

Retroactive Transfer Phenomena in Alternating User Interfaces

Reyhaneh Raissi

Sorbonne Université, CNRS, ISIR
Paris, France

Evanthia Dimara

Sorbonne Université, CNRS, ISIR
Paris, France
University of Konstanz, Germany

Jacquelyn H. Berry

Cognitive Science Department,
Rensselaer Polytechnic Institute
New York, United States

Wayne D. Gray

Cognitive Science Department,
Rensselaer Polytechnic Institute
New York, United States

Gilles Bailly

Sorbonne Université, CNRS, ISIR
Paris, France

ABSTRACT

We investigated retroactive transfer when users alternate between different interfaces. Retroactive transfer is the influence of a newly learned interface on users' performance with a previously learned interface. In an interview study, participants described their experiences when alternating between different interfaces, e.g. different operating systems, devices or techniques. Negative retroactive transfer related to text entry was the most frequently reported incident. We then reported a laboratory experiment that investigated the impact of similarity between two abstract keyboard layouts, and the number of alternations between them, on retroactive interference. Results indicated that even small changes in the interference interface produced a significant performance drop for the entire previously learned interface. The amplitude of this performance drop decreases with the number of alternations. We suggest that retroactive transfer should receive more attention in HCI, as the ubiquitous nature of interactions across applications and systems requires users to increasingly alternate between similar interfaces.

Author Keywords

skill transfer; retroactive interference; keyboard layout

CCS Concepts

•Human-centered computing → User studies;

INTRODUCTION

Several studies investigated *proactive* transfer from a previously learned interface to a new interface [39, 64, 67], for instance, how quickly users can master the new AZERTY keyboard layout after mastering the previously learned QWERTY layout. However, proactive transfer neither captures the return to the original layout, i.e. the influence of the new AZERTY on the continuous learning of the QWERTY keyboard, nor the effect of regularly alternating between the two practiced layouts.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI '20, April 25–30, 2020, Honolulu, HI, USA.

© 2020 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6708-0/20/04 ...\$15.00.
DOI: <https://dx.doi.org/10.1145/3313831.3376538>

Retroactive transfer [26, 66] is the influence of a *newly* learned skill on the retention of a *previously* learned skill. In contrast with proactive transfer, retroactive transfer has received much less attention in HCI. Given the ubiquitous nature of interaction though, users often need to alternate between devices, software or techniques to complete a single task. For instance, they can regularly alternate keyboard layouts, operating systems (e.g. Mac vs. Windows) or software applications (e.g. Keynote vs. PowerPoint).

In this paper, we investigate the impact of retroactive transfer on performance of using different interfaces. We first report on an interview study to understand user experiences when alternating between interfaces (e.g., devices, applications, operating systems). Participants suggested that alternating between *keyboard layouts* is a common cause of retroactive interference; the severity of which is attributable to the degree of *similarity* between them. We then report on a controlled experiment to investigate retroactive interference when users alternate abstract keyboard layouts and how the similarity between them affects the magnitude of such interference. Our experimental design builds on standard motor learning experiments of retroactive transfer, but is extended to cover multiple alternations and to apply to a user interface context.

BACKGROUND

This section presents the stages of learning an interface as Ultimate Performance and Intramodal Improvement. Afterward, we explain the transfer of learning from previously learned interface to a new interface as proactive transfer. We then use these concepts to go one step further explaining the retroactive transfer concept. To explain retroactive transfer, we use the example of *text input* as it is one of the most common applications of such phenomena among interaction techniques [46, 69, 80], empirical studies [20, 49], models of performance [39, 41] as well as optimization methods [10, 11, 25]. The term *method* refers to the alternative user means such as interaction techniques and keyboard layouts to input text. *Keyboard layout* refers to both the arrangement of the keys (i.e. position and size) as well as the mapping between keys and characters. *Interaction techniques* refer to the interaction sequence to select characters. Figure 1 illustrates the relationships that underlie the retroactive transfer phenomena.

Ultimate Performance

Ultimate performance [21] refers to the stage where users reach an asymptote, a high level of expertise given a certain method (e.g. a keyboard layout) and task (e.g. text input) (Figure 1a₁). Ultimate performance indicates the potential benefits of a method after enough practice. For instance, the DVORAK keyboard has shown highest ultimate performance to QWERTY by 4-5% [56, 75, 16]. However, ultimate performance does not inform how quickly users learn the method.

Intramodal Improvement

Intramodal improvement [21] refers to the magnitude of performance improvement through repetition and practice with a certain method and task (Figure 1a₂). At the individual level, intramodal improvement can be divided into three phases (three-phase model): *cognitive* (understanding what to do), *associative* (learning how to do it) and *autonomous* (motor performance is mostly automatic and requires low cognitive processing) [28, 5, 66]. When data are aggregated across users, intramodal improvement is well characterized by the equation of the "power law of practice (PLP)" [55]: $y = a * P^b + c$, where y is task completion time, P is the amount of practice, a , b and c are three parameters representing respectively the amount to be learned, the rate of learning and the asymptotic selection time (i.e. ultimate performance). The performance curve has been observed in many contexts such as text editing [17, 48, 81, 80, 46, 57, 69, 82]. For instance, Shi et al. compared the method GestAKey [69] with two regular methods (hotkey and long press) and show that users reach ultimate performance faster with the GestAKey.

The acquired skills through learning and practicing may be transferred to other skills, thus interfering or improving skills. We distinguish between two types of learning transfer: *Proactive* and *Retroactive* transfer.

Proactive Transfer

Transfer of skill occurs whenever the previous skills and knowledge affects the learning or performance of a new skill. Proactive transfer (or skill transfer) [66] refers to the gain or loss of performance with the *New* method N as a result of practice with the *Previous* method P on a given task (Figure 1b). If practicing P enhances the ability of users to use N , the transfer is considered as positive. Otherwise the transfer is considered negative (also called as anterograde interference for visuomotor learning [44, 43, 45, 77]). In this case we also say that practicing P *interferes* with the learning of N ¹.

Proactive transfer [66] has received a lot of attention in HCI. Users want to build on previous knowledge when learning a new interface. When new software, OS, platform or brand is released, users have to adapt their habits. Similarly, designers often try to create more "natural" interaction by facilitating the transfer from physical to digital interaction [9].

Proactive transfer depends on several factors such as the *amount of practice*, the *rest time* [26], or the *similarity* between the methods P and N [66]. Especially for interaction

design, similarity appears to be particularly critical [66]. Polson et al. [61] showed better skill transfer in text editors when there is a high level of similarity between the two tested interfaces. In contrast, switching from a QWERTY to an AZERTY keyboard layout can introduce some interference even with a few key differences between these two designs [42, 9]. Consequently, several novel keyboard layouts manipulated the similarity (or familiarity) factor to optimize performance with the traditional QWERTY layout [80, 10, 25, 57, 11].

Several models of proactive transfer have been proposed [41, 39, 53]. Jokinen et al. [39] recently presented a model aiming to explain the negative impact of keyboard layout change on typing performance. The key aspect, regarding skill transfer, is the use of a utility learning mechanism to decide between conflicting entries (the old and the new one) in the long-term memory. In particular, the α parameter indicates how quickly the model is able to adapt to changing layouts.

RETROACTIVE TRANSFER

Proactive transfer is the influence of the *previously* learned skills on the acquisition of *new* skills. In contrast, **retroactive transfer** is the influence of the *new* skill on the acquisition of a *previously* learned skill [14, 62]. *Negative* retroactive transfer occurs when the skill learned later disrupts retrieval of the skill learned earlier. Negative retroactive transfer is also known as retroactive interference [66, 26], retroactive inhibition [13, 14, 38, 54], after-effects [50, 8], retrograde interference [43, 45, 77], catastrophic interference [65] or catastrophic forgetting [29, 31].

Retroactive transfer has been investigated in contexts including free recall [52, 13], visual perception [51], language acquisition [59], machine learning [22], motor learning of discrete [42] or continuous movements [12, 36], visuomotor learning [44], advertisement [15] and games [32]. This paper focuses on a visual motor learning task with an interactive system.

We distinguish three concepts related to retroactive transfer. (1) *Task switching* is well studied in HCI [37, 23, 30, 74], but differs from retroactive interference in that it refers to a temporary (in seconds) distraction (e.g. a notification when writing a document) rather than a learned skill. Moreover, the tasks are categorically different, though the same interface is usually used for both. In retroactive transfer the tasks can be the same, or differ little, and the interface change is the source of the confusion. The next concept, (2) *deskilling* refers to manual skill degradation through the use of automation. So users can lose their psychomotor and cognitive skills (e.g. lose of manual driving skills when operating advanced automated vehicles [72]) [18, 19]. Unlike deskilling, in retroactive interference both tasks are manually controlled. Finally, (3) *Tetlag* [32, 33] has been inspired by the Tetris game community, Tetlag refers to the brief period of confusion when switching between different versions of the popular game. Some gamers regularly alternate Tetris versions, just as some users regularly alternate computer operating systems, e.g. Mac and Windows, resulting in a brief adjustment period. A key difference from retroactive interference is that the two interfaces in Tetlag are *both very well-learned* and their alternation is a common occurrence.

¹Zero transfer is also possible when the practice of P has no effect on the use of N [26]

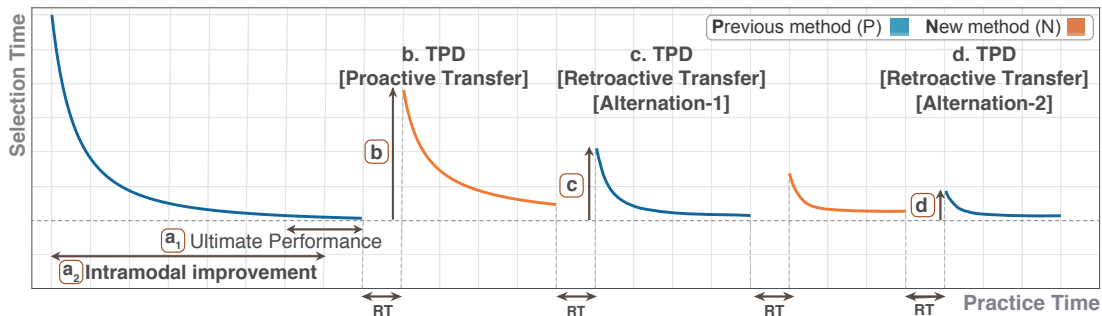


Figure 1. Conceptual model illustrating the alternation between a previous method (P) in blue and a new method (N) in orange color, as well as the corresponding phenomena: (a₁) Ultimate performance, (a₂) Intramodal improvement, (b) Proactive transfer, (c) Retroactive transfer (first alternation) and (d) Retroactive transfer (second alternation). Temporal performance drop (TPD) indicates the temporal difference between the end of one method and the beginning of another method. Rest time (RT) shows the interval time between learning of two methods.

Theories of retroactive interference include unlearning and response competition (or response-set suppression). *Unlearning* is the complete forgetting of a permanent memory (similar to deskilling). *Response-set suppression* is when users learn to suppress all of the previously-learned associations to correctly respond to the newly-learned associations. Retroactive interference is thus carryover effects from users suppressing the original interface [6]. Altmann et al. [3] proposed a theory that decay and interference are functionally-related. By their account, when an attribute is updated in memory, the old value decays to prevent interference with new values. Moreover, the decay rate adapts to the rate of updates made to memory.

In HCI, retroactive transfer has received only little attention. In the comprehensive book *Human Computer Interaction: Fundamentals* [68] retroactive interference is only briefly mentioned as a false memory phenomenon [63]. One exception is the study of Walker & Olson (1988) who investigated proactive and retroactive transfer of command shortcuts in two text editors [73]. While they did not observe proactive transfer, they observed a retroactive transfer effect between the two sets of shortcuts (EMACS and EXPRESS). EXPRESS appeared more robust to interference than EMACS. However, in this study, participants did not perform a visual-motor learning task involved with an interactive system as they did *not* execute the memorized shortcuts (they wrote them on paper).

We identified factors that can affect retroactive transfer.

Practice. One major factor influencing retroactive transfer is the amount of practice of the previous and new method [47, 2, 12]. Although increasing practice time of the new method seems to increase interference, increasing the practice of the previous method seems to be context sensitive. While some studies found that increasing the learning of the previous method reduces the interference [47, 60], others found the opposite [1]. Notably, the practice does not necessarily need to be physical (i.e. repeating a physical movement), but can be also mental. For instance, Wohldmann shows that mental practice can reduce the retroactive interference [78]. Moreover, the distribution of the amount of practice can also have an impact of retroactive transfer [36].

Rest Time. The time interval between the learning of two methods can also influence retroactive transfer [14, 66]. More precisely, when the retention test was immediate (short rest

time), they attribute retroactive interference to recency effects, but when the retention test is delayed (long rest time), the original response is more likely. When time is delayed overnight, it further fosters consolidation [27] and can also play a role in the magnitude of the transfer [36, 44].

Task. The nature of the task, recognition-based vs. recall-based also influences the retroactive transfer [6]: In cognitive memory tests of proactive and retroactive interference, participants experience several memory deficits in recall tests, but retroactive interference is almost completely eliminated with the recognition tests [6].

Similarity. The similarity factor is more relevant to HCI as designers can more easily manipulate it. Several optimization methods have been proposed to maximize both the performance of the new layout and its similarity with the previous layout [80, 10, 25, 57, 7, 11]. Similarity is defined as the number of identical elements shared in the two methods [26].

Only few studies focused on controlling the similarity factor for retroactive interference [36]. Osgood [58] showed that the amount of retroactive transfer is affected by similarity during a paired-associate task learning. Paired-associate learning involves the pairing of two items: a stimulus and a response. According to the Osgood's theory of learning transfer, retroactive interference increases when the stimuli in both layouts are constant but their responses differ.

RETROACTIVE TRANSFER IN REAL-WORLD

We run an interview-based study to understand the retroactive transfer phenomena when using interactive systems.

Interview 1: General Experiences of Retroactive Transfer Methodology

In this interview, we adopted a *semi-structured* methodology, with several memory aids. We conducted the study in participants' working environment to aid the recall of relevant situations. We explained to participants the concept of retroactive transfer by using the example of a person alternating between right-side and left-side driving when changing countries. We emphasized the fact that the "transfer" can be positive, negative or neutral to not bias participants. We showed them a printout of popular interface examples' images, including operating systems, graphical menus, mouse/touch-pad devices, keyboard devices, keyboard and gestures shortcuts,

programming languages, app release versions, the same app on different platforms (e.g. Office in MacOS and Windows). The interview was audio-recorded, lasted about 1 hour and then analyzed using annotations and timestamps on Camptasia. The time-stamped notes were further analyzed in spreadsheet software to help index episodes of interest. *Participant Profiles.* We interviewed 6 adults (< 40 years), 4 male, 2 female, 4 French, one Australian and one Taiwanese. Two were professional programmers in different application areas: embedded systems and haptic devices. The other 4 were researchers in different domains: medical, bio-mechanics, information theory, and sociology.

Data Collection & Analysis. Participants reported a total of 71 incidents of alternation between previously (P) and new (N) interfaces ($P \rightarrow N \rightarrow P$) which were either negative (33) or positive (18) retroactive transfer. Regarding the 20 remaining incidents, participants could not recall whether there was a change in their initial performance. We summarize our observations using the notation (x_i, y_p), when one observation appears in *more than one* incidents. x indicates the number of incidents and the y to the number of participants. For example, if 2 participants mentioned a total of 5 incidents when typing text, the notation is ($5_i, 2_p$).

Findings

Why do Users Alternate between Interfaces? Some alternations are due to external factors (30_i, 6_p), such as a programmer who has to code for clients who use different systems. Other alternations occur because of the need to use complementary functionalities between two interfaces (41_i, 6_p), such as the mobile version of a desktop application. In both cases, the alternation appeared as a necessity rather than a choice.

Positive Retroactive Transfer. All participants reported incidents of positive retroactive transfer (18_i, 6_p). A common pattern was when interface N highlights an unknown functionality of interface P (13_i, 6_p), e.g., learning the new language C helped to learn Python (familiar language) in more depth.

Negative Retroactive Transfer. All participants reported incidents involving retroactive interference (33_i, 6_p). Most of them involved using a keyboard (16_i, 6_p), for example, when alternating between QWERTY and AZERTY layouts. They reported a drop in their typing speed on their primary keyboard after experiencing the new layout. They also recalled being frustrated with typing errors, especially when executing an incorrect keyboard shortcut, leading sometimes to a "disaster", e.g., executing Ctrl+Q ("quit") instead of Ctrl+A ("select all"). Some participants also reported a retroactive interference while searching for a functionality in a graphical menu (4_i, 4_p), for instance, when alternating between different versions of the same application. Others reported making more errors with their previously learned programming language after learning a new one (5_i, 3_p). An expert gamer reported a drop in his gaming performance while alternating his fast mouse at home with his regular mouse device at work.

Connection to Expertise. An observation consistent across all these incidents is that perceiving a retroactive interference required a conscious prior effort to achieve expertise. For example, participants experienced retroactive interference while

trying to be fast typists, or effective programmers. In contrast, casual activities involving alternations (e.g. browsing websites on the internet) were not "demanding" enough to allow them perceive similar interference.

Factors Identified by Participants. Several participants spontaneously elaborated on reasons that can increase/reduce the previously mentioned interference. The most pronounced comment that all participants reported was the degree of *similarity* between interfaces as the main root of interference (6_p). For example, to overcome retroactive interference, some often try to consciously emphasize the differentiating factors between the two interfaces (2_p), e.g., trying to think only in French while using AZERTY, and in English while using QWERTY. One participant mentioned that he differentiates the way he interacts with each interface to avoid confusion; using mostly keyboard shortcuts with Windows at work and rather "point and click" style of interaction with his MacOS at home. Others argued that the *amount of practice* with the new interface is the factor that will determine the amount of interference later on (4_p). For example, one programmer reported to try intentionally reduce his exposure to the new interface: he is avoiding to write code on MacOS, doing only final compatibility check of his code, so as not to be confused when returning back to his primary system (Linux). In more extreme cases, participants reduced the amount of practice to zero, sacrificing functionalities for consistency.

Conclusion. Several factors involving retroactive transfer were spontaneously mentioned by our participants such as *similarity* between two interfaces (6_p), and *amount of practice* of P and N (3_p). Therefore participants' elaborated observations appear consistent with the literature described in the previous section. We also learn that using a keyboard was a main source of retroactive interference. We thus decided to conduct another round of interviews with two additional participants focusing on keyboard layouts.

Interview 2: Post-study Focusing on Keyboard Usage

The second interview focuses on the keyboard usage. We interviewed two HCI researchers focusing on their episodic experiences from 3 weeks to 3 years in their life, when they experienced retroactive interference while using a keyboard.

Text Input. The first participant was regularly alternating between his AZERTY laptop (personal and professional usage) and a QWERTY keyboard dedicated to a professional platform for about two years. During this episode, he was looking at the keyboard using several fingers. He reported doing frequent errors especially when he knew that the consequences of an error were not important. After some time, he was able to anticipate potential sequences of characters that could lead to typing errors and voluntarily slow down his typing speed to avoid them. The second participant was using an AZERTY laptop at home, but an external QWERTY keyboard connected to this laptop at work during 3 weeks. He was able to enter text without looking at the keyboard. He reported doing errors and additional visual search for non frequent special characters. However, he reported that switching keyboards was like switching modes: "I was doing one error but not two. I need to recall which mode I was using". The second participant

was also using the SWIFT keyboard on his smartphone, but for some unidentified reason the system sometimes display the default IOS keyboard. The two layouts are very similar, except that ‘.’ is closer to the return key on the IOS keyboard. He reported doing much more errors than the previous experience and did not have the impression of "switching mode". When he was asked why, he attributed this difference to the "degree of similarity" between the two interfaces.

Keyboard Shortcuts. The first participant regularly uses Inkscape on MacOS which is the only software using Ctrl instead of Cmd as main modifier. He reported doing “one, max two errors” and then being comfortable using these keyboard shortcuts. This echoes the impression of "switching mode" of the first participant when alternating keyboard layouts. He also reported that the context is so different that coming back to "normal" software is fast.

DISCUSSION AND DESIGN RATIONALE

Our review of the psychology literature highlighted several factors that contribute to retroactive transfer, from which **similarity** and **practice** were the more commonly reported in our interview study. The impact of similarity on retroactive interference was reported as task- and context-dependent, especially in the second part of the interviews. Similarity (or familiarity) has also been considered as an important criterion in the optimization of user interfaces [34, 40]. Participants also reported performing repeated **alternations** between user interfaces. Such alternations are not typically studied in experimental psychology but relevant in HCI because of the ubiquitous nature of interaction. Though we exposed participants to several UI scenarios (mouses, menus, browsers, etc.), the reported incidents mainly focused on **text entry** interfaces.

In the following section, we build on these findings to design a controlled experiment. We aim to understand whether retroactive transfer occurs when interacting with user interfaces. In particular, we investigate the impact of similarity and number of alternations on user performance. We initially contemplated to use a regular text entry task to study retroactive transfer phenomena. However, mastering a single keyboard layout requires a lot of time, i.e. several days, and studying retroactive transfer with two keyboard layouts and alternations would require at least 3 times that amount, which was not a feasible option for a lab experiment (~1h30). We thus present an abstract task (Figure 2) that operationalizes the key aspects of alternating between two keyboard layouts, alleviating possible bias related to their previous experience and accelerating users’ expertise.

EXPERIMENT

The primary goal of this experiment was to understand the retroactive interference phenomena and how learning a **New** layout (N) interferes with the retention of a **Previously learned** layout (P), as described in the sequence below:

$$P \rightarrow N \rightarrow \boxed{P} \quad (1)$$

First, we investigated the impact of the *layout* on retroactive transfer. Specifically, we manipulated the degree of *similarity* between N and P by modifying the location of certain elements in the layout. We controlled both the *number of changes* between the two layouts as well as the *spatial proximity* of these changes (i.e. how far an element has been moved).

Second, we investigated the impact of *alternation* on retroactive transfer. So we tested two alternations rather than the typical one alternation in a the learning sequence:

$$P \rightarrow N \rightarrow P \rightarrow N \rightarrow \boxed{P} \quad (2)$$

Where participants alternate two times between the layouts P and N before measuring the performance with the layout P .

All experimental materials are available on [OSF](#).

Hypotheses

We propose three hypotheses based on concepts presented in the literature review and the insights from the interview study:

- **H1** Learning a new layout N results in retroactive interference with the previously learned layout P .
- **H2** The *similarity* between the layouts P and N influences the magnitude of retroactive interference, where similarity is defined in term of number of changes and the nature (proximity) of these changes.
- **H3** Increasing the number of *alternations* between the layouts P and N decreases retroactive interference.

Participants

58 university engineers and students aged 18-40 participated to the experiment and were divided into three groups, one group per condition (see below). 4 participants were removed due to under-performance during the training phase to fairly ensure similar initial performance across conditions. It results that each condition was tested with 18 participants. The exclusion rule was decided before running the statistical analysis to prevent p-hacking [35, 76]. Participants received 15 euros for their participation. A bonus of 10 euros was awarded to the 3 fastest participants of each condition to motivate them to quickly reach a high level of performance.

Apparatus

Interface. Figure 2 illustrates the interface, a virtual grid layout of 6×3 common symbols (e.g. hat, rabbit, etc.) designed according to the following criteria: First, the interface captures the key phenomena of text input. The grid in Figure 2 represents an abstraction of a (virtual) keyboard layout. Selecting an element in the grid involves visual search, pointing, chunking, learning and motor control. The two-sided layout further allows for multi-finger and two-handed interaction. Consequently, the abstracted grid acts as an informative proxy for a wider class of interfaces such as numpads or grid menus.

Second, the interface fosters rapid skill acquisition within the time constraints of the experiment. While the number of rows (3) was similar to keyboard layouts, we used 6 instead of 10 columns to enable participants to reach a performance plateau. The advantage of the simplified grid over a real keyboard is that it can reduce visual search time and simplify finger-key assignation, both contributing to touch typing skills.

Third, we wanted to prevent confounding factors from prior user experience. Therefore, we replaced letters with symbols to ensure that the chosen interface would not remind users of existing keyboard layouts to avoid unintended skill transfer.

Finally, we wanted to isolate mapping-related errors from retroactive interference. Thus we increased slightly the size of



Figure 2. Interface displayed on the tablet. Four target symbols are displayed on the top. The green check marks confirm correct selections. Participants select the symbols from the grid at the bottom. Digits were not present in the interface. They serve to identify a symbol in this paper.

the buttons from 0.6 to 0.9 inches to reduce irrelevant pointing errors due to the fat finger problem [70].

In summary, we developed an abstracted layout to study retroactive transfer within an interactive system. If we were to use real keyboards, we would probably face differences in skill level with the chosen keyboards among participants. The nature of the keyboard layouts and the frequency of the letters would also introduce additional factors difficult to disentangle. It is likely that the targeted phenomenon will be more difficult to observe and/or to explain. This interface maintained the key aspect of text input while remaining flexible enough to control the design factors described below.

Device and setting. We used a Galaxy Note multi-touch tablet on Android 5.1. The display was 10.1 inches with a resolution of 2560x1600 pixels. The tablet was lying on a table with the landscape orientation. Participants sat on a chair and were free to move the tablet on the table to have a high level of comfort.

Experimental Design

Task and Stimulus

A sequence of four target symbols was displayed at the top of the interface (Figure 2). Participants selected the corresponding symbols by touching the grid at the bottom. When the target symbol was *correctly* selected, a green check mark confirmed the selection and participants proceeded to the next target in the sequence. When an *error* was made, a beep informed participants to re-select the symbol. Once the sequence was successfully completed, a novel sequence was displayed.

Procedure and Phases

Participants filled out a consent form and turned off all personal devices. The experimenter explained the task and asked participants to complete it as quickly as possible. The experimenter encouraged the participants to use both hands and several fingers to maximize their performance.

Table 1a shows the 5 phases (1 2 3 4 5) of the experiment. Each phase lasted about 10 min. Between phases there was minimum 1 minute break. During the phases 1, 3 and 5, all participants selected symbols using the layout *P* (the selection task was detailed in the section *Task*). During the phases 2 and 4, the participants of the condition control took a break (10+1 min), while the participants of the two other conditions selected symbols using the layout *N*. The configuration of

a	Phase	1 → 2 → 3 → 4 → 5
b	Alternation Sequence	P → N → P → N → P
c	Condition control	P → ∅ → P → ∅ → P
d	Condition 4-change	P → N ₄ → P → N ₄ → P
e	Condition 8-change	P → N ₈ → P → N ₈ → P
f	Practice Time (min)	10 → 10 → 10 → 10 → 10

Table 1. Summary of phases, alternations, conditions and practice time for each condition. Blue colors indicate the layout *P* and orange colors the layout *N*. (a) Phase: an index of exposure to the layouts. (b) Alternation Sequence: the order of appearance of the layouts. (c, d, e) the conditions determined the number of changes between layouts *P* and *N*. (f) Practice time for each phase.

the layout *P* is illustrated in Figure 2. The layout *N* differs depending on the condition (condition 4-change or condition 8-change). We next describe the design of the layout *N*.

Layout *N* and Similarity

The layout *N* was used in the phases 2 and 4. To study the impact of similarity on retroactive transfer, the layout *N* differed from the layout *P* in two components of similarity:

Number of changes. Our primary measure of similarity and dissimilarity was the number of paired symbols swapped between the layouts. More precisely, the layouts *N* and *P* were identical in geometry (3×6 cells) and in list of symbols. The dissimilarity between *N* and *P* was determined by the minimum number of (swap) operations needed to transform the layout *N* into the layout *P*. Thus the layouts *N* and *P* differed only on the location of swapped symbols.

Proximity. The number of changes alone does not capture the *nature* of the changes. We thus introduced a second similarity measure termed *proximity*. Proximity indicated the number of cells that existed between the previous and the new location of a symbol. Consider a keyboard layout; swapping either two adjacent keys or two keys far from each other might affect user strategies and the risk of errors. We considered two proximity conditions:

- *Close.* The two locations are adjacent.
- *Far.* There are at least 5 cells between the two locations. Consequently, the previous and the new location of the symbol are in different sides of the grid.

Configuring the layout *N*. In accordance with our similarity measures, we designed two layouts *N*: *N*₄ and *N*₈. In condition 4-change, participants used the layout *N*₄ as interference layout, and in condition 8-change, the layout *N*₈.

Specifically, the layout *N*₄ was identical to *P* except 2 pairs of symbols which were swapped. Swapping means 4 symbols changed location and 14 symbols remained at the same location in the grid. Among the two swaps, one was close and one was far to control the proximity. The layout *N*₈ was identical to *N*₄ except 2 (extra) pairs of symbols which were swapped. Among the 2 swaps, one was close and one was far. Comparing *N*₈ to *P*, 4 pairs of symbols were swapped (i.e. 8 symbols changed location) containing 2 close and 2 far swaps.

In summary, the *N*₄ was more similar to *P* than the *N*₈, because the *N*₄ had fewer number of changes (4 vs. 8 symbols). Thus









				Location		
				Previous	New	
	Prox.	No.	Icon	control	4-change	8-change
Static	Fix	1		(1,3)	(1,3)	(1,3)
		2		(3,5)	(3,5)	(3,5)
		3		(2,1)	(2,1)	(2,1)
		4		(1,6)	(1,6)	(1,6)
Dynamic	Close	1		(1,4)	(1,5)	(1,5)
		2		(3,2)	(3,2)	(2,2)
	Far	3		(2,6)	(3,1)	(3,1)
		4		(1,1)	(1,1)	(2,5)



Table 2. Target symbols' locations (as shown in Figure 2) for the three layouts. These targets are divided into two types: static and dynamic.

comparing the performance of the 4-change condition (using N_4) and 8-change condition (using N_8), we tested the impact of similarity on retroactive interference.

Target Symbols: Static and Dynamic

From the 18 (6×3) symbols of each layout, participants had to select 8 of them. Table 2 shows these 8 target symbols.

We distinguish between *static* and *dynamic* targets. Static targets had identical locations in all layouts (P , N_4 and N_8). Dynamic targets had changed locations between the layout P and the layouts N_4 and/or N_8 . The reason for asking participants to also select static targets was to understand whether changing the location of the learned symbols (dynamic) can influence the performance of the unchanged symbols (static).

To be able to compare the performance of static and dynamic targets, the locations of the static symbols vertically mirrored the locations of dynamic symbols (i.e.,  mirrored  in Figure 2). To measure the effect of proximity, half of the dynamic targets were involved in a close change and half in a far change.

Design

The experiment had a mixed design. We compared learning in three conditions. The participants of condition control (Table 1c) learned only the layout P performing phases (1, 3, 5). The participants of condition 4-change (Table 1d) learned both layouts P and N_4 performing all phases (1,2,3,4,5). The participants of condition 8-change (Table 1e) learned both layouts P and N_8 performing also all phases. Each phase contained 9 blocks. Each block contained 16 trials. In each trial, participants selected 4 target symbols (a total of 64 selections per block). The 8 target symbols appeared randomly within the block following a uniform distribution similar to [4].

In summary, the design was: 3 conditions \times 18 participants \times 5 phases (only 3 phases for the condition 1) \times 9 blocks \times 16 trials \times 4 targets = 134 784 selections.

Data Analysis

We considered five independent variables. The between factor was:

- CONDITION (control, 4-change and 8-change) indicating the number of changes.

The within factors were:

- TYPE (static and dynamic)
- PROXIMITY (close and far)
- ALTERNATION (1, 2) Alternation-1 occurred in phase ③ and alternation-2 in phase ⑤.
- BLOCK (1-9)

Our analysis focused on four phenomena, each of which had its own dependent variables:

Intramodal Improvement (Figure 1a): To investigate the effect of practice on learning process of P in phase ①, we measured the average *selection time* per block.

Proactive transfer (Figure 1b): To investigate the effect of learning a previously learned layout P on the performance of the layout N , we measured the *temporal performance drop* between phase ① and phase ②. It is defined as the difference of average selection time between the end of phase ① (last block²) and the beginning of phase ② (first block) [71, 66].

Retroactive transfer, Alternation-1 (Figure 1c): To investigate the effect of learning a new layout (first exposure to layout N) on the performance of the layout P , we measured the *temporal performance drop* between phases ① and ③ [79].

Retroactive transfer, Alternation-2 (Figure 1d): To investigate the effect of alternation (second exposure to layout N) on the performance of layout P , we compared the *temporal performance drop* between phases ① and ③ with the *temporal performance drop* between phases ③ and ⑤.

We also measured *error* per block, i.e. the number of incorrect attempts to select a target for these phenomena.

Statistics. To be consistent with previous retroactive transfer studies, we analyzed our data using TWO-WAY MIXED ANOVA [36]. To better communicate our findings [24], we conducted a second analysis of the same data using 95% confidence intervals (CI), which we report visually in Figure 3. The results with both methods were consistent.

RESULTS

Figure 3 compares the learning curves for the conditions control, 4-change and 8-change. Blue colors indicate the analysis of the layout P and orange colors the analysis of the layout N . We report both analyses (TWO-WAY MIXED ANOVA and CI) on intramodal improvement (a), proactive transfer (b), and retroactive transfer for alternation-1 (c) and alternation-2 (d). We only report selection time and *mean* temporal performance drop among participants (TPD) along with CI, as we did not find effects of the different factors on ERROR.

Intramodal Improvement

To measure intramodal improvement, we analyzed all blocks of phase ① for all conditions. A CONDITION (control, 4-change, 8-change) \times BLOCK (1-9) ANOVA was performed on TIME with a repeated measures analysis of the last factor. It yielded a significant effect of BLOCK, $F_{8,408} = 83.68, p <$

²We also considered the best block (instead of the last block) for proactive and retroactive transfer. We considered the best block as some participants could experience fatigue at the end of each block. However, the results were consistent for both analysis (last block vs. best block). We thus decided to only report data for the last block.

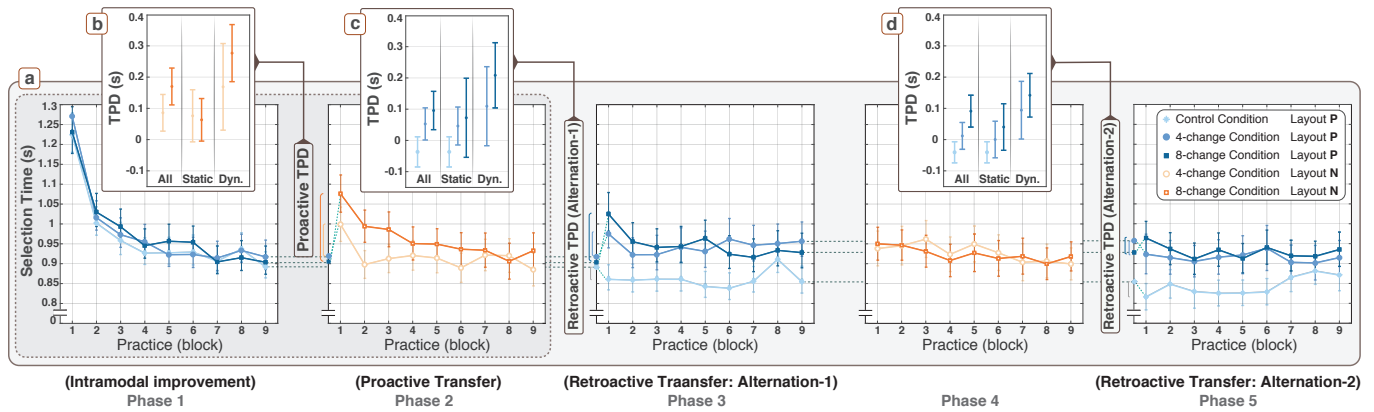


Figure 3. (a) Intramodal improvement (time) of layout P for all phases per BLOCK \times CONDITION for all targets. (b) Proactive Transfer. (c) Retroactive Transfer in alternation-1. (d) Retroactive Transfer in alternation-2. (mean TPD per CONDITION (number of changes) \times TYPE OF TARGETS (static vs. non-static) was calculated for b,c,d). Error bars indicate 95% CI.

0.0001. Post-hoc comparisons with Tukey HSD test indicated that participants saved 300ms (25% improvement) from block 1 to block 4 and no significant differences from block 4 to 9. It indicates that participants reached a plateau of performance with the layout P. We did not observe an effect of CONDITION. As all participants used the same layout P, it confirms similar initial performance across conditions. The CI analysis, illustrated in Figure 3a, also shows no observable difference in performance among the participants of the different conditions.

Proactive Transfer

We first compared the two conditions 4-change and 8-change considering *all targets*. ANOVA revealed a significant effect of CONDITION on TPD, $F_{1,34} = 4.48, p < 0.05$. A post-hoc Tukey HSD test shows that TPD with N_4 ($+86 \pm 59$ ms) is significantly smaller than the one with N_8 ($+170 \pm 59$ ms), suggesting that the TPD increases with the number of changes. The CI analysis, illustrated in Figure 3b (All targets), confirms a difference between the two conditions and that both conditions are affected by a performance drop (0 is not included in the CI).

We then refined our analysis by distinguishing *static vs. dynamic* targets (see the list of static and dynamic targets in both layouts N_4 and N_8 in Table 2). We conducted a CONDITION (4-change, 8-change) \times TYPE (static, dynamic) ANOVA with a repeated measures analysis of the last factor. It yielded a significant effect of TYPE on TPD ($F_{1,34} = 7.86, p < 0.01$). A post-hoc Tukey HSD test shows the TPD caused by static targets ($+70 \pm 53$ ms) is significantly smaller than the one caused by dynamic targets ($+240 \pm 76$ ms). However, we did not have an effect of CONDITION on TPD ($p = 0.3$) or CONDITION \times TYPE interaction effect ($p = 0.2$). It suggests that the larger performance drop of N_8 is due to the extra number of changes. However this extra number of changes does not affect the TPD of the other static and dynamic targets.

Finally, we refined again our analysis to study the impact of PROXIMITY on the TPD of dynamic targets. We conducted a CONDITION (4-change, 8-change) \times PROXIMITY (close, far) ANOVA with a repeated measures analysis of the last factor. ANOVA revealed no effect of CONDITION ($p = 0.1$), PROXIMITY ($p = 0.7$) or their interaction ($p = 0.7$) on TPD.

Retroactive Transfer: Alternation-1

We first compared the three conditions (control, 4-change and 8-change) and considered all targets. ANOVA yielded a significant effect of CONDITION on TPD, $F_{2,51} = 6.8, p < 0.005$. Post-hoc comparisons using Tukey HSD test indicated that condition control (-37 ± 48 ms) has a significant smaller TPD than condition 4-change ($+52 \pm 51$ ms) and condition 8-change ($+95 \pm 61$ ms). But there was no significant difference between the conditions 4-change and 8-change. The CI analysis, illustrated in Figure 3c (All targets), confirms a difference between the conditions and that conditions 4-change and 8-change are affected by a performance drop (0 is not included in the CI).

We next refined our analysis by investigating static and dynamic targets. As condition control does not have dynamic targets, we first performed an analysis focusing only on static targets for all conditions. We then excluded the condition control and compared static vs. dynamic targets by considering only conditions 4-change and 8-change. (similar to the analysis of proactive transfer).

Static targets. We run a CONDITION (control, 4-change, 8-change) ANOVA which showed no significant effect of CONDITION on TPD ($p = 0.3$). The average TPD for static target is -37 ± 48 ms in condition control, $+45 \pm 60$ ms in condition 4-change and $+72 \pm 126$ ms in condition 8-change.

Static vs. Dynamic. We conducted a CONDITION (4-change, 8-change) \times TYPE (static, dynamic) ANOVA with a repeated measures analysis of the last factor. It showed a significant effect of TYPE on TPD ($F_{1,34} = 5.2, p < 0.05$). A post-hoc Tukey HSD test shows the TPD caused by static targets ($+59 \pm 69$ ms) is significantly smaller than the one caused by dynamic targets ($+174 \pm 81$ ms). However, we did not have an effect of CONDITION on TPD ($p = 0.2$) or CONDITION \times TYPE interaction effect ($p = 0.6$). It suggests that the larger performance drop of N_8 is due to the extra number of changes. However this extra number of changes does not affect the TPD of the other static and dynamic targets. Finally, we studied the impact of PROXIMITY on TPD of dynamic targets. We run a CONDITION (4-change, 8-change) \times PROXIMITY (close, far) ANOVA. It showed no effect of CONDITION ($p = 0.1$), PROXIMITY ($p = 0.2$) or their interaction ($p = 0.5$) on TPD.

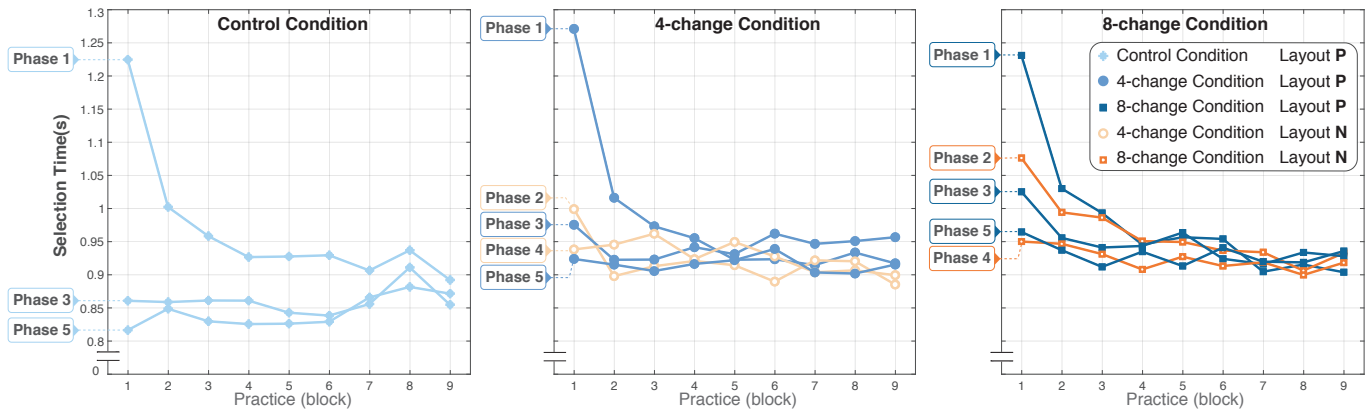


Figure 4. Intramodal improvement (selection time) for each phase for each condition.

Second Retroactive Transfer: Alternation-2

We considered all targets and performed a $\text{CONDITION (control, 4-change, 8-change)} \times \text{ALTERNATION (1,2)}$ ANOVA on TPD with a repeated measures analysis of the last factor. It yielded a significant effect of CONDITION , $F_{2,51} = 11.56, p < 0.0001$ and ALTERNATION , $F_{1,51} = 6.35, p < 0.05$ on TPD. Post-hoc comparisons using Tukey HSD test revealed that condition 8-change ($+91 \pm 51\text{ms}$) has a significant larger TPD than conditions control ($-41 \pm 34\text{ms}$) and 4-change ($+12 \pm 43\text{ms}$), confirmed by the CI analysis in Figure 3d (All targets). Moreover, Post-hoc comparisons using Tukey HSD test shows that TPD during the alternation-1 ($+52 \pm 36\text{ms}$) was significantly higher than the one during alternation-2 ($-11 \pm 34\text{ms}$).

Similar to the previous subsection, we refined our analysis comparing static targets for all conditions and comparing static vs. dynamic targets for conditions 4-change and 8-change.

Static targets. We conducted a $\text{CONDITION (control, 4-change, 8-change)} \times \text{ALTERNATION (1,2)}$ ANOVA which showed no significant effect of CONDITION ($p = 0.09$) or ALTERNATION ($p = 0.1$) on TPD. The average TPD for static target is $-41 \pm 33\text{ms}$ in condition control, $0 \pm 58\text{ms}$ in condition 4-change and $+40 \pm 74\text{ms}$ for condition 8-change.

Static vs. Dynamic. We conducted a $\text{CONDITION (4-change, 8-change)} \times \text{TYPE (static, dynamic)} \times \text{ALTERNATION (1,2)}$ ANOVA with a repeated measures analysis of the last factor. It revealed a significant effect of TYPE ($F_{1,34} = 10.16, p < 0.005$), as well as a significant effect of ALTERNATION on TPD ($F_{1,34} = 4.97, p < 0.05$) on TPD. A post-hoc Tukey HSD test shows the TPD caused by static targets ($+20 \pm 47\text{ms}$) is significantly smaller than the one caused by dynamic targets ($+126 \pm 55\text{ms}$). It also shows TPD for alternation-1 ($+108 \pm 52\text{ms}$) is significantly higher than alternation-2 ($+22 \pm 49\text{ms}$).

In another ANOVA we investigated the effect of PROXIMITY on the TPD of dynamic targets. There was no significant effect of CONDITION ($p = 0.3$), PROXIMITY ($p = 0.07$), ALTERNATION ($p = 0.1$), or of their interactions, on TPD.

Another Perspective on Data

Figure 3 illustrates the comparison of conditions but Figure 4 illustrates the comparison of phases offering another perspective on the same dataset. Figure 4 compares the performance

improvement for each phase given a condition. It illustrates that at the end of each phase, the selection time is similar regardless the phase indicating that users reached a plateau of performance. It also shows the initial selection time (block 0) decreases with the number of alternations but increases with the number of changes.

Self Estimation of Temporal Performance Drop

At the end of the experiment, participants of conditions 4-change and 8-change rated the evolution of their performance for proactive and retroactive transfer (7-Likert scale). Participants reported a TPD for both proactive and retroactive transfer, which increased with the number of changes in the new layouts (Figure 5). Therefore, participants subjective performance appeared consistent with the objective metrics. However, especially for 4-change, we note that about 33.34% participants reported the impression to improve their performance when returned to the layout P .

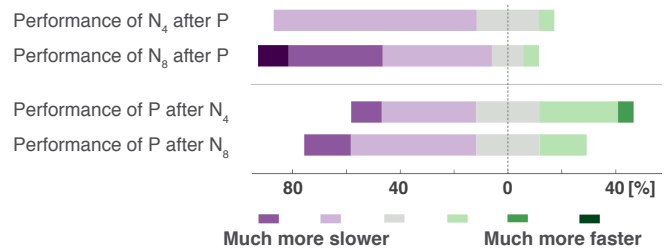


Figure 5. Participants' self estimation of their temporal performance drop (TPD) per TRANSFER (proactive - Top and retroactive - Bottom) and CONDITION (4-change vs. 8-change).

DISCUSSION

We now summarize our findings, present directions for future work and discuss implications for design.

Findings

H1: Learning a new layout N results in retroactive interference with the previously learned layout P

The primary objective of this paper was to investigate the effect of learning a new layout N on the performance of previously learned layout P . The results show that retroactive interference occurs when participants are temporally exposed to a partially changed layout (i.e. N_4 or N_8) and confirm **H1**. The performance drop is $+52\text{ms}$ for condition 4-change and

+95ms for condition 8-change which represents 16% and 37%, respectively. We were expecting some interference, but these results remain surprising, especially for N_4 . Indeed, given our experimental design, 6 out of 8 targets (75%) are at the same location in P and N_4 . So, for 75% of the targets, users should benefit practicing N_4 as it should be equivalent to continue practicing P . However, having just the two targets with different positions in N_4 were sufficient to produce a significant performance drop for the *entire* interface. In other words, even a few changes in the interference interface results in retroactive interference which significantly impairs the overall performance with the previously learned interface.

H2: The similarity between the layouts P and N influences the magnitude of retroactive interference

The ANOVA did not indicate a difference in temporal performance drop (TPD) between conditions 4-change and 8-change at either the layout level (condition) or the target level (static vs. dynamic). A key feature of our experimental design was to control two aspects of similarity: number of changes and proximity. However, neither revealed an effect on TPD. Our results thus failed to confirm a significant effect for **H2**. Indeed, CI-based analysis (Figure 3c) at both the layout (-all targets) and target (-static, -dynamic) levels suggest that TPD increases linearly with the dissimilarity between P and N (i.e. N_8 is less similar than N_4 to P). One possible explanation is that the effect exists but somehow was not captured by our experimental design or analysis. Further investigation should consider larger sample sizes, conditions with a larger number of changes and potentially more training with both P and the interference layout.

H3: Increasing the number of alternations decreases retroactive interference

Our study aimed to evaluate users' behaviour when alternating between interfaces. Our results confirm **H3** and show that the TPD significantly decreases (31%) with the number of alternations. Figure 3a illustrates this clearly with a continual decrease in the the initial selection time for each phase. It suggests that retroactive interference can be reduced with alternations and practice.

Error. Contrary to expectations, we did not find a significant effect on ERROR for the different factors. One possible explanation is that our task was cognitively demanding. In more realistic scenarios, users might focus on higher level tasks (e.g. writing an email) decreasing their attention to the keyboard. We foresee that the TPD would then translate into more errors in more ecological settings, but more investigation is necessary.

Proactive vs. retroactive interference. Most results about retroactive interference are in line with proactive interference such as the effect of TYPE (static vs. dynamic) on TPD, the lack of effect of proximity and also experiencing a TPD with both conditions 4-change and 8-change by users. It confirms that switching from one interface to another one ($P \rightarrow N$ or $N \rightarrow P$) produces interference. However, for proactive interference, the condition 4-change appeared significantly less affected by the TPD than the condition 8-change, probably because its larger effect size.

Future work

Our experiment focused on similarity as it is a critical factor for interaction design [10]. However, there were several factors we left aside that would require further investigation. For instance, manipulating the amount of practice and rest time might indicate whether expert users (i.e. skilled touch typists) would experience these phenomena to the same degree. Also, one might investigate how the distribution of target frequency influences proactive and retroactive transfer. In our experimental design, we used a uniform distribution (i.e. each target had the same probability of appearing within a block). So, it was unlikely that users learned sequences of characters [42]. However, in real scenarios, the frequency of each character and bigrams depend on the language. According to results in sequential learning [42], the magnitude of the interference could be larger for frequent bigrams. In addition, one should investigate the impact of different modalities, e.g. gestures, on retroactive interference. Which modalities minimize retroactive interference? Finally, future research is necessary to generalize our results to real-world problems, e.g., real keyboard layouts.

Implications

We encourage designers to pay more attention to the impact of introducing novel interfaces on previously learned skills because of the ubiquitous nature of interaction across applications. Our interviews revealed that there are many situations in users' daily routine where they alternate between devices, interfaces, software, or operating systems for the same type of tasks (e.g., pointing, entering text, executing commands, etc.).

Designers and researchers should evaluate more systematically their interfaces for retroactive transfer. When designing or testing new interfaces, it is common to consider intramodal improvement and proactive transfer. However, considering retroactive transfer seems essential for ecologically valid investigation of interface design.

Similarly, future models of performance should integrate phenomena related to proactive and retroactive transfer. Recently, Jokinen et al. [39] presented a model to explain the negative impact of switching to a partially changed keyboard layout on typing performance. Our work provides two main directions to extend this model. First, our results show that the dynamic targets influence performance time of other targets when switching to the new layout (proactive transfer). Such models should thus introduce mechanisms to explain how changed keys influence the performance of other keys. Second, the model could be extended to integrate phenomena related to retroactive transfer.

Intelligent systems with the capacity to capture alternations can then use these models of performance to avoid/reduce retroactive interference. For instance, they can provide some recommendations such as increasing rest time at a given alternation or increasing training to reduce the risk of interference.

ACKNOWLEDGEMENT

This work was supported by the Agence Nationale de la Recherche (ANR-16-CE33-0023 GESTURE).

REFERENCES

- [1] Jack A. Adams. 1987. Historical Review and Appraisal of Research on the Learning, Retention, and Transfer of Human Motor Skills. 101 (01 1987), 41–74.
- [2] Richard A. Schmidt and Jacques Nacson. 1971. The Activity-Set Hypothesis for Warm-Up Decrement. 90 (10 1971), 56–64.
- [3] Erik M. Altmann and Wayne D. Gray. 2002. Forgetting to Remember: The Functional Relationship of Decay and Interference. (2002).
- [4] Erik M. Altmann and Wayne D. Gray. 2008. An Integrated Model of Cognitive Control in Task Switching. *Psychological Review* 115, 3 (2008), 602–639.
- [5] John R Anderson. 1982. Acquisition of cognitive skill. *Psychological review* 89, 4 (1982), 369.
- [6] Michael C Anderson and James H Neely. 1996. Interference and inhibition in memory retrieval. In *Memory*. Elsevier, 237–313.
- [7] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. Menuoptimizer: Interactive optimization of menu systems. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 331–342.
- [8] Amy J Bastian. 2008. Understanding sensorimotor adaptation and learning for rehabilitation. *Current opinion in neurology* 21, 6 (2008), 628. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2954436/pdf/nihms242763.pdf>
- [9] François Bérard and Amélie Rochet-Capellan. 2015. The transfer of learning as hci similarity: Towards an objective assessment of the sensory-motor basis of naturalness. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1315–1324.
- [10] Xiaojun Bi, Barton A Smith, and Shumin Zhai. 2010. Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 283–286.
- [11] Xiaojun Bi and Shumin Zhai. 2016. IJqwerty: What difference does one key change make? Gesture typing keyboard optimization bounded by one key position change from Qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 49–58.
- [12] Tom Brashers-Krug, Reza Shadmehr, and Emanuel Todorov. 1995. Catastrophic Interference in Human Motor Learning. In *Advances in neural information processing systems*. pp. 19–26.
- [13] George E Briggs. 1954. Acquisition, extinction, and recovery functions in retroactive inhibition. *Journal of Experimental Psychology* 47, 5 (1954), 285.
- [14] ME Bunch. 1946. Retroactive inhibition or facilitation from interpolated learning as a function of time. *Journal of comparative psychology* 39, 5 (1946), 287.
- [15] Raymond R. Burke and Thomas K. Srull. 1988. Competitive Interference and Consumer Memory for Advertising. *Journal of Consumer Research* 15, 1 (1988), 55–68. DOI : <http://dx.doi.org/10.2307/2489172>
- [16] Pieter Buzing. 2003. Comparing Different Keyboard Layouts: Aspects of QWERTY. *DVORAK and Alphabetical Keyboards* (2003).
- [17] Moran T. P. Card, Stuart K and Allen Newell. 1983. *The psychology of human-computer interaction*. Lawrence Erlbaum Associates.
- [18] John M. Carroll (Ed.). 2003. *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [19] Klaus Christoffersen, Christopher N Hunter, and Kim J Vicente. 1996. A longitudinal study of the effects of ecological interface design on skill acquisition. *Human factors* 38, 3 (1996), 523–541.
- [20] Edward Clarkson, James Clawson, Kent Lyons, and Thad Starner. 2005. An Empirical Study of Typing Rates on mini-QWERTY Keyboards. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1288–1291. DOI : <http://dx.doi.org/10.1145/1056808.1056898>
- [21] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2, Article 31 (Nov. 2014), 36 pages. DOI : <http://dx.doi.org/10.1145/2659796>
- [22] Robert Coop, Aaron Mishtal, and Itamar Arel. 2013. Ensemble Learning in Fixed Expansion Layer Networks for Mitigating Catastrophic Forgetting. *IEEE Transactions on Neural Networks and Learning Systems* 24, 10 (2013), 1623—1634. DOI : <http://dx.doi.org/10.1109/TNNLS.2013.2264952>
- [23] Mary Czerwinski, Eric Horvitz, and Susan Wilhite. 2004. A Diary Study of Task Switching and Interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 175–182. DOI : <http://dx.doi.org/10.1145/985692.985715>
- [24] Pierre Dragicevic. 2016. Fair statistical communication in HCI. In *Modern Statistical Methods for HCI*. Springer, 291–330.
- [25] Mark Dunlop and John Levine. 2012. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2669–2678.
- [26] W.H. Edwards. 2010. *Motor Learning and Control: From Theory to Practice*. Cengage Learning. <https://books.google.fr/books?id=-Z3j9MLx7rgC>

- [27] G Ficca, P Lombardo, L Rossi, and P Salzarulo. 2000. Morning recall of verbal material depends on prior sleep organization. *Behavioural brain research* 112, 1-2 (July 2000), 159–163. DOI: [http://dx.doi.org/10.1016/s0166-4328\(00\)00177-7](http://dx.doi.org/10.1016/s0166-4328(00)00177-7)
- [28] Paul Morris Fitts and Michael I Posner. 1967. Human Performance (Belmont, CA: Brooks/Cole Publishing). (1967).
- [29] Robert M French. 1999. Catastrophic Forgetting in Connectionist Networks: Causes, Consequences and Solutions. *Trends in Cognitive Sciences* 3, 4 (1999), 128–135. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.480.7627>
- [30] Victor M. González and Gloria Mark. 2004. "Constant, Constant, Multi-tasking Craziiness": Managing Multiple Working Spheres. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 113–120. DOI: <http://dx.doi.org/10.1145/985692.985707>
- [31] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211* (2013). <https://arxiv.org/abs/1312.6211>
- [32] Wayne Gray and Jacquelyn Berry. 2018a. Limits to Training and Expertise in Helicopter Pilots and Tetris Players. *Symposia section - COGSCI 40th annual cognitive science society meeting* (2018).
- [33] Wayne D. Gray and Jacquelyn Berry. 2018b. TetLag and Other Costs of Task Switching Over Very Long Time Intervals. In *Symposium presentation at the 40th Annual Conference of the Cognitive Science Society*, Charles Kalish, Martina Rau, Jerry Zhu, and Timothy T. Rogers (Eds.). Cognitive Science Society, Austin, TX.
- [34] Jonathan Grudin. 1989. The case against user interface consistency. *Commun. ACM* 32, 10 (1989), 1164–1173.
- [35] Megan L Head, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. 2015. The extent and consequences of p-hacking in science. *PLoS biology* 13, 3 (2015), e1002106.
- [36] Alice F Healy, Erica L Wohldmann, and Lyle E Bourne. 2011. How does practice with a reversed mouse influence subsequent speeded aiming performance? A test of global inhibition. *Journal of Cognitive Psychology* 23, 5 (2011), 559–573. DOI: <http://dx.doi.org/10.1080/20445911.2011.547467org/10.1080/20445911.2011.547467>
- [37] Shamsi T. Iqbal and Eric Horvitz. 2007. Disruption and Recovery of Computing Tasks: Field Study, Analysis, and Directions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 677–686. DOI: <http://dx.doi.org/10.1145/1240624.1240730>
- [38] Lillian M Johnson. 1933. Similarity of meaning as a factor in retroactive inhibition. *The Journal of General Psychology* 9, 2 (1933), 377–389.
- [39] Jussi P P Jokinen, Sayan Sarcar, Antti Oulasvirta, Chaklam Silpasuwanchai, Zhenxin Wang, and Xiangshi Ren. 2017. Modelling Learning of New Keyboard Layouts. In *ACM Conference on Human Factors in Computing Systems (CHI)*. 4203–4215. DOI: <http://dx.doi.org/10.1145/3025453.3025580>
- [40] Wendy A Kellogg. 1987. Conceptual consistency in the user interface: Effects on user performance. In *Human-Computer Interaction-INTERACT'87*. Elsevier, 389–394.
- [41] David E Kieras and Susan Bovair. 1986. The acquisition of procedures from text: A production-system analysis of transfer of training. *Journal of Memory and language* 25, 5 (1986), 507–524.
- [42] Johan M Koedijker, Raoul RD Oudejans, and Peter J Beek. 2010. Interference effects in learning similar sequences of discrete movements. *Journal of motor behavior* 42, 4 (2010), 209–222.
- [43] John W Krakauer. 2009. Motor learning and consolidation: the case of visuomotor rotation. In *Progress in motor control*. Springer, 405–421.
- [44] John W Krakauer, Claude Ghez, and M Felice Ghilardi. 2005. Adaptation to visuomotor transformations: consolidation, interference, and forgetting. *Journal of Neuroscience* 25, 2 (2005), 473–478.
- [45] John W Krakauer, Pietro Mazzoni, Ali Ghazizadeh, Roshni Ravindran, and Reza Shadmehr. 2006. Generalization of motor learning depends on the history of prior action. *PLoS biology* 4, 10 (2006), e316.
- [46] Per Ola Kristensson and Shumin Zhai. 2007. Command Strokes with and Without Preview: Using Pen Gestures on Keyboard for Command Selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 1137–1146. DOI: <http://dx.doi.org/10.1145/1240624.1240797>
- [47] Don Lewis and Guy H. Miles. 1956. Retroactive Interference in Performance on the Star Discrimeter as a Function of Amount of Interpolated Learning. *Perceptual and Motor Skills* 6, 3 (1956), 295–298. DOI: <http://dx.doi.org/10.2466/pms.1956.6.3.295>
- [48] I. Scott MacKenzie and Shawn X. Zhang. 1999. The Design and Evaluation of a High-performance Soft Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 25–31. DOI: <http://dx.doi.org/10.1145/302979.302983>
- [49] I Scott MacKenzie and Shawn X Zhang. 2001. An empirical investigation of the novice experience with soft keyboards. *Behaviour & Information Technology* 20, 6 (2001), 411–418.

- [50] Laura A Malone, Erin VL Vasudevan, and Amy J Bastian. 2011. Motor adaptation training for faster relearning. *Journal of Neuroscience* 31, 42 (2011), 15136–15143. <http://www.jneurosci.org/content/jneuro/31/42/15136.full.pdf>
- [51] Denis Mareschal, Paul C Quinn, and Robert M French. 2002. Asymmetric interference in 3-to 4-month-olds sequential category learning. *Cognitive Science* 26 (2002), 377–389.
- [52] John A McGeoch and Arthur L Irion. 1952. The psychology of human learning. (1952).
- [53] Stephen Monsell. 1978. Recency, immediate recognition memory, and reaction time. *Cognitive Psychology* 10, 4 (1978), 465–501.
- [54] Georg Elias Müller and Alfons Pilzecker. 1900. *Experimentelle beiträge zur lehre vom gedächtniss*. Vol. 1. JA Barth.
- [55] Allen Newell and Paul S Rosenbloom. 1981. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition* 1, 1981 (1981), 1–55.
- [56] Donald A Norman and Diane Fisher. 1982. Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter. *Human Factors* 24, 5 (1982), 509–519.
- [57] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802. DOI: <http://dx.doi.org/10.1145/2470654.2481387>
- [58] Charles E. Osgood. 1949. The similarity paradox in human learning: a resolution. *Psychological Review* 56, 3 (1949), 132–143. DOI: <http://dx.doi.org/10.1037/h0057488>
- [59] C. Pallier, S. Dehaene, J.-B. Poline, D. LeBihan, A.-M. Argenti, E. Dupoux, and J. Mehler. 2003. Brain Imaging of Language Plasticity in Adopted Adults: Can a Second Language Replace the First? *Cerebral Cortex* 13, 2 (feb 2003), 155–161. DOI: <http://dx.doi.org/10.1093/cercor/13.2.155>
- [60] Stefan Panzer and Charles H Shea. 2008. The learning of two similar complex movement sequences: Does practice insulate a sequence from interference? *Human Movement Science* 27, 6 (2008), 873–887.
- [61] Peter G. Polson, Susan Bovair, and David Kieras. 1986. Transfer between text editors. *ACM SIGCHI Bulletin* 17, SI (may 1986), 27–32. DOI: <http://dx.doi.org/10.1145/30851.30856>
- [62] Leo Postman. 1971. Transfer, interference, and forgetting. In *Woodworth & Scholberg's Experimental Psychology* (3rd ed.), J. W. Kling and Lorrin A. Riggs (Eds.). Holt, Rinehart, and Winston, Inc., New York, 1019–1132.
- [63] Robert W Proctor and Kim-Phuong L Vu. 2007. Human information processing: an overview for human-computer interaction. In *The human-computer interaction handbook*. CRC Press, 60–79.
- [64] Vidya Ramesh, Charlie Hsu, Maneesh Agrawala, and Björn Hartmann. 2011. ShowMeHow: Translating User Interface Instructions Between Similar Applications. *Proc. UIST'11* (2011), 1–8.
- [65] Anthony Robins. 1995. Catastrophic Forgetting, Rehearsal, and Pseudorehearsal. *Connection Science* 7, 2 (1995), 123–146. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.3078>
- [66] R.A. Schmidt and T. Lee. 1988. *Motor Control and Learning 5th Edition*. Human Kinetics 10%. <https://books.google.fr/books?id=vBP09IHCz38C>
- [67] Jean Scholtz and Susan Wiedenbeck. 1990. Learning second and subsequent programming languages: A problem of transfer. *International Journal of Human Computer Interaction* 2, 1 (1990), 51–72. DOI: <http://dx.doi.org/10.1080/10447319009525970>
- [68] Andrew Sears and Julie A. Jacko. 2009. *Human-Computer Interaction Fundamentals* (1st ed.). CRC Press, Inc., Boca Raton, FL, USA.
- [69] Yilei Shi, Haimo Zhang, Hasitha Rajapakse, Nuwan Tharaka Perera, Tomás Vega Gálvez, and Suranga Nanayakkara. 2018. GestAKey: Touch Interaction on Individual Keycaps. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 596, 12 pages. DOI: <http://dx.doi.org/10.1145/3173574.3174170>
- [70] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Human-Computer Interaction - INTERACT 2005*, Maria Francesca Costabile and Fabio Paternò (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 267–280.
- [71] M.K. Singley and J.R. Anderson. 1989. *The Transfer of Cognitive Skill*. Harvard University Press. <https://books.google.fr/books?id=8UrN-09ZXuSc>
- [72] Sandra Trösterer, Magdalena Gärtner, Alexander Mirnig, Alexander Meschtscherjakov, Rod McCall, Nicolas Louveton, Manfred Tscheligi, and Thomas Engel. 2016. You Never Forget How to Drive: Driver Skilling and Deskillung in the Advent of Autonomous Vehicles. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (Automotive'UI 16)*. ACM, New York, NY, USA, 209–216. DOI: <http://dx.doi.org/10.1145/3003715.3005462>

- [73] Nefs Walker and Judith Reitmun Olson. 1988. Designing keybindings to be easy to learn and resistant to forgetting even when the set of commands is large. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 201–206.
- [74] Andrew Warr, Ed H. Chi, Helen Harris, Alexander Kuscher, Jenn Chen, Robert Flack, and Nicholas Jitkoff. 2016. Window Shopping: A Study of Desktop Window Switching. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3335–3338. DOI : <http://dx.doi.org/10.1145/2858036.2858526>
- [75] Leonard J West and others. 1998. The standard and dvorak keyboards revisited: Direct measures of speed. CiteSeer.
- [76] Jelte M Wicherts, Coosje LS Veldkamp, Hilde EM Augusteijn, Marjan Bakker, Robbie Van Aert, and Marcel ALM Van Assen. 2016. Degrees of freedom in planning, running, analyzing, and reporting psychological studies: A checklist to avoid p-hacking. *Frontiers in psychology* 7 (2016), 1832.
- [77] Virginia Wigmore, Christine Tong, and J Randall Flanagan. 2002. Visuomotor rotations of varying size and direction compete for a single internal model in a motor working memory. *Journal of Experimental Psychology: Human Perception and Performance* 28, 2 (2002), 447.
- [78] Erica L WOHLDMANN, Alice F HEALY, and Lyle E BOURNE. 2008. A Mental Practice Superiority Effect: Less Retroactive Interference and More Transfer Than Physical Practice. *Journal of experimental psychology. Learning, memory, and cognition* 34, 4 (2008), 823–833.
- [79] Yuko Yotsumoto, Takeo Watanabe, Li-Hung Chang, and Yuka Sasaki. 2013. Consolidated learning can be susceptible to gradually-developing interference in prolonged motor learning. *Frontiers in Computational Neuroscience* 7 (2013), 69. DOI : <http://dx.doi.org/10.3389/fncom.2013.00069>
- [80] Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 97–104. DOI : <http://dx.doi.org/10.1145/642611.642630>
- [81] Shumin Zhai, Alison Sue, and Johnny Accot. 2002. Movement Model, Hits Distribution and Learning in Virtual Keyboarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 17–24. DOI : <http://dx.doi.org/10.1145/503376.503381>
- [82] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an Invisible Keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 439, 13 pages. DOI : <http://dx.doi.org/10.1145/3173574.3174013>