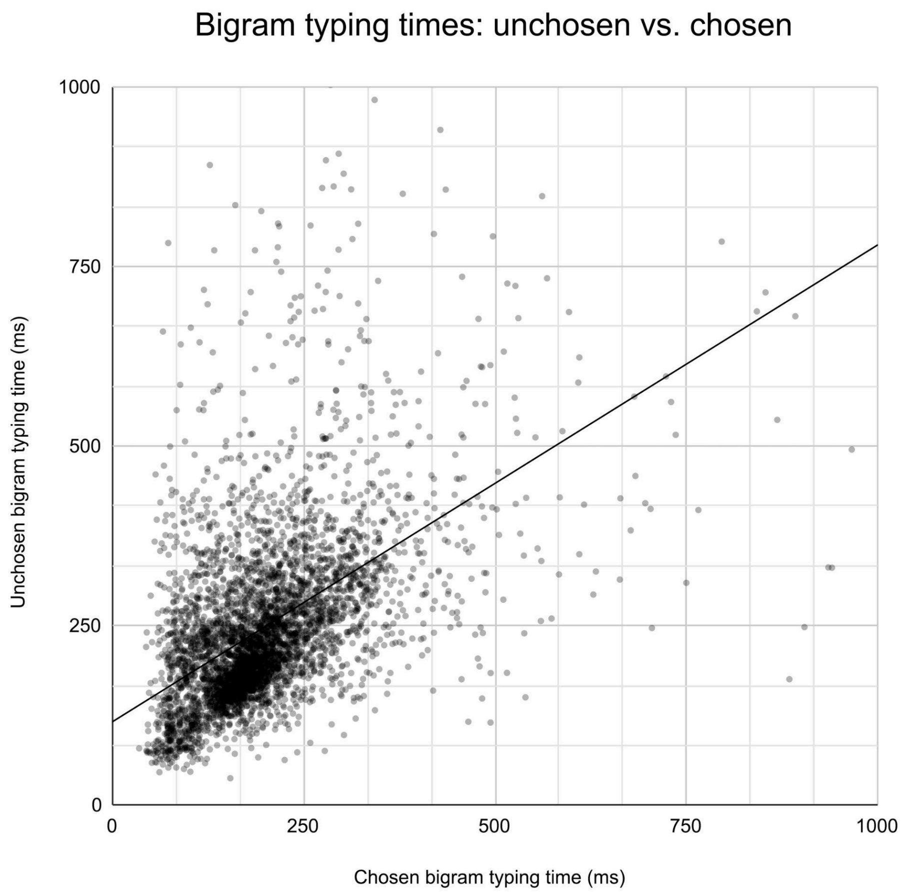


Figure 6. Bigram typing times are shorter on average when they are chosen (the trend line shows that more markers are above the x=y diagonal; axes are truncated to 1000 ms for clarity).

3.2.2. Preference magnitude and choice consistency

We expected that stronger preferences, indicated by higher absolute slider values, would indicate greater conviction in decision making. This should be reflected in consistent choices across presentations of two key-pairs. We did not find a consistent relationship, however, so we dichotomized the slider values and treated these as binary decisions for our analyses.

3.3. Test results and scoring criteria

The test results for each objective can be used to establish evidence-based ergonomics objectives and scoring criteria for optimizing and evaluating keyboard layouts. Briefly, the participants preferred typing: (1) keys in roughly seven tiers of preference; (2) within the same row vs. reaching and reaching vs. hurdling; (3) adjacent keys in the same row, and finger sequences toward the thumb in the same row; (4) with separate fingers vs. the same finger; and (5) within the finger-columns vs. lateral stretches. Below we report Prolific / non-Prolific results, with *n* representing the number of key-pairs evaluated per condition. All tests exclude same-column key-pairs except *key preference* and *column separation* tests, and all tests exclude keys in center columns except the *lateral stretch* test.

Key preferences. A Bradley-Terry analysis of the home-block same-key-pair preference data (non-Prolific, n=756, 36 unique) resulted in seven tiers of decreasing preference. Statistical tests of mixed-key-pairs confirmed these divisions. Figure 7 shows the Bradley-Terry strengths with confidence intervals and Table 4 includes a list with a normalized average of Bradley-Terry strengths per tier and keys within the tier (QWERTY key positions are provided for reference). Keys outside of the home-block are assigned a score of zero, which doesn't affect our optimization. See Supplement 4: “Empirical Basis for Engram Scoring” for details.

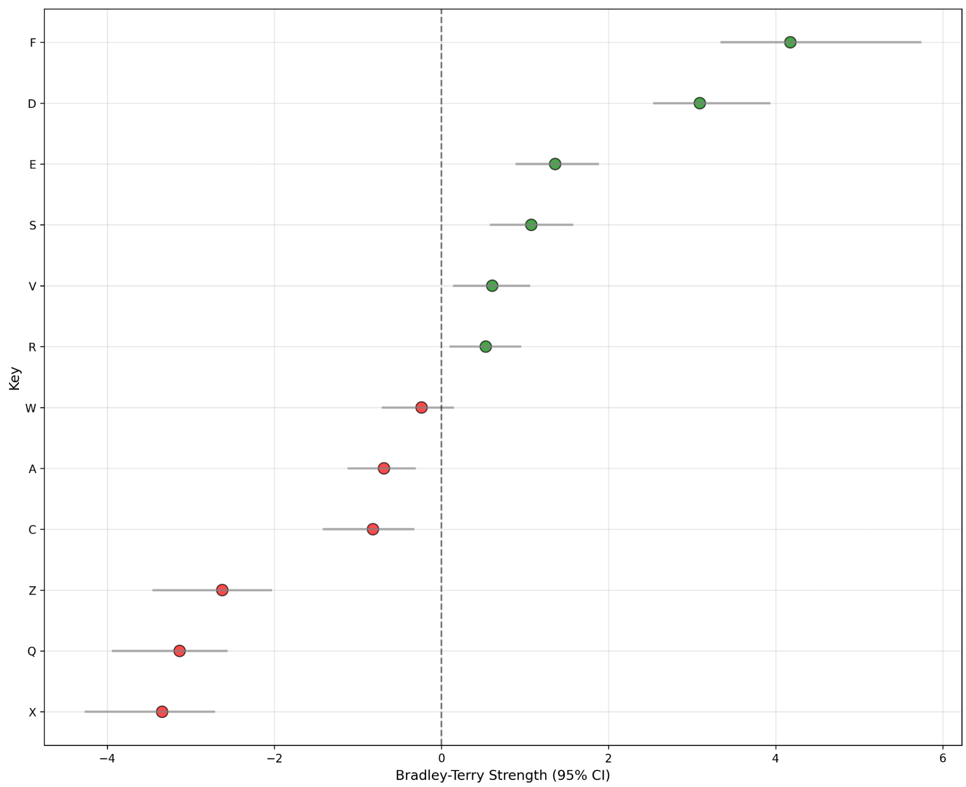


Figure 7. Bradley-Terry strengths and confidence intervals for left home-block keys (labeled as QWERTY letters).

Table 4. Key-pair and key-triple base scoring criteria

Key preference 1. 1.000: 24 (F), 27 (J) 2. 0.870: 23 (D), 28 (K) 3. 0.646: 13 (E), 18 (I), 22 (S), 29 (L) 4. 0.568: 34 (V), 37 (M), 14 (R), 17 (U) 5. 0.472: 12 (W), 19 (O) 6. 0.410: 21 (A), 30 (;), 33 (C), 38 (,) 7. 0.137: 31 (Z), 40 (/), 11 (Q), 20 (P), 32 (X), 39 (.)
Row separation 1.000: 2 hands 1.000: 2 keys in the same row 0.588: 2 keys in adjacent rows (reach) 0.000: 2 keys straddling the home row (hurdle)
Same row (finger sequence and column separation) 1.000: 2 hands 1.000: adjacent columns, same row, inward 0.779: adjacent columns, same row, outward 0.750: remote columns, same row, inward 0.584: remote columns, same row, outward (0.779×0.750) 0.500: different rows, different fingers 0.000: same finger
Same finger 1.000: 2 hands 1.000: 2 fingers 0.000: 1 finger
Lateral stretch 1.000: 0 fingers outside the 8 finger-columns 0.500: 1 finger outside the 8 finger-columns 0.000: 2 fingers outside the 8 finger-columns
Key-triple finger sequence 1.000: 2 hands, alternating twice to type 3 keys 1.000: single typing direction within each hand 0.000: switch typing direction on one hand

Row separation. An analysis of same-row, reach, and hurdle key-pairs confirmed that participants favor typing in the same row vs. reaching to an adjacent row (n=1,576 / 502; effect size: 21.1% / 15.7%; 95% CI: [68.8%, 73.3%] / [61.5%, 69.8%]; p=0.0000 / 0.0000), and favored reaching vs. hurdling (n=1,462 / 654; effect size: 26.1% / 36.5%; 95% CI: [73.8%, 78.2%] / [83.7%, 88.9%]; p=0.0000 / 0.0000). A sample-size-weighted meta-analysis of the data generated the normalized scores of 0, 0.588, and 1 for hurdles, reaches, and same-row bigrams (see Supplement 4).

Column separation (including same- vs. different-finger). There was a small but statistically significant effect favoring typing in different columns or fingers vs. same column in the larger Prolific dataset (n=1,474 / 354; effect size: 4.4% / 1.4%; 95% CI: [51.9%, 56.9%] / [46.2%, 56.6%]; p=0.0007 / 0.5951). Participants favored typing in adjacent vs. distant columns within the same row (n=56 / 120; effect size: 25.0% / 25.0%; 95% CI: [62.3%, 84.5%] / [66.6%, 81.9%]; p=0.0002 /

0.0000) and when reaching (n=24 / 20; effect size: 41.7% / 30.0%; 95% CI:[74.2%, 97.7%] / [58.4%, 91.9%]; p=0.0000 / 0.0073), but results were inconsistent across datasets for hurdling (Prolific favored adjacent while non-Prolific favored distant: n=130 / 14; effect size: 16.2% / 35.7%; 95% CI: [57.7%, 73.7%] / [60.1%, 96.0%]; p=0.0002 / 0.0075).

Finger sequence. Prolific participants favored typing inward sequences toward the thumb within the same row (n=208; effect size: 22.1%; 95% CI: [65.7%, 77.8%]; p=0.0000), but results were insignificant for reaches (n=426; effect size: 2.6%; 95% CI: [47.8%, 57.3%]; p=0.2865) and hurdles (n=222; effect size: 3.2%; 95% CI: [46.6%, 59.6%] p=0.3474).

Lateral stretch. Prolific participants preferred typing within the 8 finger-columns rather than laterally stretch to type center-column keys (n=804; effect size: 15.4%; 95% CI: [62.1%, 68.6%]; p=0.0000). This result confirms the choice to restrict letters to finger-column keys.

For our MOO objectives, the *lateral stretch* objective is only relevant in Step 4 (2.4.2), since Steps 1–3 restrict layouts to the home blocks. We will also combine the *column separation* and *finger sequence* objectives into a single *same row* objective, since they both have consistently significant results only for same-row key-pairs. An indiscriminate application of this *same row* objective would, however, have us treat the key-triple FDS (represented by QWERTY keys) as two left-outward key-pairs (D←F + S←D) and score it as worse than FSD, with one outward (S←F) and one inward (S→D) key-pair. However, when typing three keys in quick succession, you will quickly see that it is easier to type in one direction (S←D←F) than reversing direction midway (S←F + S→D). To overcome this intrinsic limitation of key-pair scoring, we include a trigram finger sequence score as a MOO objective. A single key-triple base score has a binary assignment, with a zero assigned if typing the key-triple requires a hand to switch the direction of its finger sequence.

The resulting six objectives are converted to the base scoring criteria in Table 4 (see Supplement 4 for details). Every possible bigram in a layout receives up to five scores and every trigram one score. Each score is the multiple of the key-pair (or -triple) base score and the letter-pair's (or triple's) normalized frequency. A layout's total score for each objective is the average objective score across all possible bigrams and trigrams.

3.4. Engram-en optimized keyboard layout for English

We will outline English keyboard layout construction results following the five steps in 2.4.2:

Step 1. Optimally assign 10 letters to 10 keys.

There are P(4,2)×P(8,8) = 483,840 possible arrangements of eight letters in 10 keys (after constraining the 2 most frequent letters within the 4 most preferred keys). Figure 8 shows one 10-letter layout out of 102 Pareto solutions found in an exhaustive search of these arrangements.

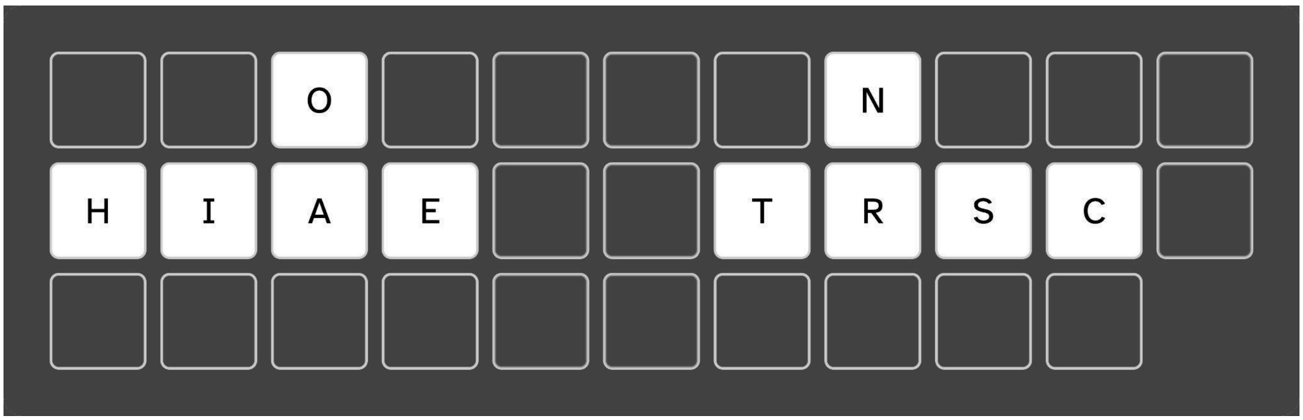


Figure 8. Step 1: Example of 10 letters assigned to 10 keys.

Step 2. Remove 2 letters and optimally assign 10 letters to fill 18 keys.

There are 96 unique 8-letter layouts after removing 2 letters from the 102 solutions from Step 1, and 96×P(10,10) = 348,364,800 possible arrangements of 10 letters to fill the 18 keys. Figure 9 shows one layout out of 101 filtered from 1,551 Pareto solutions from an exhaustive search of these arrangements. Available keys are shown in white, unavailable keys with previously assigned letters are gray, and unavailable keys with no previous assignment are black.

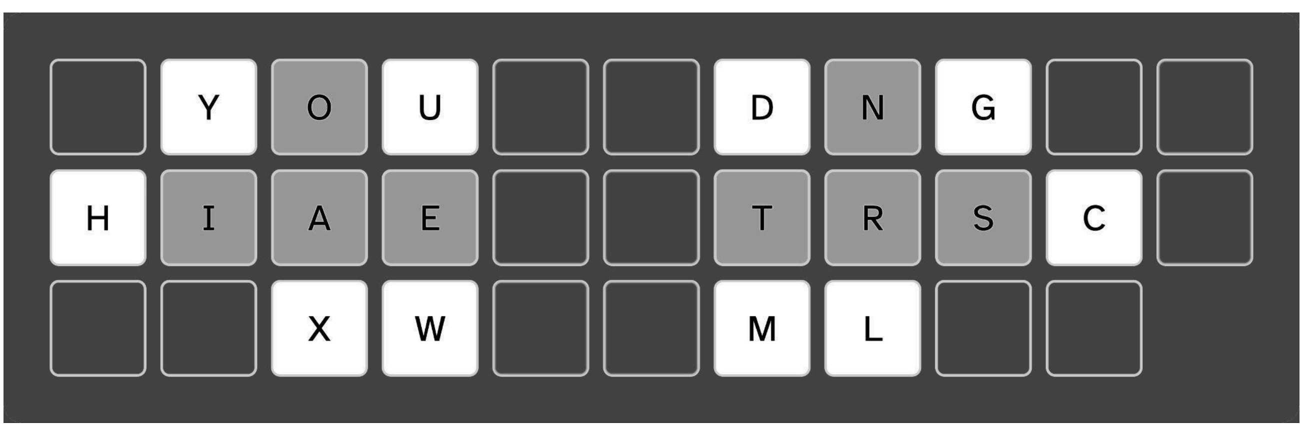


Figure 9. Step 2: Example of 10 letters assigned to fill 18 keys.

Step 3. Remove 4 letters and optimally assign 10 letters to fill 24 keys.

There are 96 unique 14-letter layouts after removing 4 letters from the 101 solutions from Step 2, and again 96×P(10,10) possible arrangements of 10 letters to fill the 24 keys. Figure 10 shows one layout out of the 582 Pareto solutions from an exhaustive search of these arrangements.

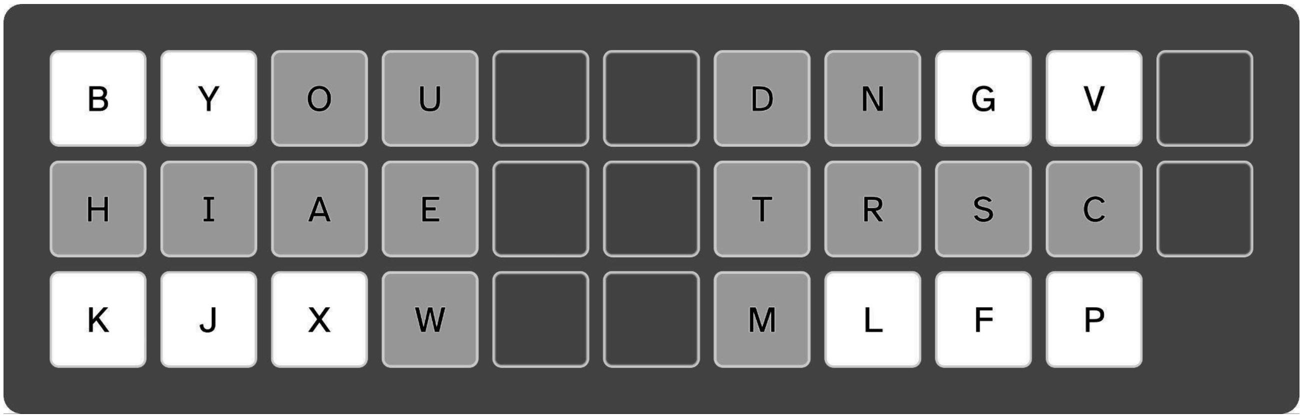


Figure 10. Step 3: Example of 10 letters assigned to fill 24 keys.

Step 4. Assign 2 letters to fill 26 keys.

Figure 11 shows the layout with the highest average score across the four objectives tested in 3.3 for the two possible placements of the final two letters. The 16 most stable letter→key assignments across the 844 layouts are all represented in this layout, suggesting some degree of convergence across the MOO solutions. A search for similar keyboard layouts on the Keyboard-design.com website (2025) surfaced 42 layouts with HIEA on the home row, but not a single one with HIAE. Given that EA is a very common bigram, coupled with a preference for inward rolls (finger sequences toward the thumb), we would expect that HIEA should be favored over HIAE. However, a careful analysis of bigram frequencies reveals that the other less frequent bigrams made from these letters collectively outweigh EA's frequency and favor the sequence HIAE.

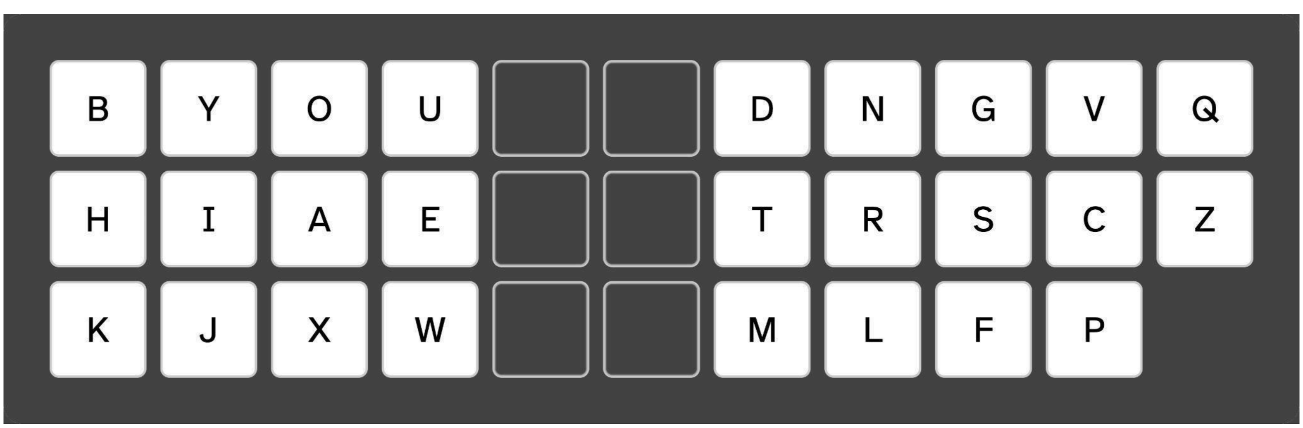


Figure 11. Step 4: Top arrangement of 26 letters.

Step 5a. Assign punctuation to the two center columns.

In addition to the 26 letters, we will address the assignment of non-letter characters, taking into account frequency of punctuation, logical grouping, and ease of recall. We first assign the most frequent punctuation (1.2) to the six keys in the center two columns (Figure 12). The left center column contains punctuation for separating and joining text and the right center column contains punctuation for closing text (explanations below). The Shift-key accesses related punctuation marks.

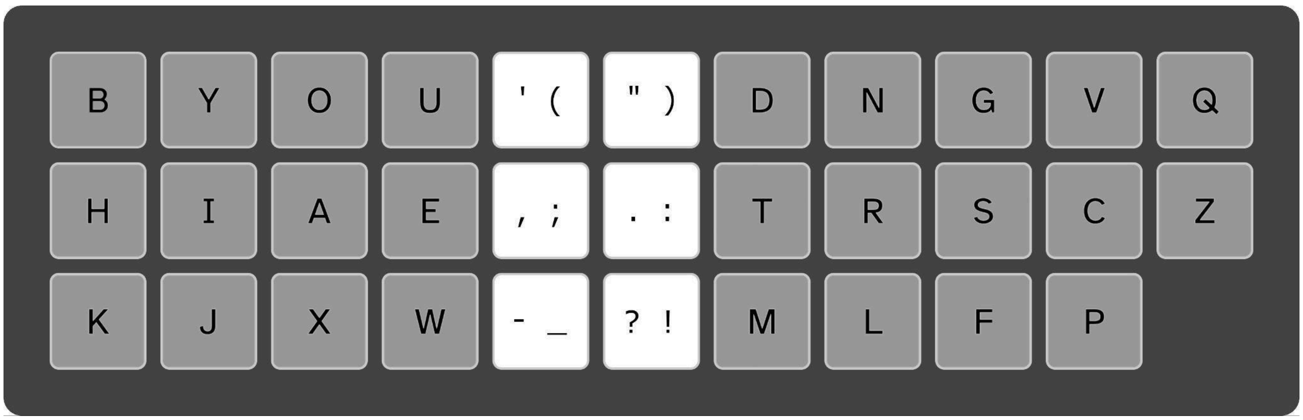


Figure 12. Layout with center-column punctuation marks.

Separating and joining marks (left middle column):
The comma separates text in lists; the semicolon can be used in place of the comma to separate items in a list (especially if these items contain commas); open parenthesis sets off an explanatory word, phrase, or sentence. The apostrophe joins words as contractions; the hyphen joins words as compounds; the underscore joins words in cases where whitespace characters are not permitted (such as in variables or file names).

Closing marks (right middle column):
A sentence usually ends with a period, question mark, or exclamation mark. The colon ends one statement but precedes the following: an explanation, quotation, list, etc. Double quotes and right parenthesis closes a word, clause, or sentence.

Step 5b. Assign symbols to the Shift-number keys.
We reserve the entire number key row for mathematical and logic symbols. We enclose the numbers with [square brackets], and the Shift-key symbols with {curly brackets}.

```
{  |  =  ~  +  <  >  ^  &  %  *  }  \
[  1  2  3  4  5  6  7  8  9  0  ]  /
```

Each number is paired with a symbol based on resemblance, except for < and >, which are placed above (and):

1. | logical OR
2. = equal (resembles the “2” character in Chinese)
3. ~ almost equal (≅ resembles “3” in Chinese)
4. + plus
5. < greater than
6. > less than
7. ^ logical XOR, exponent
8. & logical AND
9. % percent
0. * multiplication

The three remaining keys in many common keyboards (flanking the upper right-hand corner Backspace/Delete key) are displaced in special keyboards, such as the Kinesis Advantage and Ergodox. For the top right key, we will assign the forward slash and backslash (/ \). For the remaining two keys, we will assign two symbols that in modern usage have significance in social media and communications: the hash/pound sign (#) and the "at sign" (@).

The complete layout consists of all letter and non-letter key assignments (available by default or via the Shift-key, shown as left and right pairs below):

```
[{ 1| 2= 3~ 4+  5<  6>  7^ 8& 9% 0* ]} /\
   bB yY oO uU  '(  ")  dD nN gG vV qQ #$ @`
   hH iI aA eE  ,;  .:  tT rR sS cC zZ
   kK jJ xX wW  -_  ?!  mM lL fF pP
```

The result is the “Engram-en” layout (Figures 1 and 2), available for anyone to use on multiple platforms via the Keyman application. (The name “Engram” is a pun, referring both to "n-gram", letter permutations and their frequencies that are used to compute the layout, and "engram", or memory trace, the postulated change in neural tissue to account for the persistence of memory, as a nod to the attempt to make this layout easy to remember.)

3.5. Comparison of keyboard layouts

Figure 13 shows a heatmap of typing activity for the Engram-en layout generated by KLAnext using the same text corpus as in the evaluation. It is clear from the brighter green color that letters E, T, and A get the most activity, and that from the purple, the bulk of typing activity is centered on the home row and above the home row for the middle finger.

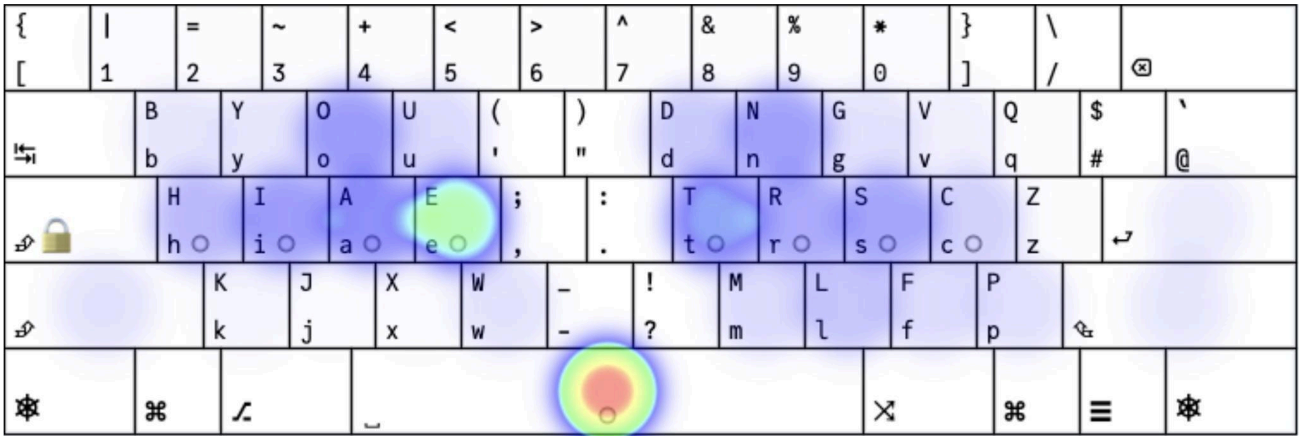


Figure 13. Finger usage heatmap for the Engram-en layout.

Figures 14 and 15 display all normalized scores in heatmap and parallel coordinates plots to compare Engram-en with the 31 layouts in Table 1. While the Engram-en layout had the highest average normalized score for the Engram Scorer metrics, as well as overall, it can be plainly seen in the parallel coordinates plot that the Engram-en layout performs worse than many other layouts on a variety of metrics, including some of the Engram Scorer objectives. Its worst relative performance is for the Dvorak criterion favoring non-adjacent fingers when typing bigrams, as well as for distance, and some single-finger metrics. These results were expected, since the Engram method for optimizing keyboard layouts intrinsically flouts these three types of measures. First, as we saw in 3.3, adjacent finger typing is actually preferred over non-adjacent finger typing, at least for same-row bigrams and reaches. This preference may reflect relative typing ease or comfort, and not efficiency. Dvorak's criteria, on the other hand, were developed principally for increasing typing efficiency, and we validated these criteria on typing speed data. Second, the empirically derived key preferences favor typing two keys off the home row with the middle fingers over typing on the home row with the little fingers. Assignment of higher frequency letters to the former will increase most distance metrics, independent of the relative strengths and lengths of the fingers. Third, since constraining layouts within the home blocks forces otherwise laterally displaced letters to be vertically stacked in finger-columns, we should expect a slight increase in single-finger metrics.

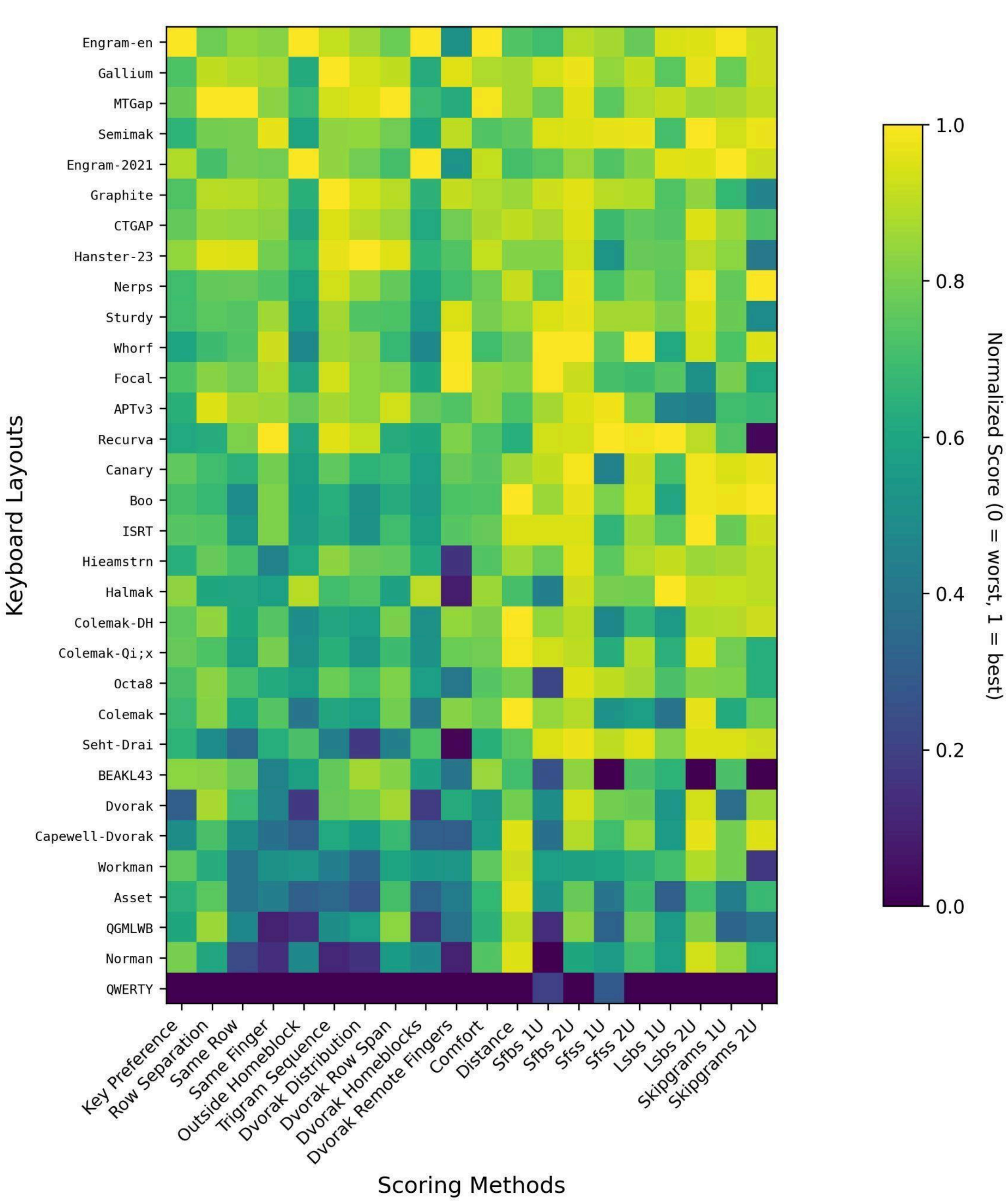


Figure 14. Heatmap comparison of 32 layouts.

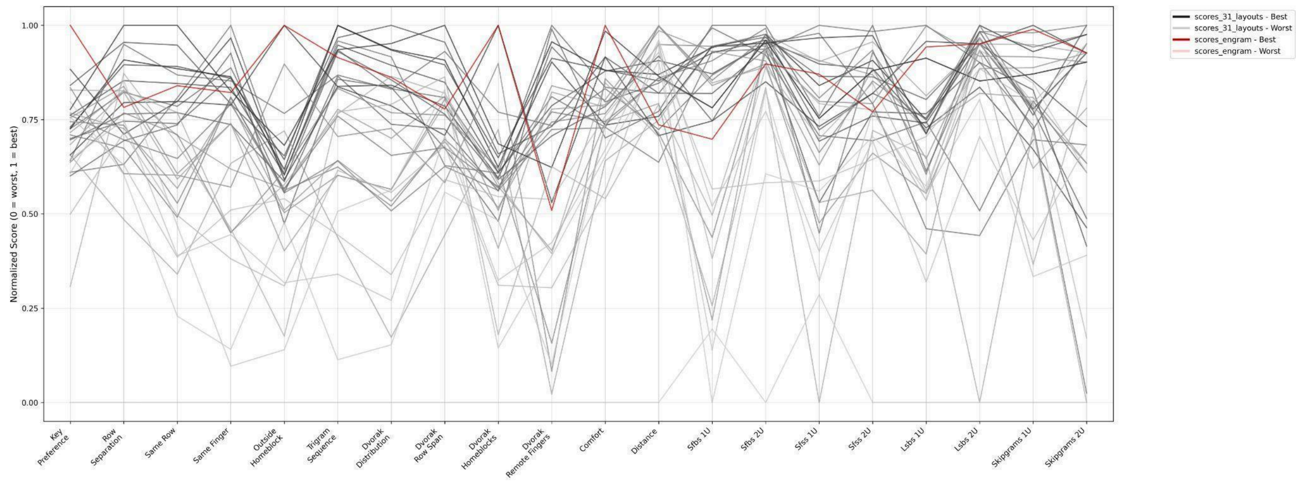


Figure 15. Parallel coordinates comparison of 32 layouts.

4. Conclusions

In this study, we introduced an open-source, crowdsource data-driven approach to establishing and applying typing ergonomics principles and scoring criteria called Engram to optimize keyboard layouts. This approach is based on language-dependent usage frequencies of letters, letter-pairs, and letter-triples, and on language-independent key-pair typing preference data. We collected typing data, converted typing preferences to ergonomics objectives, and implemented layout optimization and evaluation software that can be applied to keyboard layouts in almost any language. We applied our approach to optimize keyboard layouts for English (Engram-en: engram-layouts.xyz/engram-en) and Spanish (Engram-es: engram-layouts.xyz/engram-es; see Appendix 2). Both layouts can be previewed online and installed on Linux, macOS, and Windows. All software, documentation, and layouts are open source and freely available (see Data and Software Availability).

We used multiple scoring methods to compare Engram-en with 31 other English keyboard layouts. The Engram-en layout performed well in our analysis, but more importantly, the differences in performance between the Engram-en and other layouts for different metrics may indicate that some of the criteria that keyboard layout enthusiasts have focused on in the past may not be in the best interest of designing the most comfortable layouts. As an indication of this, our analysis supports some but not all of the ergonomics principles advocated by Dvorak, and does not support the traditional assumption that typing speed can be used as a proxy for typing comfort.

4.1. Limitations and future work

There are several limitations to the present study that warrant (encourage!) future efforts in optimizing keyboard layouts. Before listing these limitations, it is important to emphasize three caveats. First, the keyboard is not an efficient means of interacting with the computer, and better methods will supplant it in the future. Second, optimizing a keyboard layout may have limited gains compared to optimizing a physical keyboard design to better accommodate the shape, strength, positioning, and kinematics of the hands, wrists, and fingers. Third, the keyboard is only one component of typing that can impact one's health, well-being, and productivity. Other factors include those related to the computer setup and environment, such as furniture, lighting, display, operating system, software interfaces, and peripheral devices. Still other factors include physical health, psychosocial dimensions, posture, accessibility factors, visual aids, typing durations, and rest breaks.

The principal limitation to the current study is that unsupervised, subjective self-reports of any kind, let alone about preferences between unfamiliar typing patterns, is subject to possible confounds. The fact that the data quality was much better when collected from participants who were solicited via email vs. from anonymous participants paid through the Prolific platform suggests that compliance, motivation, attention, and distraction really matter for this task. For example, participants were directed to take part in the study only if they agreed to touch type on a QWERTY computer keyboard, but some participants could possibly have done otherwise, such as tapping with their thumbs on their phone screen. While it seems unlikely that participants would systematically report preferences at random or opposite of what they actually preferred, we analyzed the data to look for such activity, by including comparisons with one key-pair that was clearly more difficult to type than the other, and by looking for systematic choices that would be statistically extremely unlikely. Much more likely is that participants were left to guess when they couldn't decide which key-pair they preferred to type. To account for this, we analyzed how consistently they selected one key-pair over another across presentations. Some participants reported being confused about the instructions, but given that they all had to correctly type text on dozens of screens to submit their results, this may have been a minor and fleeting issue from a data quality standpoint.

As with any study, the results raise yet more questions that demand more data be collected. Our data for this study focused on predominantly right-handers typing with their left hand on the left side of the keyboard. Since it is a safe bet that every participant typed on a traditional staggered/diagonal keyboard, the asymmetry between the physical arrangement of keys on the left and right sides is a confound that questions the degree to which we can assume bilateral symmetry of the results. We did not collect participant preferences between various factors, including 1-handed vs. 2-handed key-pairs. Such data will one day inform more comprehensive and more accurate scoring systems underlying optimization algorithms and evaluation protocols.

The study did not test the influence of different characteristics of physical keyboards or their keys, such as shape, size, separation, orientation, or response characteristics, nor test virtual keyboards on touch screens or otherwise, and did not consider idiosyncratic typing styles. Instead, we assumed that participants followed instructions and employed standard touch typing methods using all eight fingers, each mapped to specific columns of QWERTY keys. We further assumed that people typed on familiar, conventional, flat, diagonally arrayed (staggered) computer keyboards.

We also do not have longitudinal, clinical data to directly substantiate the claim that the specific layouts herein would reduce musculoskeletal disorders. Instead, we claim that subjective accounts of what is easier or more comfortable to type is useful information to guide the design of comfortable keyboard layouts. We envision in the future remotely tracking individual typing behavior coupled with contextualizing self-reports as part of a longitudinal study. Tracking typists' behavior and physiology could take the form of passive sensors that monitor movement, electrodermal activity, heart and breathing rate, etc., or in the form of software that monitors, for example, error rates and fatigue through keystroke logging. Self-reports can take the form of ecological momentary assessments administered by a mobile device to remotely monitor contextual factors such as in-the-moment thoughts, behaviors, and psychosocial dimensions (Klein et al., 2021; https://www.gettingcurious.com).

Acknowledgments

This article is the culmination of years of effort devoted to studying and optimizing keyboard layouts, so there are plenty of people and resources I wish to acknowledge.

First, I would like to thank all of the people who have been so supportive over the years. To my family, friends, colleagues, and online participants and discussants for supporting this endeavor, whether by funding the GoFundMe campaign, volunteering as online participants for this study, or by engaging in helpful discussions. To Jack Grinband and Ian Douglas in particular for their helpful discussions. The Spanish version would not have been possible were it not for Ian Douglas cleaning up the Leipzig Spanish corpus and computing Spanish character and bigram frequencies. To those who have tried out earlier versions of my keyboard layouts and to those who have expressed interest in seeing Engram applied to different languages in the future. I would also like to express my gratitude to all of my predecessors who have made valiant efforts to improve our relationship with computers by advancing the ergonomics of keyboard designs and keyboard layouts.

Second, I would like to express my gratitude for the resources that have made this work possible. The NSF and the Pittsburgh Supercomputing Center (PSC) have generously provided Bridges-2 computing resources through allocation MED250010 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296 (Brown et al., 2021). The Keyman tools and community help to make distribution of the Engram (and thousands of other) layouts very convenient. Anthropic's Claude has served as a close coding partner of late, accelerating the pace of this work n-fold. I must also express my appreciation to that damned DEC workstation at the MIT Media Lab that introduced me to repetitive strain injury 30 years ago, which has prompted me over the years to experiment with voice dictation, one-handed and keyless keyboards, foot pedals, foot mouse, and various ortholinear ergonomic keyboards like the Kinesis Advantage and Ergodox. This article was comfortably typed using the Engram-en and a direct predecessor to the Engram-en layout via Keyman on Kinesis Advantage, Kinesis 360, and (macos and linux) laptop keyboards.

Funding details

This work was independently supported by the author and through a GoFundMe campaign.

Disclosure statement

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Data availability statement

The entire Engram study — all data collection, layout optimization, and evaluation software and data — is available on a public Open Science Framework repository (https://osf.io/63cy8/, DOI: 10.17605/OSF.IO/63CY8). All updated versions of the software are available on public GitHub repositories listed below, under an MIT or Apache v2.0 License.

Engram keyboard layouts (CC-BY License):

- Engram-en (English):
    - Website: engram-layouts.xyz/engram-en
    - Keyman: https://keyman.com/keyboards/engram-en
- Engram-es (Spanish):
    - Website: engram-layouts.xyz/engram-es
    - Keyman: https://keyman.com/keyboards/engram-es

Data collection:

“Typing Preference Study” JavaScript web app and Python code to collect and analyze timing and preference data (Apache v2.0 License):

- GitHub: https://github.com/binarybottle/typing_preference_study
- OSF: https://osf.io/jmha3/ (DOI: 10.17605/OSF.IO/JMHA3)

Layout optimization:

“Optimize keyboard layouts” Python code for general-purpose layout optimization (MIT License):

- GitHub: https://github.com/binarybottle/optimize_layouts
- OSF: https://osf.io/am4zt/ (DOI: 10.17605/OSF.IO/AM4ZT)

Evaluation:

- “Keyboard layout scorers” Python code implementing keyboard layout scoring methods:
    - GitHub: https://github.com/binarybottle/score_keyboard_layouts
    - OSF: https://osf.io/6dt75/ (DOI: 10.17605/OSF.IO/6DT75)
- “Process 3.5M keystrokes” Python code to extract bigram typing speed data from the 136 Million Keystrokes Dataset:
    - GitHub: https://github.com/binarybottle/process_3.5M_keystrokes
    - OSF: https://osf.io/6t9dn/ (DOI: 10.17605/OSF.IO/6T9DN)
- “Typing preferences to comfort scores” Python code to implement Bayesian preference learning (MIT License):
    - GitHub: https://github.com/binarybottle/typing_preferences_to_comfort_scores

References

"Alt Keyboard Layouts" Discord server. https://discord.gg/keyboard-layouts (Accessed October 10, 2025).

Amell, T. K., and Kumar, S. (2000). Cumulative trauma disorders and keyboarding work. *International journal of industrial ergonomics*, *25*(1), 69-78.

Armstrong, T.J., Foulke, J.A., Martin, B.J., Gerson, J., Rempel, D.M. (1994). Investigation of applied forces in alphanumeric keyboard work. *Am Ind Hyg Assoc J.* 1994; 50: 30-35.

Baker, N.A., Cham, R., Cidboy, E.H., Cook, J., Redfern, M.S. (2007). Kinematics of the fingers and hands during computer keyboard use. *Clinical Biomechanics* 22, 34–43.

Brown, ST, Buitrago, P, Hanna, E, Sanielevici, S, Scibek, R, Nystrom, NA (2021). Bridges-2: A Platform for Rapidly-Evolving and Data Intensive Research. In *Practice and Experience in Advanced Research Computing* (pp 1-4). DOI:10.1145/3437359.3465593

Bufton, M.J., Marklin, R.W., Nagurka, M.L., Simoneau, G.G. (2006). Effect of keyswitch design of desktop and notebook keyboards related to key stiffness and typing force. *Ergonomics* 49:10, 996-1012, DOI:10.1080/00140130600577437

Çakir, A. (1995). Acceptance of the adjustable keyboard. *Ergonomics* 38:9, 1728-1744, DOI:10.1080/00140139508925223

CDC. "Work-Related Musculoskeletal Disorders & Ergonomics". Centers for Disease Control and Prevention. https://www.cdc.gov/workplacehealthpromotion/health-strategies/musculoskeletal-disorders/index.html (Accessed September 20, 2021).

Cook, V.J. (2013). Standard punctuation and the punctuation of the street. In M. Pawlak and L. Aronin (eds.), *Essential Topics in Applied Linguistics and Multilingualism*, Springer International Publishing Switzerland (2013), 267-290.

Cooper, K. and Campbell-Kyureghyan, N., Sommerich, C. (2009). Comparative study of computer usage patterns and prevalence of musculoskeletal symptoms among college students. *Human Factors and Ergonomics Society Annual Meeting Proceedings* 53, 615-619. DOI:10.1518/107118109X12524442636021.

Dhakal, V., Feit, A.M., Kristensson, P.O., Oulasvirta, A. Observations on Typing from 136 Million Keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018.

Dept. of Labor. "Survey of Occupational Injuries and Illnesses Data". U.S. Bureau of Labor Statistics report. (2019). https://www.bls.gov/iif/soii-data.htm

Deshwal, P.S. and Deb, K. (2006). Ergonomic Design of an Optimal Hindi Keyboard for Convenient Use. *2006 IEEE International Conference on Evolutionary Computation*, 2187-2194.

Dhakal, V., Feit, A.M., Kristensson, P.O., Oulasvirta, A. (2018). Observations on Typing from 136 Million Keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018.

Douglas, I. (2021a, March 28). “Keyboard Layout Analysis: Creating the Corpus, Bigram Chains, and Shakespeare's Monkeys (Version 1.0.0)”. Zenodo. http://doi.org/10.5281/zenodo.4642460

Douglas, I. (2021b, September 5). “Creating a corpus and chained bigrams for Spanish keyboard development and evaluation”. Zenodo. http://doi.org/10.5281/zenodo.5501914

Dvorak, A. and Dealey, W.L. (1931-1935). *Unpublished Experiments*. University of Washington.

Dvorak, A. and Dealey, W.L. (1936). “Typewriter keyboard”. US Patent 2,040,248.

Dvorak, A., Merrick, N.L., Dealey, W.L., and Ford, G.C. (1936). *Typewriting Behavior: Psychology applied to teaching and learning Typewriting*. American Book Company, New York, USA.

Eggers, J., Feillet, D., Kehl, S., Wagner, M. O., & Yannou, B. (2003). Optimization of the keyboard arrangement problem using an ant colony algorithm. *European Journal of Operational Research*, *148*(3), 672-686.

Fagarasanu, M., and Kumar, S. (2003). Carpal tunnel syndrome due to keyboarding and mouse tasks: a review. *International Journal of Industrial Ergonomics*, *31*(2), 119-136.

Ferguson, D., and Duncan, J. (1974). Keyboard design and operating posture. *Ergonomics*, *17*(6), 731-744.

Garza, J.L.B., Eijckelhof, B.H.W., Johnson, P.W., Raina, S.M., Rynell, P.W., Huysmans, M.A., van Dieën, J.H., van der Beek, A.J., Blatter, B.M., Dennerlein, J.T. (2012). Observed differences in upper extremity forces, muscle efforts, postures, velocities and accelerations across computer activities in a field study of office workers, Ergonomics, 55:6, 670-681, DOI:10.1080/00140139.2012.657692

Gerard, M.J., Jones, S.K., Smith, L.A., Thomas, R.E., Wang, T. (1994). An ergonomic evaluation of the Kinesis Ergonomic Computer Keyboard, Ergonomics, 37:10, 1661-1668, DOI:10.1080/00140139408964943

Gerr, F., Marcus, M., Ensor, C., Kleinbaum, D., Cohen, S., Edwards, A., Gentry, E., Ortiz, D.J., Monteilh, C. (2002). A prospective study of computer users: I. Study design and incidence of musculoskeletal symptoms and disorders. *American Journal of Industrial Medicine* 41, 221–235.

Getreuer, P. (2025a). “Stats table comparing alt keyboard layouts”. https://getreuer.info/posts/keyboards/alt-layouts/stats.html (Accessed October 10, 2025)

Getreuer, P. (2025b). “Keyboard Layouts doc (3rd edition)”. https://docs.google.com/document/d/1W0jhfqJI2ueJ2FNseR4YAFpNfsUM-_FlREHbpNGmC2o (Accessed October 10, 2025)

Gillespie, P., Douglas, I. Keyboard Layout Analyzer: KLAnext v0.06. https://klanext.keyboard-design.com/#/config (Accessed October 10, 2025)

Goettl, J.S., Brugh, A.W., Julstrom, B.A. (2005). Call me e-mail: arranging the keyboard with a permutation-coded genetic algorithm. In *Proceedings of the 2005 ACM symposium on Applied computing* (pp. 947-951).

Keyboard-design.com’s Homerow explorer. https://www.keyboard-design.com/homerow-explorer.html (Accessed October 15, 2025)

Hopkins, A. (1990). Stress, the quality of work, and repetition strain injury in Australia. *Work & Stress* 4 (2), 129–138.

İşeri, A., Ekşioğlu, M. (2015). Estimation of digraph costs for keyboard layout optimization. *International Journal of Industrial Ergonomics*, Volume 48, 2015, Pages 127-138. DOI:10.1016/j.ergon.2015.04.006.

Keyboard Layout Analyzers. Google Docs spreadsheet. https://docs.google.com/spreadsheets/d/1yBwRpn3Gkp8qHje_cwDj6tcZ2ReqC9rfvsioXzt1xdg (Accessed October 10, 2025)

Khorshid, E., Alfadli, A. and Majeed, M. (2010). A new optimal Arabic keyboard layout using genetic algorithm, *Int. J. Design Engineering*, Vol. 3, No. 1, pp.25–40.

Klein, A., Clucas, J., Krishnakumar, A., Ghosh, S.S., Van Auken, W., Thonet, B., Sabram, I., Acuna, N., Keshavan, A., Rossiter, H., Xiao, Y., Semenuta, S., Badioli, A., Konishcheva, K., Abraham, S.A., Alexander, L.M., Merikangas, K.R., Swendsen, J., Lindner, A.B., Milham, M.P. (2021). Remote Digital Psychiatry: MindLogger for Mobile Mental Health Assessment and Therapy. *J Med Internet Res*. DOI:10.2196/22369

Krzywinski, M. (2015). “Typing Effort Model” in “Carpalx”. http://mkweb.bcgsc.ca/carpalx/?typing_effort (Accessed December 1, 2020)

Ladany, S. P. (1975). A model for optimal design of keyboards. *Computers & Operations Research* 2(1), 55-59.

Layouts Wiki. https://layouts.wiki/ (Accessed October 10, 2025)

Lee, D.L., Kuo, P.L., Jindrich, D.L., Dennerlein, J.T. (2009). Computer keyswitch force–displacement characteristics affect muscle activity patterns during index finger tapping. *Journal of Electromyography and Kinesiology* 19(5), 810-820.

Lee, X. (2013). “Computer Languages Character Frequency”. Date: 2013-05-23. Last updated: 2020-06-29. http://xahlee.info/comp/computer_language_char_distribution.html (Accessed Dec. 1, 2020)

Light, L.W. and Anderson, P.G. (1993). Typewriter Keyboards via Simulated Annealing. *AI Expert.*

Lopez, M.S. (1993). *An ergonomic evaluation of the design and performance of four keyboard models and their relevance to carpal tunnel syndrome*. PhD thesis, Texas A&M University

Long, M.M. (1994). Kinematics of the fingers during typing. *Master of Science thesis* (page 112), Virginia Polytechnic Institute and State University.

Malas, T.M., Taifour, S.S., Abandah, G.A. (2008). Toward optimal Arabic keyboard layout using genetic algorithm. *Proc. 9th Int’l Middle Eastern Multiconf. on Simulation and Modeling (MESM 2008)*, 50-54.

Malik, S. and Findlater, L. (2013). Punctuation input on touchscreen keyboards: analyzing frequency of use and mode-switching costs. University of Maryland, College Park. HCIL-2013-17.

Martin BJ, Armstrong TJ, Foulke JA, et al. (1996). Keyboard reaction force and finger flexor electromyograms during computer keyboard work. *Human Factors.* 1996;38(4):654-664. doi:10.1518/001872096778827288

Martin, J. (1981). A study of keying skills and various alphanumeric keyboards. PhD Thesis. Loughborough University. https://repository.lboro.ac.uk/articles/thesis/A_study_of_keying_skills_and_various_alphanumeric_keyboards/9360608 (Accessed Dec. 1, 2020)

McLoone, H.E., Jacobson, M., Clark, P., Opina, R., Hegg, C., Johnson, P. (2009). Design and evaluation of a curved computer keyboard, *Ergonomics* 52:12, 1529-1539, DOI:10.1080/00140130903215321

Nemshilov, N. (2016). “Halmak Keyboard Layout”. GitHub. https://github.com/MadRabbit/halmak (Accessed December 1, 2020)

NIOSH. Musculoskeletal disorders and workplace factors: A critical review of epidemiologic evidence for work-related musculoskeletal disorders of the neck, upper extremity, and low back. U.S. Department of Health and Human Services (National Institute for Occupational Safety and Health) Publication Number 97-141. (July 1997). https://www.cdc.gov/niosh/docs/97-141/

Norvig, P. (2012). “English Letter Frequency Counts: Mayzner Revisited or ETAOIN SRHLDCU” http://www.norvig.com/mayzner.html (Accessed Dec. 1, 2020)

Noyes, J. (1983). The QWERTY keyboard: a review. *International Journal of Man-Machine Studies* 18(3), 265–281. doi:10.1016/s0020-7373(83)80010-8

Onsorodi AHH and Korhan O. (2020). Application of a genetic algorithm to the keyboard layout problem. *PLoS ONE* 15(1): e0226611. doi:10.1371/journal.pone.0226611

Pacheco, G., Palmeira, E, Yamanaka, K. (2020). Using genetic algorithms to design an optimized keyboard layout for Brazilian Portuguese. In: *Encontro Nacional de Inteligencia Artificial e Computacional (ENIAC)* 17, 2020, Evento Online. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 437-448. doi:10.5753/eniac.2020.12149

Radwin, R.G., Ruffalo, B.A. (1999). Computer key switch force-displacement characteristics and short-term effects on localized fatigue. *Ergonomics* 42(1), 160-170.

Reddit Corpus (small). https://convokit.cornell.edu/documentation/reddit-small.html (Accessed October 10, 2025)

Rempel, D. (1992). Work-related cumulative trauma disorders of the upper extremity. *Journal of the American Medical Association* 267, 838. DOI:10.1001/jama.1992.03480060084035.

Rempel, D., Serina, E., Klineberg, E., Martin, B.J., Armstrong, T.J., Foulke, J.A., Natarajan, S. (1997). The effect of keyboard keyswitch make force on applied force and finger flexor muscle activity. *Ergonomics* 40(8), 800–808. https://doi.org/10.1080/001401397187793

Rempel, D., Barr, A., Brafman, D., & Young, E. (2007). The effect of six keyboard designs on wrist and forearm postures. *Applied ergonomics 38*(3), 293-298.

Rempel, D., Nathan-Roberts, D., Chen, B.Y., Odell, D. (2009). The effects of split keyboard geometry on upper body postures. *Ergonomics* 52:1, 104-111, DOI:10.1080/00140130802481040

Rose, M. J. (1991). Keyboard operating posture and actuation force: Implications for muscle over-use. *Applied Ergonomics 22*(3), 198-203.

Ruhlen, H., and Pressey, S. (1924). A Statistical Study of Current Usage in Punctuation. *The English Journal* 13(5), 325-331. doi:10.2307/802253

Salvo J.M.R., Raagas C.J.B., Medina M.T.C.M., Portus A.J.A. (2016). Ergonomic Keyboard Layout Designed for the Filipino Language. In: Goonetilleke R., Karwowski W. (eds) *Advances in Physical Ergonomics and Human Factors. Advances in Intelligent Systems and Computing* v.489. Springer, Cham. doi:10.1007/978-3-319-41694-6_41

Serina, E.R., Tal, R., Rempel, D. (1999). Wrist and forearm postures and motions during typing. *Ergonomics* 42:7, 938-951, DOI:10.1080/001401399185225

Sommerich, C.M. (1994). *A biochemical analysis of repetitive finger motion during typing activities.* PhD thesis, The Ohio State University.

Sommerich, C.M., Marras, W.S., Parnianpour, M. (1996). A quantitative description of typing biomechanics. *J Occup Rehab* 6, 33–55. DOI:https://doi.org/10.1007/BF02110393

Sun, Kun & Wang, Rong. (2018). “Frequency distributions of punctuation marks in English: Evidence from large-scale corpora.” English Today. doi:10.1017/S0266078418000512

Wagner, M. O., Yannou, B., Kehl, S., Feillet, D., & Eggers, J. (2003). Ergonomic modelling and optimization of the keyboard arrangement with an ant colony algorithm. *Journal of Engineering Design* 14(2), 187–208. doi:10.1080/0954482031000091509

Walker, C.P. (2003). “Evolving a more optimal keyboard”. Course project: Introduction to evolutionary computation, Missouri University of Science & Technology.

Koichi, Y. and Motoko, Y. (2011). On the Prehistory of QWERTY. *ZINBUN* 42 161-174. DOI:10.14989/139379. Website: https://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/139379/1/42_161.pdf

Yin, P.Y. and Su, E.P. (2011). Cyber Swarm optimization for general keyboard arrangement problem. *International Journal of Industrial Ergonomics 41*(1), 43-52.

Zecevic, A., Miller, D.I., Harburn, K. (2000). An evaluation of the ergonomics of three computer keyboards. *Ergonomics* 43:1, 55-72, DOI:10.1080/001401300184666

Zipp, P., Haider, E., Halpern, N., & Rohmert, W. (1983). Keyboard design through physiological strain measurements. *Applied Ergonomics 14*(2), 117-122.

Appendix 1: Dvorak Scoring System

In this Appendix, we extract scoring criteria for evaluating keyboard layouts from Dvorak's "Typing Behavior" book (Dvorak, et al., 1936) and "Dvorak-Dealey 'Simplified' Typewriter Keyboard" patent (Dvorak and Dealey, 1936).

A1. Dvorak's keyboard layout evaluation principles

Dvorak's patent provides a clear rationale for inventing a new keyboard layout, to:

> "increase…operating speed by eliminating awkward sequences…[and reducing] errors", and lessen "fatigue of the typist", all through a "better arrangement of the keys for typing the sequences most frequently used, and the rhythmical flow of typing induced thereby, and because of more evenly distributed labor for the individual fingers and the two hands."

However, neither Dvorak's book nor patent outlines an explicit procedure for translating his observations of typing behavior into the design or evaluation of a keyboard layout, let alone an optimization algorithm. On pages 217-218 and 221 flanking the figure of the patented layout, he cites unpublished experiments:

> Isn't it obvious that faster, more accurate, less fatiguing typing can be attained in much less learning time provided a simplified keyboard is taught? Studies by Dvorak and Dealey,[27] accordingly, employing sequence counts that are statistically adequate, have led to a simpler design of key locations. This design fits hand-stroking skills to the sequence superior operating ease that naturally flows from a "simplified" design.[28] …The "simplified" keyboard is the inevitable result of the application of the Gilbreth efficiency engineering analysis of motion study to typewriting.
>
> [27]Dvorak, August and Dealey, W. L., Unpublished Experiments, University of Washington, 1931-1935.
>
> [28]Dvorak, August and Dealey, W. L., "Simplified Keyboard Arrangement," U. S. Patent Office, Serial No. 612,738, 1932.

We can infer evaluation principles from his critiques of and improvements on the QWERTY layout. In the patent, he makes it clear how he would prioritize these behaviors, at least from the standpoint of speed and efficiency He primarily focused on behaviors that increase typing time and introduce awkward sequences:

Increasing time:

1. 2 hands, same row (ex: EI (2 time-units))
2. 2 hands, upper and lower rows (ex: EM (4 time-units))
3. 1 hand, adjacent fingers (ex: adjacent row: ES (5 time-units))
4. 1 hand, remote fingers, hurdle home row (ex: ON (10 time-units))
5. 1 hand, same finger (ex: adjacent row: DE (11 time-units))

Awkward sequences:

- 1 hand, adjacent fingers, except strong fingers: AS, WQ
- 1 hand, skip home row (hurdles): ON, IM, NY
- 1 hand, different rows (reaches)
- 1 hand, same finger: DE
- 1 hand, same finger, lateral movement: FT, RT

Preferable sequences:

- inward roll (strum)
- strong fingers

A2. Deriving scoring criteria for evaluating bigrams

We can derive good and bad typing behaviors from Dvorak's work and create a scoring system to optimize or evaluate keyboard layouts. Below, we distill these behaviors and apply them to the case of key-pair finger transitions when touch typing bigrams, since in his patent he calls out the key-pair as the fundamental unit to quantify: "The stroking unit, accordingly, is not an isolated key-stroke, but embraces a complete interval of time between two successive strokes." Bigram typing behaviors may be evaluated by the following **7 criteria**:

1. Distribution: Typing with 2 hands or 2 fingers
2. Strength: Typing with stronger fingers (middle and index)
3. Home row: Typing within the home/middle row
4. Row separation: Typing in the same row, reaches, and hurdles
5. Finger columns: Typing within the 8 finger columns
6. Remote fingers: Typing with non-adjacent fingers (except stronger fingers)
7. Inward direction: Finger sequence toward the thumb

1. Distribution across hands and fingers.
The first criterion penalizes the repetitive use of the same hand or finger. Alternating use of the left and right hands reduces load, increases speed, and improves rhythm:

> We have discovered the desirability of distributing the keys to the two hands, in such manner that the stroking of common sequences involves the use of alternate hands, insofar as possible. Being in opposite hands, reaches and hurdles are automatically eliminated. [patent]

When typing with the same hand, Dvorak reserves the greatest opprobrium for repeated typing with the same finger:

> Even though idling by your entire hand is far more extensive and serious, the occasional forcing of one finger to type an entire sequence inflicts very striking and indefensible damage to speed and rhythm. Any bigram stroked with the same finger obviously is one of the slowest, poorest sequences in typing. [p. 215]

2. Finger strength. The second criterion emphasizes the use of the stronger fingers (index and middle) fingers:

> ...our new method of keyboard revision launches itself for study of sequences and of the possibilities for speed and accuracy gained by increasing the number of sequences stroked by… the first and second adjacent fingers of the right hand. [patent]
>
> Our keyboard arrangement has the following advantages:... 7. A reduced number of bigrams involves both the fifth and fourth fingers only of both hands (1% as compared with 3%). [patent]
>
> … the more dominant bigrams appear to hold down errors and facilitate speed despite handicaps of the present haphazard keyboard. The first finger patterns, for example, are dominated by the terrific frequencies of *th, on, an, he, in, nd, ha, at, en*. Because of these keystrokes, *t, h, n,* participate in very dominant, common sequences their relative efficiency is high despite keyboard handicaps. [patent]

3. Home row. This is the only criterion to favor a location (the home row), rather than relative position or sequence:

> We have discovered the necessity, to avoid breaks in rhythm even in common sequences, of forming as many such sequences as possible in the home row. This avoids dropping to the lower bank and returning therefrom, which is very slow, and stepping up to the upper row and recovering, which is somewhat faster but still slower than sequences in the home row. [patent]

4. Row separation. Dvorak makes many references to “awkward hurdles and reaches” and clearly prefers sequences typed in the same row. Dvorak is clear that it is preferable to type within the same row (even when alternating hands). Consequently, high-frequency letter-sequences should not span rows. Reaching from one row to an adjacent row is considered bad, and skipping over (“hurdling”) the home row is worse:

> ...what is your prime notion of awkward fingering? Undoubtedly, it is outright finger hurdling to keys in upper and lower banks. In general, the crowding of most common fingering combinations out of the home row and into one hand has an unforeseen result. The "guide" keys must be abandoned while the fingers play between upper and lower banks of keys. [p. 216]

5. Finger columns. When touch typing letters on almost any physical keyboard, all eight fingers are expected to move from the home row to the keys directly above and below (“home block” keys), with minimal lateral stretching for conventional, staggered keyboard designs. Index fingers must stretch laterally to type keys in the two middle columns. Dvorak recognized that this is an awkward movement. The patent excerpt above that lists unfavorable bigrams includes the “invariably slow and difficult…movements by a single finger laterally”.

Due to the natural roundedness of the hand, others have considered the most easily accessible keys from the home row to include the home row keys themselves, the top-center keys for each hand that allow the longer middle and ring fingers to uncurl upwards as well as the bottom corner keys that allow the shorter little and index fingers to curl downwards (Long, 1994).

6. Remote fingers. This criterion penalizes the use of adjacent fingers for reaches and hurdles. When using different fingers on the same hand, Dvorak specifies that typing with non-adjacent fingers is preferable to typing with adjacent fingers with exceptions for the stronger fingers:

> Scarcely less objectionable than single-finger reaches and hurdles are those requiring the use of adjacent fingers. These, too, can be classified as naturally awkward. [patent]

7. Inward direction. This is the only criterion to prefer a sequence or direction. Dvorak favors the ease and speed with which one can strum from outer to inner fingers (toward the thumb) on a surface as compared to strumming with a finger sequence away from the thumb:

> Likewise, we have discovered, and employ as a factor in determining the location of the keys, that sequences in one hand (which cannot be wholly avoided) are most satisfactorily made when stroked from an outer finger inwardly. It is easier to tap in rapid succession from the fifth finger to the index finger, than the reverse. [patent]

A3. Proposed Dvorak bigram scoring method

The 7 base scoring criteria in Table A1 are derived from the 7 criteria in the previous section that reflect favorable typing behaviors. When applied to a single bigram, each criterion may be given a base score of 0, 0.5, or 1 generally to indicate when 0, 1, or 2 fingers or keys satisfy the criterion. For criteria that are inherently 1-hand concepts, we don't want to penalize 2-handed typing, so we set a neutral base score since the criterion doesn't apply.

Table A1. Base scoring criteria for typing a single bigram

Scoring criterion	2 hands	1 hand	1 finger
Distribution - 1.0: 2 fingers on 2 hands to type 2 keys - 0.5: 2 fingers on 1 hand to type 2 keys - 0.0: 1 finger on 1 hand to type 1-2 keys	1	0.5	0
Finger strength - 1.0: 2 keys typed with strong finger(s) - 0.5: 1 key typed with strong finger - 0.0: 0 keys typed with strong finger	0, 0.5, or 1	0, 0.5, or 1	0 or 1
Home row - 1.0: 2 middle row keys - 0.5: 1 middle row key - 0.0: 0 middle row keys	0, 0.5, or 1	0, 0.5, or 1	0, 0.5, or 1
Row separation - 1.0: 2 keys in same row (or 2 hands) - 0.5: 2 keys in adjacent rows (reach) - 0.0: straddling home row (hurdle)	1	0, 0.5, or 1	0, 0.5, or 1
Finger columns - 1.0: 2 keys within finger column(s) - 0.5: 1 key within finger column - 0.0: 0 keys within finger column	0, 0.5, or 1	0, 0.5, or 1	0, 0.5, or 1
Remote reaches - 1.0: non-adjacent fingers, adjacent fingers on the same row, strong finger pair, or opposite hands - 0.0: same finger, or adjacent fingers where at least 1 is weak	1	0 or 1	0
Inward direction - 1.0: inward roll, or opposite hands - 0.0: outward roll, or same finger	1	0 or 1	0

For each bigram and for each criterion, the bigram's criterion score is the multiple of the key-pair base score and the letter-pair's normalized frequency. This multiple is averaged across all possible bigrams to generate a layout's criterion score. The overall Dvorak score is simply the average of the criterion scores. In the empirical validation section below, we correlate combinations of criteria with typing speeds.

A4. Empirical validation

We conducted comprehensive empirical validation of the Dvorak scoring criteria above applying our open-source software (https://github.com/binarybottle/keyboard_layout_scorers.git) on bigrams extracted (https://github.com/binarybottle/process_3.5M_keystrokes.git) from real typing performance data (136 Million Keystrokes Dataset, Dhakal et al., 2018) and using English bigram frequency data (Norvig, 2012).

Participant information

Original number of participants in the 136M Keystrokes Dataset: 168594

Filters:

- Types on a QWERTY layout: 165324 participants (98.1%)
- Types with 9-10 fingers: 107063 participants (63.5%)
- Types on a full keyboard or laptop: 165009 participants (97.9%)
- Error rate < 1.167: 103495 participants (61.4%)

All criteria combined: 69062 participants (41.0%)

Age distribution:

Mean (SD):	25.9 (11.1) years
Range:	0–120 years

Gender distribution:

Female:	33876 (49.1%)
Male:	28312 (41.0%)
None:	6874 (10.0%)

Typing course training:

Trained:	27747 (40.2%)
Untrained:	41315 (59.8%)

Country distribution:

US:	47754 (69.1%)
PH:	3823 (5.5%)
CA:	3507 (5.1%)
IN:	3238 (4.7%)
GB:	1870 (2.7%)

... and 189 other countries

Native Language Distribution:

en:	59112 (85.6%)
tl:	1743 (2.5%)
zh:	1422 (2.1%)
es:	1100 (1.6%)
hi:	589 (0.9%)

... and 139 other languages

Data

- 19,809,877 valid bigrams (413 unique) from dvorak9-scorer.git
- Average typing time: 172.2ms ± 147.3ms
- Kept 19,395,597 bigrams (97.9%) that are between 50-2000ms

A5. Analysis and results

The keyboard_layout_scorers/prep/*dvorak7_speed_validation.py* script correlates every possible combination of the 7 criteria to actual typing speed data. Since typing speed is heavily influenced by letter frequency, we try to control for this using log-transform frequency, a regression model, and a frequency-adjusted time (residual). For multiple testing corrections with 127 combinations (7+21+35+35+21+7+1 for 1-way through 7-way), we apply Benjamini-Hochberg FDR correction to control false discovery rate. The empirical analysis reveals which of Dvorak's theoretical principles actually correlate with typing speed in practice.

We found that most Dvorak principles are statistically valid but practically weak, with maximum effect sizes for individual criteria (row separation) up to $|r| = -0.1655$ ($p=0.0000$) and for combinations (distribution + row separation + column separation + remote fingers) up to $|r| = -0.1912$ ($p=0.0000$). Considering Cohen's effect size conventions (small ≥0.1, medium ≥0.3, large ≥0.5), effect sizes for individual criteria were negligible to small ($|r| < 0.2$). The only individual criterion that contradicted Dvorak typing principles was finger strength, but this was negligible.

A6. Criteria for multi-objective layout evaluation or optimization

Comprehensive empirical validation using 19.4 million correctly-typed bigrams from the 136M Keystrokes Dataset reveals that not all Dvorak criteria contribute equally to predicting typing speed, and some should be excluded from multi-objective evaluation or optimization. Testing all 127 possible combinations of the 7 criteria with frequency-adjusted typing times and FDR correction, we found that a 4-criterion model (distribution + vspan + columns + remote) achieves the strongest correlation with actual typing speed ($r = -0.1912$, $p < 0.001$), explaining 3.7% of speed variance. This 4-criterion model significantly outperforms the full 7-criterion model ($r = -0.1564$), demonstrating that three criteria—strength, middle, and inward—add noise rather than signal when used in combination.

Individual criterion analysis reveals why certain criteria should be excluded from multi-objective evaluation or optimization. Most critically, the strength criterion (typing with stronger fingers) shows a positive correlation with typing time ($r = +0.0441$, $p < 0.001$), directly contradicting Dvorak's theoretical predictions: layouts optimized for stronger fingers actually result in slower typing. The middle and inward criteria show negligible effect sizes ($|r| < 0.10$) and their inclusion in 5-way and 6-way models consistently weakens predictive power compared to the optimal 4-criterion model. The empirically validated criteria—distribution ($r = -0.1621$), vspan ($r = -0.1655$), columns (estimated $r \approx -0.05$), and remote ($r = -0.1404$)—all validate Dvorak's principles and capture complementary aspects of typing performance.

A7. Conclusion

The weak correlations suggest that Dvorak's principles explain only about 3.7% of typing speed variance ($r^2 = 0.1912^2 \approx 0.037$). For multi-objective keyboard layout optimization, we recommend using four objectives: distribution (hand/finger alternation), vspan (row separation behavior), columns (finger-column adherence), and remote (non-adjacent finger usage).

Table A2. Reduced set of base scoring criteria for typing a single bigram

Scoring criterion	2 hands	1 hand	1 finger
Distribution - 1.0: 2 fingers on 2 hands to type 2 keys - 0.5: 2 fingers on 1 hand to type 2 keys - 0.0: 1 finger on 1 hand to type 1-2 keys	1	0.5	0
Row separation - 1.0: 2 keys in same row (or 2 hands) - 0.5: 2 keys in adjacent rows (reach) - 0.0: straddling home row (hurdle)	1	0, 0.5, or 1	0, 0.5, or 1
Finger columns - 1.0: 2 keys within finger column(s) - 0.5: 1 key within finger column - 0.0: 0 keys within finger column	0, 0.5, or 1	0, 0.5, or 1	0, 0.5, or 1
Remote reaches - 1.0: non-adjacent fingers, adjacent fingers on the same row, strong finger pair, or opposite hands - 0.0: same finger, or adjacent fingers where at least 1 is weak	1	0 or 1	0

Appendix 2: Engram-es Spanish Keyboard Layout

Creating an optimized layout for a different language introduces two significant challenges: the layout may need to accommodate additional characters not commonly used in English, and our approach requires representative letter bigram frequency data for the language. For a Spanish version, we need to include additional punctuation as well as diacritical marks, and need a representative text corpus. For the latter, we used the Leipzig Spanish corpus (https://wortschatz.uni-leipzig.de/en/download/Spanish). Ian Douglas computed Spanish letter and letter bigram frequency data from a cleaned-up version of this corpus created for this study (Douglas 2021). The largest file was downloaded from each row of the original version, except for rows specified as not from Spain, and lines containing non-Spanish names and words were removed.

Engram-es, our Spanish version of the Engram layout, makes use of two special keys to type in conjunction with other keys to generate additional characters: the diacritical key and the AltGr key. The diacritical key (the center key in the right middle column in the Figure below, denoted below by a star icon) is typed by the right index finger in conjunction with a letter to produce the most common diacritical marks, such as the acute accent, the virgulilla, and the sedilla. When typed in conjunction with the Shift and/or AltGr letter keys, it produces either an umlaut, a grave accent, or a circumflex. The AltGr key is typed in conjunction with a letter or number to produce punctuation or symbols such as curly braces, comillas, em dash, backtick, interpunct, ordinals, or different currencies. In order to support both ANSI and ISO keyboards, there is an additional ISO key that needs to be expendable. This key (next to the left Shift key) is assigned a duplicate asterisk.

```
([ 1| 2= 3~ 4+  5<  6>  7^ 8& 9% 0* )] /\
   zZ hH oO bB  .:  "'  nN dD vV jJ wW -_ @#
   gG iI aA eE  ,;  ☆    rR cC sS tT kK
   xX yY uU qQ  ¿¡  ?!  lL pP fF mM

☆ + aeiouAEIOU = áéíóúÁÉÍÓÚ (acute accent)
☆ + nN = ñÑ (virgulilla)
☆ + cC = çÇ (sedilla)
☆ + Shift         + [letter] = [letter] with a diaeresis/umlaut: ü
☆         + AltGr + [letter] = [letter] with a grave accent: è
☆ + Shift + AltGr + [letter] = [letter] with a circumflex: â
AltGr + ( = { (open curly brace)
AltGr + ) = } (close curly brace)
AltGr + 5 = « (open quote/comillas)
AltGr + 6 = » (close quote/comillas)
AltGr + - = — (em dash)
AltGr + ' = ` (backtick)
AltGr + . = • (middle dot, or "interpunct")
AltGr + o = º (ordinal)
AltGr + a = ª (ordinal)
AltGr + e = € (euro currency)
AltGr + l = £ (pound currency)
AltGr + s = $ (dollar currency)
AltGr + c = ¢ (cent currency)
```

Engram-es, like the English Engram layout, is also open source (https://github.com/binarybottle/engram-es). An earlier version underwent extensive usability testing by Spanish-speaking typists.

Figure A2.1. Engram-es Spanish layout.