# Computational Model of the Transition from Novice to Expert Interaction Techniques

Gilles Bailly, Mehdi Khamassi, Benoît Girard

# Computational Model of the Transition from Novice to Expert Interaction Techniques

GILLES BAILLY, Sorbonne Université, CNRS, France
MEHDI KHAMASSI and BENOÎT GIRARD*, Sorbonne Université, CNRS, France

Despite the benefits of expert interaction techniques, many users do not learn them and continue to use novice ones. This article aims at better understanding if, when and how users decide to learn and ultimately adopt expert interaction techniques. This dynamic learning process is a complex skill-acquisition and decision-making problem. We first present and compare three generic benchmark models, inspired by the neuroscience literature, to explain and predict the learning process for shortcut adoption. Results show that they do not account for the complexity of users' behavior. We then introduce a dedicated model, Transition, combining five cognitive mechanisms: implicit and explicit learning, decay, planning and perseveration. Results show that our model outperforms the three benchmark models both in terms of model fitting and model simulation. Finally, a post-analysis shows that each of the five mechanisms contribute to goodness-of-fit, but the role of perseveration is unclear regarding model simulation.

CCS Concepts: • **Human-centered computing → HCI theory, concepts and models**.

Additional Key Words and Phrases: Computational models; Interaction Techniques; Shortcut; Menus; Computational Rationality

**ACM Reference Format:**

## 1 INTRODUCTION

Expert interaction techniques such as keyboard shortcuts, gesture shortcuts or command languages allow users to reach a high level of performance in comparison with novice interaction techniques such as menus, palettes or ribbons. Expert interaction techniques are generally faster and let users focus on their main task because they do not rely on visual search [7]. However, they require an initial effort to learn how to use them and memorize the mapping between the commands and the corresponding shortcuts. This learning effort might be too high and many users, even experienced users do not adopt expert interaction techniques and continue to use what might appear as 'suboptimal' interactions [26, 87]. It results that several commercial (e.g. ShortcutFoo, KeyRocket, CheatSheet, Application Shortcut Mapper) and academic methods (e.g.[40, 44, 58, 65, 71, 72, 99, 99, 103]) are regularly developed to address this problem.

While several methods have been proposed, it remains unclear what are the human factors and cognitive mechanisms facilitating expert interaction technique adoption. Several theoretical constructs and frameworks have been proposed to explain why many users do not adopt expert interaction techniques [18, 36, 41, 43, 69]. This includes the Einstellung

*Both authors contributed equally to this research.

Authors' addresses: Gilles Bailly, Sorbonne Université, CNRS, France, first.name@sorbonne-universite.fr; Mehdi Khamassi, first.name@sorbonne-universite.fr; Benoît Girard, Sorbonne Université, CNRS, France, first.name@sorbonne-universite.fr.

effect [69], the "soft constraint hypothesis" [43] or the paradox of the active user [18]. For instance, users tend to "exploit" prior experiences (or previous techniques such as menus) to achieve the task (short term productivity) rather than "exploring" more efficient techniques such as shortcuts for long-term efficiency [18]. While these theories and frameworks present high-level explanations, they do not allow fine-grained predictions, i.e. simulating various cognitive, individual and environmental factors to test hypothetical designs and scenarios.

Predicting why, when and how users adopt or do not adopt expert interaction techniques is challenging because several phenomena are involved. First, it is a high-level *decision making* problem as the two interaction techniques co-exist: for each action, the users can decide to use the novice or expert interaction technique. This is also a *learning* problem as users, for instance, need to learn the mapping between the commands and the shortcuts to successfully exploit the expert interaction technique. Users can also forget what they have previously learned due to memory decay introducing complex learning and decision making dynamics.

In this paper, we focus on scenarios where a single operation can be performed with one among two possible interaction techniques, always available in a Graphical User Interface (GUI). One is dedicated to novice users and the other provides a higher level of performance for experienced users but requires learning efforts. A typical scenario is the transition from menus to keyboard shortcuts in the aim of selecting commands more efficiently. We present and compare computational models predicting the *whole* learning process leading (or not) to the adoption of shortcuts. In other words, the models explain and predict if and when users decide to learn shortcuts, then how they learn them and if ultimately they adopt them. Our general approach relies on computational rationality.

Computational rationality is at the convergence of artificial intelligence, cognitive science and neuroscience [39, 66]. The key idea is that humans (or animals) are rational agents with bounds: they choose actions maximising long-term expected rewards (or utility) given their limited cognitive resources (e.g. their own neural architecture), the constraints of the environment (e.g. task, technique, device) and their own experience. In practice, it consists of computing the values of actions, reflecting the long-term outcomes associated with these actions under uncertainty. In our context, we assume that users make a choice among three high-level actions before executing a command (as opposed to low-level actions, such as clicking): deciding to execute a command *as fast as possible* (1) within the novice technique or (2) with a known expert technique or (3) deciding to learn the expert technique now to be able to successfully use it later on. The problem the users face is which action to choose now to minimize time on a finite horizon given limited cognitive resources, e.g. memorization, and the constraints of the environments, i.e. the available set of commands and their relative frequencies.

Reinforcement Learning (RL) is an appropriate formal framework for computational rationality and to study subtle interactions between learning and decision-making [27, 39, 57, 96]. For instance, it is extensively used in neuroscience [57, 96], cognitive sciences [27] and more recently in HCI [21, 22] to determine the policies that maximise long-term expected utility.

We first introduce three benchmark RL models in neuroscience, which are widely used in decision-making tasks involving a learning process [101]: Rescorla-Wagner, Choice Kernel and the combination of both. They have in common to rely on an exploration - exploitation mechanism. While the Rescorla-Wagner model learns the expected value of each action based on the history of previous rewards, the Choice Kernel model captures the tendency of users to repeat previous actions regardless of their outcome. Rescorla-Wagner+Choice Kernel mixes the two models. These model-free RL models, i.e. RL models without a representation of the environment, have several advantages to model how users adopt expert interaction techniques. They are task-independent, have few free parameters, are easy to implement, are fast and have been shown to well capture learning and decision-making dynamics in different contexts [101].

We compare these models on the data collection of Grossman et al. [44] investigating the impact of three interaction methods on keyboard shortcut adoption. One key aspect is to apply cutting-edge methods from the decision-making field [101] to HCI: We first compare their goodness of fit to reflect the capacity of these models to replicate each participant's trial-by-trial action choice, i.e. whole learning process for each participant. We then estimate the best parameters for each model and each participant. We finally simulate these models with the best parameters to test whether they do reproduce the main behavioral properties of the participants. The results suggest that these benchmark models are not sufficient to capture the complexity of shortcut adoption.

We then present a novel computational model, called Transition, dedicated to explain and predict shortcut adoption. It relies on the computational rationality principles and is inspired by neuroscience. The core of this model is the combination of five mechanisms to update the cognitive state of the users:

(1) a *planning* mechanism reflects the ability of users to consider several actions ahead. This mechanism is necessary to explain why users invest some time *now* foreseeing the benefits of using shortcuts *later*.
(2) a mixture of *implicit* and *explicit learning* mechanisms serves to consolidate at different learning rates the command-to-shortcut mapping in memory.
(3) a *decay* mechanism reflecting that the command-to-shortcut mappings encoded in memory fades due to the passage of time.
(4) a *perseveration* mechanism, based on evidence in neuroscience [101], reflects the fact that users are likely to repeat the previous strategy, regardless of the strategy.

We tested our model on the Grossman et al. data collection [44]. Results show our Transition model outperforms the three benchmark models both in term of likelihood and BIC score. The Transition model also synthesises more realistic data when simulating.

We then conducted a post-analysis to better understand the impact of the different mechanisms involved in our model. To achieve this, we compared five variants of the model by enabling/disabling the different mechanisms. Results show that (1) all the proposed mechanisms play a role in shortcut adoption as they improve the goodness-of-fit; but (2) the variant of the Transition model without the perseveration mechanism better synthesizes human behavioral data.

From a methodological point of view, these results highlight the importance of combining the two evaluation methods [80] — goodness of fit and model simulation — to validate models of shortcut adoption as they provide different perspectives on the model and its variants.

In summary, our primary contribution is the development, analysis and evaluation of a new computational model of expert interaction technique adoption and more precisely shortcuts adoption. This model is a first step towards a better understanding of the complex learning dynamics involved in expert interaction technique adoption; Our long term objective is to facilitate designers workflow to choose interaction techniques. Indeed, computational models can serve to analyze different designs and scenarios by running model simulations, reducing the cost (time, money) of experimental studies. Once these models can evaluate a design, they can be integrated in optimisation algorithms to propose high-value solutions [10] for a *population of users*. Finally, these models can be embedded in intelligent systems to dynamically predict the effect of an intervention at the level of an individual (instead of the population). They allow for an AI to assist *individual users* and promote the adoption of interaction methods best suited at their task [94].

Finally, our contribution is also to promote valuable cross-disciplinary exchanges on questions, models and methods between neuroscience and HCI about user behavior with interactive systems involving subtle interactions between learning and decision making such as the challenging transition from novice to expert interaction techniques.

## 2 RELATED WORK

We first contextualize our research in the field of command selection with a focus on the transition from novice to expert interaction techniques. We then provide background in Reinforcement Learning on which our computational models are built.

### 2.1 Command Selection

*2.1.1 Novice and expert Interaction Techniques.* Common interfaces make several interaction techniques available to select a command. Novice interaction techniques such as menus, toolbar, ribbons, palette etc. require little training as they rely on visual guidance (recognition). They are easy to discover, to learn and to use [7]. However, they require visual attention, and several operations to execute a command. For instance, selecting the command "Edit > Find > Replace" in Microsoft Word menubar requires three pointing and click operations which are time consuming.

Expert interaction techniques such as keyboard shortcuts, gesture shortcuts and command lines generally rely on "recall" forcing users to make some efforts to learn how to execute commands [7]. Expert interaction techniques are intented for more experienced users. They have been shown to be faster as they require less operations [4, 17, 59, 77, 79, 85]. For instance, users can execute a command and choose the parameters with a simple gesture. Moreover, expert interaction techniques can be performed partially or totally eyes-free, i.e. without visual feedback, letting users focus on their main task [7].

Several studies show that many people do not use these expert interaction techniques despite their benefits [60, 81, 92]. These studies motivated the design of several methods to favor the transition from novice to expert interaction techniques. For instance, in the context of keyboard shortcuts, previous methods include the use of advanced feedback mechanisms, e.g. visual or audio feedback [11, 44, 72, 102], feedforward mechanisms [40, 71], the use of easy-to-learn mappings [11, 65, 99, 103] or temporal penalties [44, 58]. For instance, Grossman et al. [44] present and compare different methods: Audio is a method playing the keyboard shortcut orally by a voice synthesizer when a command is executed in the menu to expose the users to the shortcut; Disabled is a method letting the user navigate through the menu, but does not allow clicking on the items to execute it. It forces users to execute keyboard shortcuts. Both methods favor keyboard shortcut use. Similar methods (feedback, feedforward, penalty, etc.) have been proposed to favor the use of gesture shortcuts, e.g. [4, 8, 45, 59] as well as command lines [87, 95].

These methods aim at promoting awareness of the expert techniques, motivate their use, facilitate the learning and/or improving their performance to favor adoption. We build on this literature to elaborate our model as it identifies and highlights key factors (e.g. temporal cost of the method, the nature of the feedback, etc. ). However, it remains a long term challenge to predict and explain how these factors precisely interact together and their magnitude. It also remains unclear why users do not adopt expert interaction techniques.

*2.1.2 Theories and framework.* Several theoretical constructs have been proposed to explain why users do not use expert interaction techniques [18, 36, 41, 43, 69, 87]. This includes the Einstellung effect [69], the "soft constraint hypothesis" [43] or the paradox of the active user [18]. For instance, users tend to "exploit" prior experiences (or previous methods such as menus) to achieve the task (short term productivity) rather than "exploring" more efficient methods such as shortcuts for long-term efficiency [18]. Users tend to favor well-practiced methods with fast and incremental feedback rather than methods based on recall [36].

Some frameworks [42, 87] characterize phenomena related to intramodal and intermodal expertise development. For instance, Scarr et al. [87] highlight three main reasons why users would not adopt an expert interaction technique:

users are not aware that a (more efficient) expert interaction technique is available; they can under-estimate the benefits of the these techniques; the temporal performance dip when switching is perceived as too high. Gray and Lindstedt [42] extend this framework and study how individuals discover and invent new methods to develop their expertise. In particular they highlight three main phenomena: plateaus, dips and leaps. Users first reach a performance "plateau" (or performance ceiling [87]) with a *given* method (i.e. novice interaction technique) after practice. Authors distinguish performance plateaus and performance asymptote, the latter being reached only with the *optimal* method (typically the expert interaction technique). "Dips", refer to the performance dip, when users explore, experiment, learn and switch modalities, typically when they make the transition from the novice to the expert interaction technique. This performance dip is essential because it can prevent users to adopt the expert interaction technique and maintain users in a local optimum. They would then not be able to experience performance "leaps" offered by the use of expert interaction techniques. In neuroscience, several studies show similar human (and animal) behaviors where the decision process is not necessarily based on long-term rewards, but also on short-term rewards [32], in particular when a learning effort is required [100] or even without rewards [73], e.g. habits, intrinsic motivations, etc. These studies suggest that Reinforcement Learning (model-free, model-based, or both) can play a key role to explain these behaviors [30].

These works present high-level explanations. However, they do not allow for fine-grained predictions. They do not permit to predict which users, when and how they make the transition from novice to expert interaction techniques. In this article, we build on this theoretical grounding and present a computational model of this transition allowing to simulate various cognitive, individual and environmental factors to test hypothetical designs and scenarios.

*2.1.3 Computational Models.* Several computational models have been proposed in the field of command selection. They generally predict selection time in linear menus depending on several factors such as menu organisation, menu length, item position or item frequency [9, 15, 21, 24, 25, 47, 61, 67]. They rely on empirical laws of pointing (Fitts' law [35]) and visual search (e.g. [12]). However, only few of them focus on the learning process by considering practice as a factor [9, 25, 67, 94]. Among them, Bailly et al. [9] combine two visual search strategies (serial and directed search) and a pointing component which are modulated by practice. The learning component relies on the Power Law of Practice (PLP) [76]. This law is appropriate at the population level, but does not capture individual learning dynamics.

Very few computational models have been proposed to estimate the production time of expert interaction techniques and focus on gesture shortcuts [16, 49, 63, 83]. For instance, the CLC model [16] predicts the amount of time it takes for users to make a gesture shortcut based on its geometry. The model partitions the gesture into segments, where each segment is a Curve, a straight Line, or a Corner. The total time to execute this gesture shortcut is the sum of the time to produce each segment. However, the model includes neither a learning component to reflect how users performance evolves with practice nor a decision making component to predict if and when users adopt gesture shortcuts. We are not aware of a computational model to explain or predict how users switch from a novice to an expert interaction technique.

Our work is also related to computational models of habits and/or behavioral change [13, 55, 74, 93], where “habits" refers to the cognitive associations between users' behaviors and the triggering of contexts. These models aim to predict the habit strength as a function of behavior repetition. While these models can be sufficient to explain the behavior of users who *only* use menus, they can not explain the behavior of users who switch to shortcuts without external interventions. Our models include a perseveration mechanism (as a Choice Kernel, see below) to reflect habits, but also learning and planning mechanisms necessary to explain shortcuts adoption.

*2.1.4 Summary.* In summary, the transition from novice to expert interaction techniques is a well identified and long-time challenge in HCI with several technical, theoretical, and empirical contributions. However, no computational model has been proposed to predict how users adopt expert interaction techniques probably because it involves both learning and decision making phenomena. Existing computational models mainly focus on human performance with menus or well learned gesture shortcuts. In contrast, this article presents a computational model to predict and explain how users adopt shortcuts. It relies on the Reinforcement Learning framework.

### 2.2 Reinforcement Learning

*2.2.1 Markov Decision Process.* Markov Decision Process (or MDP) is a mathematical framework for decision making under uncertainty [48]. The MDP is a four-tuple $(S, A, P, R)$ where $S$ is a set of states (also called state space), $A$ a set of actions (action space), $P$ the state transition probability for going from a state $s$ to state $s'$ after performing action $a$ ($P(s'|s, a)$) and $R$ a function ($R : S \times A \rightarrow \mathbb{R}$ ) returning the immediate reward as a function of the state $s$ and the performed action $a$. The goal for an agent in a Markov Decision Process is to perform the serie of actions that maximize the expected cumulative random reward:

$$E[\sum_{t=0}^{t=\infty} \gamma R(s_t, a_t)]$$

where $\gamma \in [0, 1]$ is the discount factor determining the importance of future rewards.

In several applications, the agent is a user, an animal, a robot; the actions are the behaviors of the agent (e.g., clicking a button, pressing a lever) and the state characterizes the environment (e.g. current state of the interface, a position within a maze) and the rewards are obtained from the environment (e.g. task achieved, food). Sometimes, the agent can not fully observe the environment. The problem can then be formulated as a POMDP, a partially observable MDP. This specific formulation have been used several times in HCI [21, 22], cognitive science [28] and neuroscience [30] to model human behaviour.

*2.2.2 Reinforcement Learning.* Reinforcement Learning (RL) solves MDP (or variants such as POMDP) by learning state action value function $Q(s, a)$. $Q(s, a)$ is a real scalar value which represents the estimated expected value of executing action $a$ in the state $s$ as a common currency for potentially any type of long-term reward, i.e. time, food, money, etc. There are two main classes of algorithms: model-free and model-based. Model-free algorithms use neither the state transition probability function nor the reward function from the MDP to estimate the Q-values. Examples of model-free algorithms include Q-Learning, Rescolar-Wagner (RW), Choice Kernel (CK). In the following sections, we use (and detail) the RW and CK algorithms as well as their combination (RWCK) to predict human behavior and shortcut adoption.

In contrast, Model-based algorithms exploit the state transition probability function and the reward function from the MDP. They have the advantage to be much more efficient to find the optimal solutions [51], but at the expense of a high computational cost [19, 34]. In the second part of this article, we present model-based algorithms to predict and explain shortcut adoption.

*2.2.3 Reinforcement Learning and HCI.* Reinforcement Learning is receiving an increasing interest in many fields (cognitive science, neuroscience) and recently in HCI [13, 20–22, 37, 38, 52, 62, 64, 70, 89, 94]. In HCI, several perspectives are used to represent the user and the system in the MDP framework. In the “machine perspective", the agent represents the system and the user is part of the environment providing some reward, i.e. teaching to the system how to react to users actions [33]. In the “user” perspective, the agent represents the user, the environment includes the system/interface. The primary goal is then to understand users’ behavior, for instance, understanding how visual search strategies spontaneously emerge based on previous experiences with the interface [21]. However, these models can also be embedded in the system. For instance, Todi et al. [94] present a system simulating different machine and user behaviors and choose the best adaptions.

Our approach relies on the latter (user perspective). Our models aim at predicting and explaining users' behavior when facing both novice and expert interaction techniques. It is a first step towards the elaboration of intelligent systems that dynamically predict and trigger interventions to foster the adoption of expert interaction techniques.

*2.2.4 "Learning" as a cognitive process?* "Learning" in Reinforcement Learning (RL) refers to how the *algorithm* incrementally updates the State-action values (Q-values) to determine the optimal policy. However, the dynamic of the Q-values does not necessarily reflect the cognitive process of skill/knowledge acquisition of the agent. It depends on the objective and thus the field of research: Machine Learning or Neuroscience.

In Machine Learning, the dynamic of the Q-values only reflects the quality of the solver. The faster the Q-values converge during the training phase, the better is the solver to find the optimal policy. Previous RL-based HCI models generally adopt this perspective. During the training phase, the dynamics of the Q-values do not reflect how the user/agent learns. Once the optimal policy is determined, the model can be simulated using *static* Q-values. The model then predicts how users behave once they reach a plateau of performance, i.e. once they have already learned the task [13, 21, 22, 38]. For instance, Chen et al. [21] study visual search in previously unseen menus and acknowledge that their RL model does not aim to explain "how people learn specific menus and the location of specific items". Todi et al. [94] recently proposed a model-based RL algorithm to predict how users find and select items in a linear menu. While they introduce a learning component, the Q-values are only used to train the model. The learning dynamic is encoded in the model, i.e. it is estimated from the history of actions by using the base-level equation (ACT-R [3]) at the end of each session.

In Neuroscience, the approach is generally different. The dynamic of the Q-values in RL models is of importance as it reflects how the agent (human, animal) learns [101]: the Q-values varies during the simulation of the model and illustrates the fact that the choice of actions for a same state can evolve with practice. The dynamics of Q-values' evolution is also of paramount importance in Neuroscience since it can be used to analyze whether neural representations of action values, as recorded with brain imaging, reflect the same dynamics as reinforcement learning models or not. Finally, precisely analyzing the dynamic of the Q-values as obtained with model simulations is also important in this field so as to compare whether the learned Q-values of different models can account for behavioral properties observed in humans at different moments across task learning. Such simulations are critical to not only compare different parameter values of the same model and whether they are consistent with the dynamic of Q-values, but also to compare different models and eliminate those that cannot account for the properties of human behavior [101].

Our approach builds on this Neuroscience perspective as we aim at predicting and explaining how the users learn the task, i.e. efficiently executing commands, when repeatedly facing the same set of commands. This is a more challenging objective as our models should predict *all* trial-by-trial actions (instead of the last trials once participants reached a plateau of performance).

*2.2.5 Summary.* Our main contribution is the elaboration and validation of the first computational model of the transition from menus to shortcuts. Because we address complex phenomena related to not only decision making and learning dynamics, we build on the field of neursocience, approaching the RL framework differently.

## 3 APPROACH

Command selection relates to the execution of commands (e.g. "Open") using several interaction techniques. In this article, we consider the Menu (M) as novice interaction technique and Shortcuts (S) as expert interaction techniques, although the following models could be extended to other methods (e.g. palettes, ribbons). We now describe the type of decisions the models focus on, the formulation of the problem and the general approach.

### 3.1 Type of decisions

When the users have to execute a command, they make several decisions:

- Decision about the **strategy**. Users first choose which strategy to use: executing the command as fast as possible (1) using the menu, (2) using a known shortcut or (3) learning the shortcut now to be able to successfully use it later.
- Decision about the **mapping**. The second level of decision is choosing which item to click on, the gesture to perform or which keys to press given the chosen strategy. For instance, given the functionality "Quit" and the choice of using keyboard shortcuts, users have to decide which combination of keys to press (e.g. Ctrl+Q or Shift+Q). This can also happen when interacting with the menu as the user does not necessarily know how the desired functionality is entitled in the menu [7].
- Decision about **execution**. Finally, the last level of decision is a plan for movement. For instance, for executing "Ctrl+Q" without looking at the keyboard, users might decide which finger to use, which might depend on the keyboard layout.

So, choosing between behavioral strategies is hierarchically organized [46, 56]. Our main goal is to better understand *when users decide to learn, then learn and then adopt shortcuts*. We therefore focus on understanding how users choose **strategies** (the highest level of this hierarchy). We thus leave as future work several phenomena related to the interaction within a menu (e.g. the position of items) and the specificities of the shortcuts (e.g. the position of keys on the keyboard, the shape of the gesture). In particular, we do not aim at explaining the nature of errors, e.g. explaining why users press “Shift+Q" or “Ctrl+A" when executing the command “Quit". Instead, we focus on behavioral changes at the strategy level.

### 3.2 Problem formulation

The problem of shortcut adoption can be described as a discrete-time stochastic control of Markov Decision process.

*3.2.1 State space S:.* A state $s \in S$ represents the target command to execute (e.g. “Open"). A specificity of our definition is that the next state only depends on the frequency of the commands. In the next sections, we indifferently use state or command when referring to $s$.

*3.2.2 Action space A:.* Given a command, the user chooses an action $a \in A$. In our context the set of actions are the **M**enu strategy ($a_M$), the (keyboard or gesture shortcut) **S**hortcut strategy ($a_S$) and the **L**earning strategy ($a_L$). $a_M$ and $a_S$ are characterized by the fact that users execute the command *as fast as possible* with the corresponding method (menu or shortcut). In contrast, the learning strategy ($a_L$) consists in opening the menu, dedicating some time to explicitly learn the shortcut mapping and then executing the command (either with the menu or a shortcut). Indeed, opening a menu does not only serve to select a command but also to gather information about the shortcut by gazing at the visual cue on the right side of the menu item [7].

#### 3.2.3 *Q-Value.*

The expected value of an action $a$ in a state $s$ at the time $t$ is represented by the *Q-value* $Q(s_t, a)$. In our context, a *Q-value* represents the relative expected benefit of using a specific strategy for executing a given command. During the simulation of an interaction task, the Q-values of each strategy are not static, they evolve, trial after trial, based on the history of interactions and the environment and thus constitute predictions of each participant's upcoming behavior.

The way the Q-values are updated, or used to choose actions is model-dependent and described in the following sections.

#### 3.2.4 *Model's Input/output.*

The input of the model is a command $s_t$ and its output is the chosen strategy (or action) for this command. We remind that our model describes a whole learning process (the policy is not fixed). The strategies evolve over time depending on the evolution of the Q-values.

### 3.3 Computational rationality

Our problem formulation is inline with the computational rationality view of human behavior [39, 66] where the users' strategies (policy) emerge from the user's goal (utility), their cognitive mechanisms and the task environment:

#### 3.3.1 *Utility.*

The user aims at minimizing total execution time for executing commands.

#### 3.3.2 *Cognitive mechanisms.*

Plethora of cognitive mechanisms are likely to be involved in the learning and decision making process of adopting shortcuts. In this article, we consider five main mechanisms: decay, perseveration, planning, implicit learning and explicit learning and their different combinations. We detail these mechanisms in the following sections.

#### 3.3.3 *Task environment.*

We consider two main aspects for defining the task environment. First, the **sequence of commands**. Liu et al. [68] show that the user's behavior is sensitive to different frequency distributions and the execution time of a given command depends on, not only its frequency, but also the frequency of the other commands. It is thus important to refine the definition of sequence of commands because three components might influence shortcut adoption: 1) the size of the set of unique commands (e.g. "Open", "Save"), 2) the total number of command execution and 3) the relative frequency distribution (e.g. uniform distribution, Zipfian distribution, etc. [68]).

The second aspect is related to the **teaching methods** available to favor shortcut usage. While several methods have been proposed, it still remains unclear how they modify users' behavior (see Section Related Work). In this article, we focus on the three teaching methods tested in [44]: Traditional, Audio and Disabled.

### 3.4 Outline

We first introduce three benchmark RL models in neuroscience, which are widely used in decision-making tasks involving a learning process [101] (section 4). We then present the data collection of Grossman et al. [44] (section 5) and state-of-the art methods from the decision-making field (section 6) to evaluate and compare the models. Results (section 7) show that these benchmark models are not sufficient to capture the complexity of shortcut adoption. We then present our model in sections 8 and 9 and evaluate it in the sections 10 and 11.

Table 1. Key notations

| Notation | Description |
|---|---|
| $s$ | State: Target command |
| $a$ | Action: User strategy |
| $Q(s,a)$ | Q-Value |
| $a_p$ | Previous action |

Table 2. Free (top) and task-related (bottom) parameters of the model. The range of the free parameters is the one used to fit the models

| Symbol | Range | Description |
|---|---|---|
| $\alpha$ ($\alpha_{RW}, \alpha_{CK}$) | $[0, 1]$ | Learning rate |
| $\beta$ ($\beta_{RW}, \beta_{CK}$) | $[1, 20]$ | Softmax inverse temperature |
| $T_{KS}$ | 0.9 | Keyboard shortcut strategy time |
| $T_M$ | 2 | Menu strategy time |
| $T_L$ | 3.8 | Learning strategy time |
| $c_p$ | 3 | temporal penalty associated to an error |

## 4 BENCHMARK RL MODELS

We are not aware of existing models of shortcut adoption. We thus chose three benchmark models in Cognitive Neuroscience which are widely used in decision-making tasks involving a learning process [101]. These models are especially appropriate for multiarmed bandit problem[5] where the agent receives a reward after each action. Indeed, in our context, users choose a strategy and immediately receive a reward (e.g. the inverse of the execution time) at each trial. The different notations and parameters are summarized in the Table 1 and Table 2.

### 4.1 Rescorla-Wagner (RW)

In this model, the agent learns the expected value $Q$ of each action based on the history of previous rewards:

$$Q(s_{t+1}, a) = Q(s_t, a) + \alpha\ (\ r(s_t, a) - Q(s_t, a)\ ) \tag{1}$$

where $\alpha$ is the learning rate and $r(a, t)$ is the reward of using the action $a$ at the date $t$. Here, the reward, is the inverse of the execution time. To simplify the problem, we assume that the time only depends on the strategy with: $T_S < T_M < T_L$ Where $T_S$, $T_M$, $T_L$ are respectively the execution times of the strategies Shortcut, Menu and Learning, which are empirically estimated. The RW model is a myopic version of standard temporal-difference learning algorithms [90], such as Q-learning, where the discount factor $\gamma = 0$.

To compare and choose the action given the *Q-values*, the model relies on a Boltzmann soft-max function. This function converts the Q-values into action probabilities $P(a|s_t)$:

$$P(a|s_t) = \frac{e^{\beta\, Q(s_t,a)}}{\sum_a e^{\beta\, Q(s_t,a)}} \tag{2}$$

where the parameter $\beta$ is the inverse temperature which controls the trade-off between exploitation and exploration, i.e. a small value of $\beta$ reflects almost random choice (exploration) while a high value of $\beta$ indicates that the user always chooses the action with the highest Q-value (exploitation).

This model has only two parameters $\alpha$ (eq. 1) and $\beta$ (eq. 2)

### 4.2 Choice Kernel (CK)

This model captures the tendency of users to repeat previous actions regardless of execution time. It computes $CK$ values:

$$CK(s_{t+1}, a) = CK(s_t, a) + \alpha\ (\ (a == a_p) - CK(t, a)\ ) \tag{3}$$

where $\alpha$ is the learning rate and $a_p$ the previous action. The equation 2 then converts the $CK$ values into action probabilities. The Choice Kernel model also has two parameters ($\alpha$, $\beta_{CK}$).

The CK model is thus insensitive to the rewards. While this model can appear too simple, several models without rewards have been shown to well capture human behavior in different contexts [73]. The CK model is part of these models and focuses on the *perseveration* effect: It is based on evidence in neuroscience [101] that human beings are likely to repeat the previous strategy: not only repeating the Menu strategy, but potentially also the Shortcut strategy once the shortcuts have been learned.

### 4.3 Rescorla-Wagner + Choice Kernel (RWCK)

This model mixes the two previous models, following the principle that subjects both try to maximize reward and tend to show some degree of perseveration at the same time. The model estimates the action probabilities according to the equation 4:

$$P(a|s_t) = \frac{e^{\beta_{RW}\ Q(s_t,a)+\beta_{CK}\ CK(s_t,a)}}{\sum_a e^{\beta_{RW}\ Q(t,a)+\beta_{CK}\ CK(t,a)}} \tag{4}$$

The behavior of the agent is thus sensitive to the reward (inverse of the execution time) and to the strategies previously used. This model has four parameters ($\alpha_{RW}$,$\beta_{RW}$, $\alpha_{CK}$,$\beta_{CK}$).

### 4.4 Discussion

These three models have in common to rely on an exploration - exploitation mechanism. While the Rescorla-Wagner model learns the expected value of each action based on the history of previous rewards, the Choice Kernel model captures the tendency of users to repeat previous actions regardless of the reward. Rescorla-Wagner+Choice Kernel mixes the two models. These model-free RL models have several advantages to model how users adopt expert interaction techniques. They are task-independent, have few free parameters, are easy to implement, are fast and have been shown to well capture learning and decision-making dynamics in different contexts [101].

## 5 DATA COLLECTION

We compare the three models (RW, CK, RWCK) on Grossman et al. data [44]. We summarize the experimental design and the collected data.

### 5.1 Experimental Design

The interface consists of a menu bar opening 6 drop-down menus and a button at the bottom of the screen (Figure 1). The participants move the cursor within the button and press the space bar to start the trial. The button then displays an image representing the command to be executed as a stimulus (1). The participants execute the command by selecting the corresponding item in the menu (2-3) or by executing the corresponding keyboard shortcut. When an error occurs, the participants have to wait for 3s before the command can be executed again. The trial ends when the participants hit again the space bar with the cursor within the button (4).

This study compares three interaction techniques: (1) the **Traditional** menu visually highlights the keyboard shortcut of the selected item; (2) the **Audio** menu offers audio feedback: The keyboard shortcut was played orally by a voice synthesizer once the command was executed from the menu. Finally, (3) the **Disabled** menu lets the user navigate through the menu, but does not allow clicking on the item to execute it. This forces users to use keyboard shortcuts.

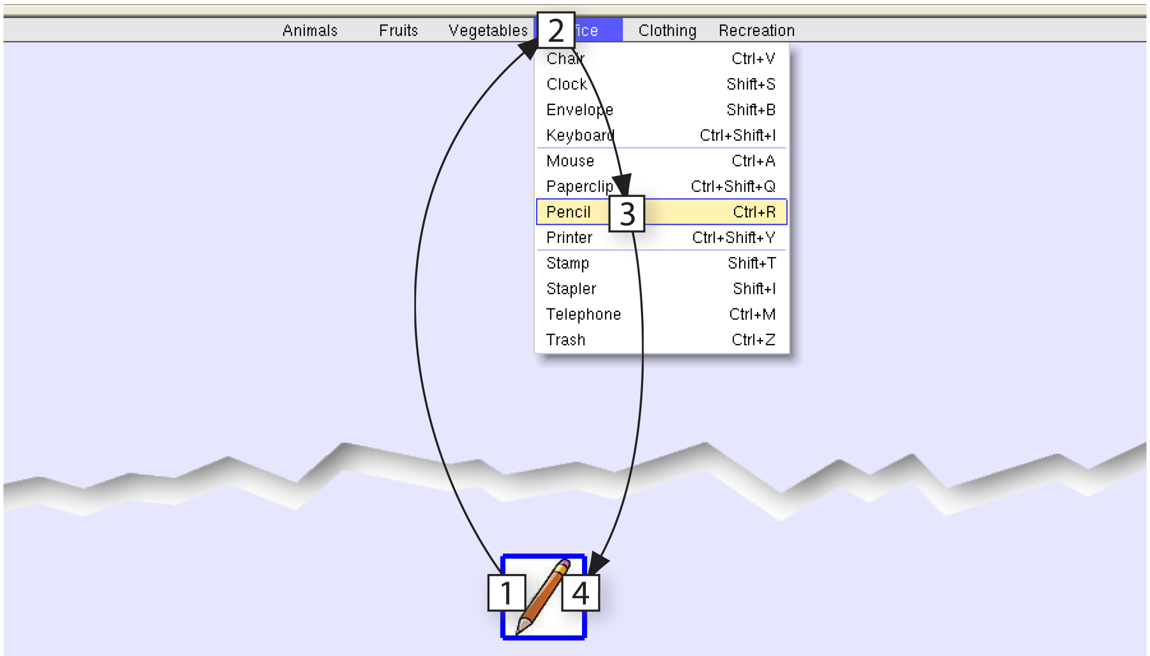


Fig. 1. Experimental task. Participants move the cursor within the button at the bottom of the screen and hit the space bar to display the stimulus, an image representing the command to select (1). The participants then execute this command by selecting the corresponding item ("Pencil") in the menu (2-3) or by executing the corresponding keyboard shortcut (Ctrl+R). The trial finishes when the participants press again the space bar with the cursor within the button. Reprinted from [44] with the permission of the authors.

The fourteen commands have different frequencies. The most frequent commands are executed 144 times and the least frequent commands are executed 12 times. Each of the 42 participants (12 female, 30 male, ranged in ages from 18-28) is assigned to a technique (14 participants per techniques) and executes 720 commands organized in 12 blocks of 30 executions. Figure 9-top shows the 144 executions of the most frequent command of the user 1 with the Audio technique.

position the cursor inside of it, and then hit the space bar

### 5.2 Collected Data and derived strategies

At each trial, the participant's id, technique, block, trial, target command with frequency, name and keyboard shortcut are recorded. The dependent variables are time (ms), success (0/1) and the method: Menu or Keyboard shortcuts. Because they are not directly recorded in the logs, we derive the three strategies as:

- *Menu* strategy when the user only uses the menu method.
- *Shortcut* strategy when the user only uses the keyboard shortcut method.
- *Learning* strategy when the menu is visited, but the keyboard shortcut is executed[1]

[1]When the participant learned the keyboard shortcut, but executed the command in the menu, the Learning strategy cannot be detected. This behavior is interpreted as the Menu strategy. The number of *Learning* is thus under-estimated and the number of *Menu* is over-estimated.

### 5.3 Task-related parameters

This study includes a 3s penalty ($c_p = 3$) when an error occurs. We also analyzed the empirical data and estimated the correct execution time for each strategy: $T_M = 2.0$, $T_{KS} = 0.9$, $T_L = 3.8$. We used the mode (rather than the mean or the median) because the distributions were highly skewed.

## 6 EVALUATION METHODS

The objective of this section is to present the methods to estimate the parameters of the three models as well as the methods to compare them.

### 6.1 Parameter estimation

A main challenge in elaborating and using computational cognitive models is the number of *parameters* as well as their variability across individuals. In some contexts, the value (or the distribution of values) of the parameters are known and can be directly adopted from the literature [21, 52]. However, in our context of shortcut adoption, we do not have specific priors on the values of these parameters (see table 2). We thus aim to estimate values of the parameters that best explain behavioral data, i.e. the parameters that minimize the fitness function.

*6.1.1 Maximum-Likelihood Approach.* The fitness function reflects the capacity of a model to replicate a participant's trial-by-trial action choice. In Bayesian terms, it is the likelihood of the data given the model, that is the maximum probability that the model chooses the same series of actions as the participant [27, 96]. It consists of analyzing the posterior prediction of the model *conditioned on the past history*, i.e. evaluating the likelihood of the participant's action $a_t$ given the past data $d_{1:t-1}$, where the past data includes the actions made by the participant, *not* the actions made by the model. Formally, we estimate :

$$LL(m, p) = \sum_t \log P(a_t^p | S_{1:t-1}^p, m, \theta_m^p) \quad (5)$$

where $m$ is the tested model, $p$, a given participant, $\theta_m^p$ the set of parameters of the model $m$ for the participant $p$ and $S_{1:t}^p$ the sequence of actions performed by the participant until the date $t$. We use the differential evolution algorithm as optimization method (from the Scipy.opimize python library [97]) to find the set of parameters $\theta_m^p$ which maximizes $LL(m, p)$ for each model $m$ and each participant $p$.

*6.1.2 Fitness function properties.* Our fitness function has two key properties. First, it considers *individual models* (i.e. a different set of parameters for each participant) rather than a population model (the same set of parameters for all participants), which is important to address inter-individual variability in decision-making problems [52]. Indeed, different users can have radically different policies leading to different behaviors. Consider an extreme case with two users, one using only Menu and one using only Keyboard shortcuts. The notion of "average" user does not mean that she will use 50/50 Menu and Keyboard shortcuts.

Moreover, our fitness function considers *trial-by-trial action choices* rather than aggregate fitting as it fits each action individually. While it is not common practice in HCI, this approach is now well adopted in cognitive sciences and neuroscience [27, 96]. This permits to model the temporal evolution of participant's behavioral strategy, e.g. initially using menus and then progressively switching to shortcuts, rather than modeling again an average 50/50 Menu and Keyboard shortcuts for a single participant.

## 6.2 Model fitting

*6.2.1 Log-Likelihood comparison.* We aim to determine which of the three models best describes the behavioral data, as a way to understand which mechanisms underlie behavior. Given the best identified parameters $\theta_m^p$, we compare their likelihood $LL(m, p)$ (equation 5). The model with the largest likelihood is likely to better explain the observed data.

*6.2.2 BIC score comparison.* In the process of model comparison, it is common to include a penalty term for model complexity, i.e. for the number of parameters [101]. The Bayesian Information Criterion (BIC) is commonly used [23] and estimated as $BIC = -2LL + k \times log(N)$ where $LL$ is the likelihood (equation 5), $k$, the number of parameters and $N$, the number of points to predict. As each participant executes 720 commands in the experiment, $N = 720$. It is common practice to consider that there is a "strong evidence" in favor of the winning model when the BIC difference is > 6 [84].

Table 3 and 6 report both likelihood and BIC score.

## 6.3 Model Simulations

We can use the best set of parameters $\theta_m^p$ to simulate the different models. In some cases, model simulation can lead to very different results from model fitting if the path of actions sampled by the participant is widely different from the paths likely to be selected by the model [80, 101]. It is thus important to also simulate the models and verify that they do reproduce the main behavioral properties of the participants [101], in our context, the evolution of the percentage of correct shortcut execution, which is commonly used to compare interaction techniques favoring shortcuts [7]. For each model, we ran 50 simulations[2] per participant using individual parameters, (i.e. $50 \times 42 = 2100$ *simulations* per model). We then aggregated per technique (14 participants).

Table 3. Comparisons of three benchmark model-free RL models and our Transition model in term of free parameters, total number of free parameters (N), Likelihood and BIC. Our Transition model minimizes both the inverse of the likelihood and the Bic score.

| **Model** | **Free parameters** | **N** | **- Likelihood** | **BIC** |
|---|---|---|---|---|
| Rescorla-Wagner (*RW*) | $\beta_{RW}, \alpha_{RW}$ | 2 | 193.7 | 400.7 |
| Choice Kernel (*CK*) | $\beta_{CK}, \alpha_{CK}$ | 2 | 174.0 | 361.2 |
| *RWCK* | $\beta_{RW}, \alpha_{RW}, \beta_{CK}, \alpha_{CK}$ | 4 | 159.5 | 345.4 |
| Transition | $\alpha_E, \alpha_I, d, h, w, \beta$ | 6 | **148.5** | **336.5** |

# 7 RESULTS

We now present our fitting and simulation results at different levels of granularity.

## 7.1 Fitting Results for Action Choices

*7.1.1 Model.* Table 3 indicates the likelihood and BIC score for the three benchmark RL models. As expected, there are strong evidence (BIC difference > 6) that the combination of the Rescorla-Wagner model and Choice Kernel model, RWCK ($LL = -159.5$; $BIC = 345.4$) outperforms each model individually, ie, Choice Kernel, CK ($LL = -174.0$; $BIC = 361.2$) and Rescolar-Wagner, RW ($LL = -193.7$;$BIC = 400.7$) even when considering the penalty associated to the BIC score for additional parameters. Interestingly, the CK is the second best model while it is myopic to rewards and tends to repeat the previous strategies.

[2] a compromise between testing models and a reasonable expenditure of experimenter effort

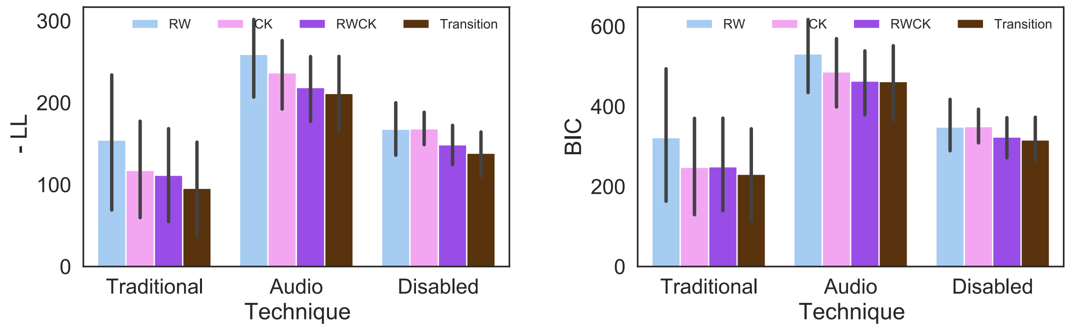


Fig. 2. Model comparisons in term of likelihood (LL) and BIC Score for each technique. The lower the better. Error bars show 95% bootstrap confidence intervals. TRANSITION outperforms the three benchmark RL models (Rw, CK and RWCK) both in term of Likelihood and BIC score.

*7.1.2 Interaction Method level.* Figure 2 compares the three models for each interaction method. Results indicate strong evidence (BIC difference > 6) that RWCK outperforms CK and RW for both Audio (RWCK: 463.4; CK: 486.4; RW:531.4) and Disable (RWCK: 323.6; RW: 348.6; CK: 349.5). However, for Traditional, where users are less likely to transition and thus to repeat the same Menu strategy, both CK (247.8) and RWCK (249.2) outperform RW (322.0).

*7.1.3 User level.* Finally, we analyze the data for each participant. Results indicate that $RWCK$ is the best model (BIC score) for 26 participants, $CK$ for 9 participants (7 of them using Traditional) and $RW$ for 7 participants. These results provide a complementary picture illustrating the variety of users' behavior, i.e. there is not a single model that best fits all participants. "Simple" behaviors such as rarely using shortcut can be explained with a simple $CK$ model (e.g. only implementing the perservation mechanism). However, this model fails as soon as the users really make a transition.

*7.1.4 Parameters.* Figure 3 illustrates the distribution of the values of parameters for the three models.

### 7.2 Model simulations

*7.2.1 Block-by-block: Evolution of shortcuts.* Figure 4 shows the evolution of shortcut use (%) per block and per method for the three benchmark models. We also report the Mean Square Error (MSE) as a measure of discrepancy. Surprisingly, we observe here that RWCK is not the best model to synthesize users' behavioral data. RW (MSE=274.9) outperforms RWCK (MSE=329.9) and CK (MSE=855.1) is by far the last model. However, a closer inspection reveals that none of them is fully satisfactory. First the initial percentage of shortcut use is too high regardless of the model and the method. Second, the performance of Audio is always under-estimated, regardless of the model.

*7.2.2 Trial-by-trial: individual participant actions.* We visually inspected the 588 (42 users × 14 commands) sequences of strategies for each model. Figure 9 is one example illustrating the limit of the RWCK simulations to reproduce users' behavior. Indeed, we observe several instances where the models switch back to menus (or learning) for a long period ($> 7 trials$). This can be explained by the fact that "optimal" $\beta_{RW}$ and/or $\beta_{CK}$ are small enough to favor exploration even after having switched to shortcuts. In comparison, we did not observe this pattern in participants' data.

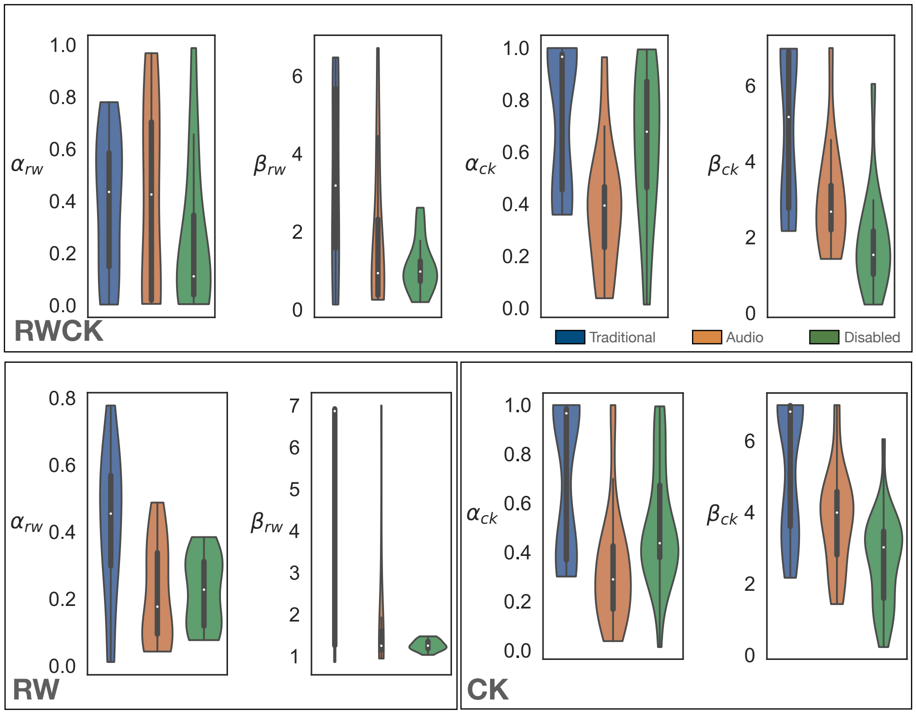


Fig. 3. Summary of the parameters per technique and per model.

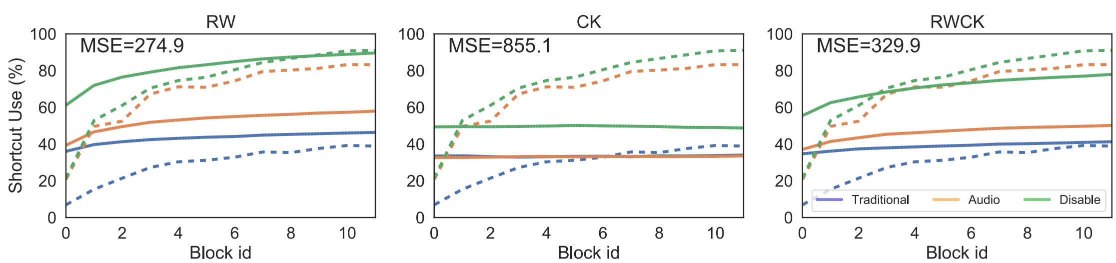


Fig. 4. Shortcut use (%) per block and method. Observed participants' data are represented with dots. Synthetised data (solid line) are produced by aggregating 50 simulations per participant with individual parameters.

### 7.3 Discussion

In summary, these model fitting results indicate that the Rescorla-Wagner+Choice Kernel (RWCK) model better accounts for the empirical data, suggesting that both the adaptation to reward (from RW) and the perseveration (from CK) mechanisms play a role in explaining and predicting the transition from menus to shortcuts. Moreover, the Choice Kernel (CK) is the second best model while it is myopic to the rewards, suggesting that perseveration is an important mechanism to explain users' behavior. However, simulation results suggest that Resocla-Wagner (RW) better synthetizes users' behavioral data. So, the role of perseveration is at this point not clear: while it significantly contributes to the goodness-of-fit (model fitting), the data produced by the models *without* perseveration better reflect participants' behavior (model simulation).

From a methodological point of view, our results highlight the importance of combining model fitting and model simulation [80] to validate models of shortcut adoption as they show a different picture. Finally, a close inspection of our results suggests that this first set of classical models are not satisfactory as they tend to overestimate initial shortcut adoption and underestimate the performance of the Audio technique. This motivates us to elaborate a dedicated model of shortcut adoption, called Transition.

## 8 TRANSITION: MODEL OVERVIEW AND THEORETICAL ASSUMPTIONS

In the previous sections, we showed that the benchmark model-free RL models in neurosciences are not sufficient to explain users' behaviors. In this section, we present a model-based RL model dedicated to explain and predict shortcut adoption. It also relies on the computational rationality principles but combines five mechanisms, grounded in neuroscience and cognitive science, likely to participate in the transition from menus to shortcuts:

The first mechanism, **planning** [29, 31, 53, 78, 88], means that the users are able to consider $h$ actions ahead for a given command, i.e. they mentally simulate their future strategy choices in response to the $h$ next times. A higher $h$ means that the users are more likely to transition as they can foresee the future benefits of learning shortcuts now. Moreover, humans do not necessarily value all future actions/strategies with the same weight [18, 88]. A discount factor is generally introduced to determine the importance of future rewards [91]. However, this planning mechanism is not sufficient *alone* as it can only explain behavior when users never transition (short horizon) or transition immediately (larger horizon).

Then, **implicit and explicit learning** mechanisms consolidate at different learning rates the command-to-shortcut mapping in memory. When repeatedly selecting an item in the menu, users unconsciously and slowly gather information about the shortcut thanks to their peripheral vision (implicit learning). When reaching a certain level of knowledge, users can then perceive the benefits of intentionally learning shortcuts (explicit learning). Some evidence in neuroscience indicate that learning for action selection relies on a balance between planning and implicit/explicit learning [78].

A **decay** mechanism [3, 14] reflects that the command-to-shortcut mappings encoded in memory fades away due to the passage of time.

The fifth mechanism, **perseveration**, reflects the fact that users are likely to repeat the previous strategy. This general behavioral tendency, at the heart of the Choice Kernel Model, has been documented for a long time in decision-making, from psychological, neuroscience and modeling points of view [1, 39, 57, 101].

In the next section, we describe our model using the RL formalism (e.g. states, actions, Q-Values, etc.). We then evaluate it and further demonstrate that these mechanisms are necessary altogether to explain and predict the transition from menus to shortcuts. Finally, we discuss the limitations of the model and provide several directions to refine it.

## 9 TRANSITION: MODEL DEFINITION

The different notations and parameters are summarized in Table 4 and Table 5.

### 9.1 State and Action

We reused the same definitions for the states (the target to execute) and the actions: the **M**enu strategy ($a_M$), the **S**hortcut strategy ($a_S$) and the **L**earning strategy ($a_L$).

Table 4. Key notations

| Notation | Description |
|---|---|
| $s$ | State: Target command |
| $a$ | Action: User strategy |
| $Q(s, a)$ | Q-Value |
| $C(s, a)$ | (Temporal) Cost |
| $E_C(s, a)$ | Expected Cost |
| $E_{CC}(s, a)$ | Expected Cumulative Cost |
| $C_T(s, a)$ | Successful execution cost |
| $C_R(s, a)$ | Repair cost |
| $K(s, a)$ | User knowledge |
| $a_p$ | Previous action |

Table 5. Free (top), fixed (center) and task-related (bottom) parameters of the model. The range of the free parameters is the one used to fit the model

| Symbol | Range | Description |
|---|---|---|
| $\beta$ | $[1, 20]$ | Softmax temperature |
| $w$ | $[0, 1]$ | tendency to repeat the previous action |
| $\alpha_E$ | $[0, 1]$ | Explicit learning rate |
| $\alpha_I$ | $[0, 0.33]$ | Implicit learning rate |
| $d$ | $[0, 0.02]$ | Decay |
| $h$ | $[0, 7]$ | Horizon |
| $\gamma$ | 0.9 | Discount factor [91] |
| $T_{KS}$ | 0.9 | Keyboard shortcut strategy time |
| $T_M$ | 2 | Menu strategy time |
| $T_L$ | 3.8 | Learning strategy time |
| $c_p$ | 3 | temporal penalty associated to an error |

### 9.2 Q-values

$Q(s_t, a)$ remains the expected value of an action $a$ in a state $s$ at the time $t$. To compare those Q-values and finally choose an action, the model relies on the Boltzmann soft-max function of Equation 2. The expected value $Q(s_t, a)$ is now calculated as:

$$Q(s_t, a) = (1 - w) \times (-E_{CC}(s_t, a)) + (a = a_p) \times w \tag{6}$$

where $E_{CC}(s_t, a)$ is the expected cumulative temporal cost of using the strategy $a$ to execute the command $s_t$; $a_p$ is the previous action used for the command $s_t$ and $w \in [0, 1]$ is a weight reflecting the tendency of people to repeat the previous action, i.e. the degree of **perseveration**. In other words, Equation 6 reflects the fact that the agent faces a multi-objective optimization problem by trying to minimize the expected cumulative temporal cost of current command execution while maximizing stability in the choice of the strategy. The formulation of the action values is thus quite similar to the one of RWCK 4, which corresponds to weighting into a common currency the RW values (reward) and CK values (perseveration).

Before defining the expected cumulative cost $E_{CC}$, i.e. the cost associated to a sequence of actions, we first need to define the cost function $C(s_t, a)$ and the expected cost $E_C$.

### 9.3 Cost function $C(s_t, a)$

The temporal cost $C$ to execute a command is the sum of the execution time $C_T$ and the repair time $C_R$ in case of error:

$$C(s_t, a) = C_T(s_t, a) + b \times C_R(s_t, a) \tag{7}$$

where $s_t$ is the target command at time $t$, $a$ the chosen strategy and $b$ a Boolean indicating whether users perform an error or not. To simplify the problem, we assume that the correct execution time $C_T$ only depends on the strategy with:

$$T_S < T_M < T_L \tag{8}$$

where $T_S$, $T_M$, $T_L$ are respectively the correct execution times of the strategies Shortcut, Menu and Learning. The repair Time $C_R$ is the sum of the time to analyse the error (or penalty) $c_p$ and the time to correctly re-execute the command.

To simplify, we consider the users reuse the Menu strategy to repair their errors:

$$C_R(s_t, a) = C_T(s_t, a) + c_p \tag{9}$$

## 9.4 Expected cost $E_C(s_t, a)$

$E_C(s_t, a)$ is the expected temporal cost of using the strategy $a$ to execute the command $s_t$. It derives from Equation 7 and is the weighted sum of the correct execution time $C_T(s_t, a)$ and incorrect execution time $C_R(s_t, a)$, where the weight depends on the user knowledge $K(s_t, a)$:

$$E_C(s_t, a) = K(s_t, a) \times C_T(s_t, a) \quad + \quad (1 - K(s_t, a)) \times (C_T(s_t, a) + C_R(s_t, a_t)) \tag{10}$$

## 9.5 Knowledge $K$

*9.5.1 Definition.* $K(s_t, a) \in [0, 1]$ is a latent variable representing the knowlegde of the user. It is the probability to successfully execute the command $s_t$ with the strategy $a$. More precisely, $K(s_t, a_M) = K(s_t, a_L)$ and represents how well the mapping between a command and its location in the menu is encoded in the user's memory (the user only interacts with the menu with these two strategies). Reciprocally, $K(s_t, a_S)$ represents how well the mapping Command-to-Shortcut is encoded in the user's memory.

$K(s_t, a_S)$ is a key variable to explain the transition from menus to shortcuts. Indeed, it is likely that users will not try to execute shortcuts if they do not have enough prior knowledge, i.e. if the probability of success is not high enough. In contrast the knowledge of item locations $K(s_t, a_M) == K(s_t, a_L)$ is likely to have an impact on menu selection time (amount of visual search), but less on accuracy (pointing task in a menu has a high accuracy) and the transition to shortcuts. For this reason, one simplification is to assume that the users have a "perfect" knowledge of the location of menu items for a given command, i.e. the probability of successfully selecting the target item in the menu is equal to 1:

$$K(s_t, a_M) = K(s_t, a_L) = 1 \tag{11}$$

We can then rewrite Equation 10 for the strategies Menu and Learning, assuming that users do not make errors:

$$E_C(s_t, a) = K(s_t, a) \times C_T(s_t, a), \quad a \in \{a_M, a_L\} \tag{12}$$

*9.5.2 Updating Knowledge.* We propose 2+1 mechanisms to update $K(s_t, a_S)$. The two first mechanisms are **explicit and implicit learning**. Explicit learning occurs when users successfully use the Learning strategy or the Shortcut strategy: The users intentionally read/learn the shortcut cue or execute the shortcut correctly. Implicit learning occurs when users repeatedly execute a command in the menu: the users unconsciously gather information in the surroundings thanks to their peripheral vision. Explicit and implicit learning depend on the strategy and are used to increase the knowledge of shortcuts:

$$\begin{aligned} K(s_t, a_S) &= K(s_t, a_S) + \alpha_E \times (\ 1 - K(s_t, a_S)\ ) \\ K(s_t, a_S) &= K(s_t, a_S) + \alpha_I \times (\ 1 - K(s_t, a_S)\ ) \end{aligned} \tag{13}$$

Where $\alpha_E$ and $\alpha_I \in [0, 1]$ are the **e**xplicit and **i**mplicit learning rates. While explicit learning is more efficient than implicit learning to memorize the shortcut mapping ( $\alpha_E >> \alpha_I$ ), we will demonstrate (section Results) that implicit learning is essential for explaining the transition from menus to shortcuts.

The third mechanism is **decay**. At each time step, the shortcut knowledge of each command $s_t$ is updated to account for memory decay:

$$\forall s \in S, \quad K(s_t, a_S) = K(s_t, a_S) + d \times (\ 0 - K(s_t, a_S)\ ) = K(s_t, a_S)(\ 1 - d\ ) \tag{14}$$

Where $d \in [0, 1]$ is the decay factor. The mechanisms to update memory ($\alpha_E$, $\alpha_I$, $d$) are related to the ones in ACT-R [3] but this definition is more appropriate to an RL framework and does not require to store the whole user history.

### 9.6 Expected Cumulative Cost $E_{CC}$

We can now define the expected cumulative cost $E_{CC}$ used in Equation 6. We formulate the problem of command selection as a **planning** problem with an horizon $h$, i.e. users plan a sequence of $h$ actions for a given command to minimize the expected cumulative cost for this command:

$$E_{CC}(s, a, h) = E_C(s, a) + \gamma \times E_{CC}(s, argmin_a(\ E_C(s, a)\ ), h - 1) \tag{15}$$

where $E_C(s_t, a)$ is the expected temporal cost (Equation 10), $h$ is the horizon, and $\gamma \in [0, 1]$ is a discount factor determining the importance of future rewards. Typically, $\gamma$ close to 0 indicates that users only consider the temporal cost of the current strategy, while $\gamma$ close to 1 indicates that the weight of each strategy in a given horizon is very similar. In the RL literature [91], it is common to choose $\gamma = 0.9$. The two parameters $h$ and $\gamma$ allow to control for the cognitive bias consisting in valuing more the present than the future (in line with theoretical constructs such as the paradox of active users [18]). Our hypothesis is that users with a large horizon are more likely to perceive the benefits of learning shortcuts *now* so as to use them *in the future*. In practice, the users estimate the cumulative cost $E_{CC}(s_t, a)$ of each of the $3^h$ decision branches and choose the one with the minimal cost as if the commands were performed in a row. To achieve this, they simulate each decision and their effect on the internal values, i.e. the shortcut knowledge necessary to estimate the utility of each strategy.

## 10 VALIDATION

### 10.1 Methods

We test our model, Transition, on the Grossman et al. data collection [44] and compare its likelihood and simulation performance to the three benchmark models RW, CK and RWCK.

### 10.2 Fitting results for Action Choices

*10.2.1 Overall.* Table 3 indicates the likelihood and BIC score of the Transition model. The results indicate that our model outperforms the best benchmark RL model, RWCK in terms of likelihood ( Transition: -148.5; RWCK: -159.4). More surprisingly, despite the larger number of parameters, the results indicate strong evidence (BIC difference >6) in favor our model (Transition: 336.5; RWCK: 345.4).

*10.2.2 Method level.* Figure 2 compares Transition to the three benchmark RL models for each method. Results indicate a strong evidence (BIC score >6) that Transition outperforms the best benchmark models for Traditional ( Transition: 230.6; CK: 247.9) and for Disabled ( Transition: 316.3; RWCK: 323.6). Results do not show significant differences between Transition (462.7) and RWCK (463.4) regarding the Audio method.

*10.2.3 User level.* Results indicate that the best model (BIC score) is Transition for 31 participants, RWCK for 8 participants, RW for 2 participants and CK for 1 participant. It is a strong difference with the comparison of the three

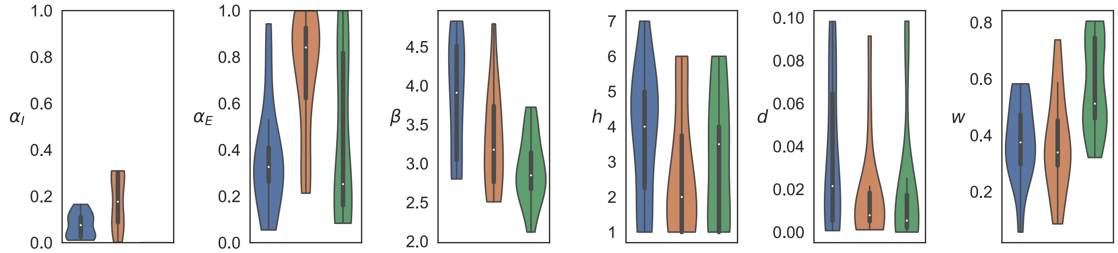


Fig. 5. Summary of the Transition model parameters per methods: Traditional (blue), Audio (orange), Disabled (green).

benchmark models where RWCK was the best model for 21 participants. These results indicate that Transition better captures the variability of users' behavior, regardless of the interaction method.

*10.2.4 Parameters analysis.* Figure 5 illustrates the distribution of values for each parameter and each method. To analyze these parameters we distinguish Traditional and Audio which have three actions and Disabled which has only two (The menu strategy is not available).

Regarding Traditional and Audio, results confirm that the distribution of values for parameters related to users' profile – ability to plan $h$, decay $d$ and tendency to repeat previous actions $w$ – seem independent from the interaction method. Results also confirm that participants tend to learn explicitly $\alpha_E$ and to try $\beta$ shortcuts more often with Audio than with Traditional. More surprisingly, Audio feedback seems to also influence implicit learning $\alpha_I$ while we were expecting this to be method-independent.

Regarding the user profiles' parameters of Disabled ($h$, $d$, $w$), the results are similar except that $w$ appears higher probably due to the fact that this method relies on two actions instead of 3. Similarly, users tend to learn and try shortcuts more easily than Traditional. Finally, Disabled does not have an implicit learning parameter $\alpha_I$ (as the Menu Strategy is not available) explaining probably the longer tail for $\alpha_E$.

### 10.3 Model simulations

*10.3.1 Block-by-block: Evolution of shortcuts.* Figure 6 illustrates the percentage of correct shortcuts per block and per method. The results indicate that Transition synthesizes data which better reflect users' behavior (MSE=40.7) than the three benchmark RL models (RW: 274.9; RWCK:329.9; CK: 855.1). Indeed, we observed that the simulation of Transition better captures the learning dynamics and thus does not suffer from the two limitations of the simulation of the benchmark RL models: The initial performances predicted by the model are now close to the one observed for our participants and the prediction of the relative performance between the methods (in particular Audio) well reflects the one of the actual techniques. However, a closer inspection reveals that the predicted performances of the three methods are slightly over-estimated during the first blocks (0-6) and slightly under-estimated during the last blocks (7-11) in comparison with observed participants' data.

### 10.4 Discussion

Our results indicate that our Transition model outperforms the three benchmark models both in terms of model fitting (likelihood and BIC score) and simulation. The results are especially impressive regarding the quality of the

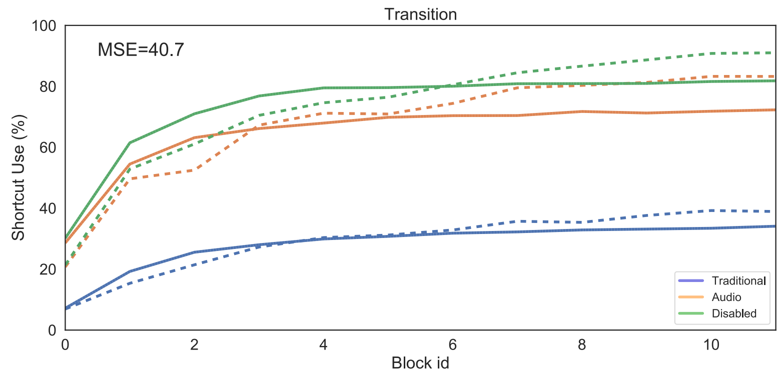


Fig. 6. Shortcut use (%) per block and method for the TRANSITION model. Observed participants' data are represented with dots. Synthetised data (solid line) are produced by aggregating 50 simulations per participant with individual parameters.

Table 6. Comparison of the TRANSITION model with five variants where one mechanism has been disabled in term of free/fixed parameters, total number of free parameters (N), Likelihood and BIC. Removing one mechanism from the TRANSITION model decreases not only the likelihood but also the BIC score suggesting that the five mechanisms contribute to explain and predict the transition from novice to expert interaction techniques.

| **Model** | **Free parameters** | **Fixed parameters** | **N** | **- Likelihood** | **BIC** |
|---|---|---|---|---|---|
| TRANSITION | $\alpha_E, \alpha_I, d, h, w$ + $\beta$ | $\gamma = 0.9$ | 6 | **148.5** | **336.5** |
| TRANSITION - $\alpha_E$ | - , $\alpha_I, d, h, w$ + $\beta$ | $\gamma = 0.9$ + $\alpha_E = 0$ | 5 | 199.2 | 431.4 |
| TRANSITION - $\alpha_I$ | $\alpha_E$, - , $d, h, w$ + $\beta$ | $\gamma = 0.9$ + $\alpha_I = 0$ | 5 | 164.8 | 362.5 |
| TRANSITION - $d$ | $\alpha_E, \alpha_I$, - , $h, w$ + $\beta$ | $\gamma = 0.9$ + $d = 0$ | 5 | 162.2 | 357.3 |
| TRANSITION - $h$ | $\alpha_E, \alpha_I, d$, - , $w$ + $\beta$ | $\gamma = 0.9$ + $h = 1$ | 5 | 156.1 | 345.1 |
| TRANSITION - $w$ | $\alpha_E, \alpha_I, d, h$, - + $\beta$ | $\gamma = 0.9$ + $w = 0$ | 5 | 175.9 | 384.7 |

synthetized data for our model in comparison with the ones synthetized by the benchmark RL models. Indeed, our TRANSITION model well reflects for each method the absolute and relative evolution of shortcut use over time.

## 11 MODEL VARIANTS

Our model combines five key mechanisms, but it remains unclear whether all of them are useful to replicate the trial-by-trial evolution of strategy choices. We thus decided to compare five variants of our models (Table 6). Each of these variants corresponds to the TRANSITION model where one of the mechanism (e.g. implicit learning) has been disabled. The objective is to study the influence of disabling each mechanism on likelihood and BIC score.

### 11.1 Fitting Results for Action Choices

*11.1.1 Overall.* Table 6 summarizes the five variants of our model depending on the different combinations of free and fixed parameters. We observe that the best model both in terms of likelihood and BIC score is the one implementing the five mechanisms, thus the full TRANSITION model (Table 6): implicit learning $\alpha_I$, explicit learning $\alpha_E$, decay $d$, planning $h$, and perseveration ($w$). The second best model is the one without planning which has a difference of BIC score larger

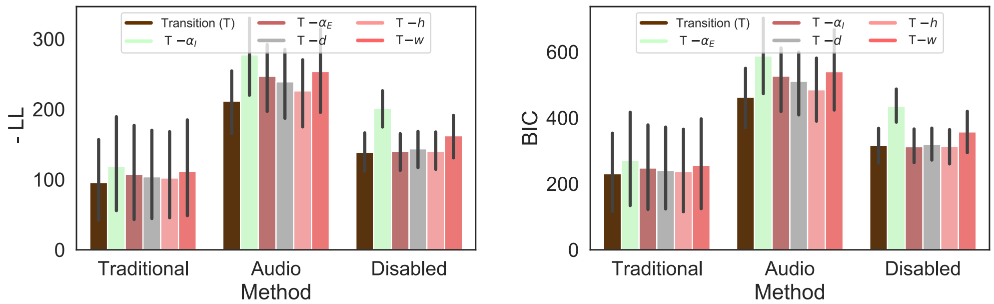


Fig. 7. Comparison of the Transition with its variants where a mechanism has been disabled in term of Likelihood (Left) and BIC Score (Right) per technique. The lower, the better. Error bars show 95% bootstrap confidence intervals

than 6 (336.5 vs. 345.1). These results suggest that all five mechanisms contribute to explain the transition from menus to shortcuts in these participants.

*11.1.2 Method level.* Figure 7 summarizes the goodness-of-fit (likelihood and BIC) of each model variant per method. We observe that the model implementing all five mechanisms, Transition, outperforms (Likelihood and BIC) all variants for Traditional and Audio.

Regarding Disabled, Transition and the variant without $\alpha_I$ has the same likelihood (138.4) and outperform the other variants. The fact that, that these two models have the same likelihood is not surprising as the implicit learning mechanism is not used in this interaction method: Disabled does not let users use the Menu strategy and thus can not implicitly learn shortcuts. However, Transition is penalized with the BIC score as $\alpha_I$ is not used. In term of BIC score, the variant without planning $T - h$ (312.9) is similar to $T - \alpha_I$ (312.6) and outperform Transition (316.3). These results refine our understanding of interacting with the Disabled technique: Not only users do not implicitly learn keyboard shortcuts as the menu is disabled but they also do not need to plan as the choice of strategies is limited.

*11.1.3 User level.* Results indicate that the best model (likelihood) is Transition for 24 participants, the one without planning for 7, the one without implicit learning for 4, the one without decay for 4, the one without explicit learning for 2 and finally the one without perseveration for 1. However, when considering the BIC score, no model really emerges: none of the models is the best model for more than 12 participants out of 41. Altogether, these results indicate that the 5 mechanisms are necessary but not with the same weight for each participant / interaction method.

### 11.2 Model simulations

We analyze the simulated data at different levels of granularity:

*11.2.1 Block-by-block: Evolution of keyboard shortcuts.* Figure 8 illustrates the percentage of shortcuts per block and per interaction method for each model variant. The simulation of model variants provides a slightly different picture than model fitting. Indeed, two model variants, the one without decay $d$ (MSE=26.3) and the one without perseveration $w$ (MSE=26.3) outperform Transition (MSE=39.9).

These results echo the ones obtained when comparing the three benchmark RL models. Indeed, both RWCK and Transition were the best models in term of goodness of fit, but their variants without perseveration (i.e. RW and

Transition - $w$ ) better synthesize data. The good performance of the variant without decay $d$ is surprising. One possible explanation is that the absence of decay artificially compensates for the presence of perseveration when simulating data. For this reason, we also analysed the performance of the variant T$-d-w$ corresponding to the Transition model without the decay and perseveration mechanisms. Regarding goodness-of-fit, as expected T$-d-w$ does not outperform the previous variants in terms of likelihood (206.4) and BIC score (439.2). Regarding model simulation, the results are similar, where T$-d-w$ (MSE = 41.4) does not outperform the variant without decay T$-d$ (MSE=26.3) and T$-w$ (MSE=26.3), confirming our hypothesis that the absence of decay artificially compensates for the presence of perseveration when simulating data. Further investigations are necessary to precisely understand the role of perseveration.

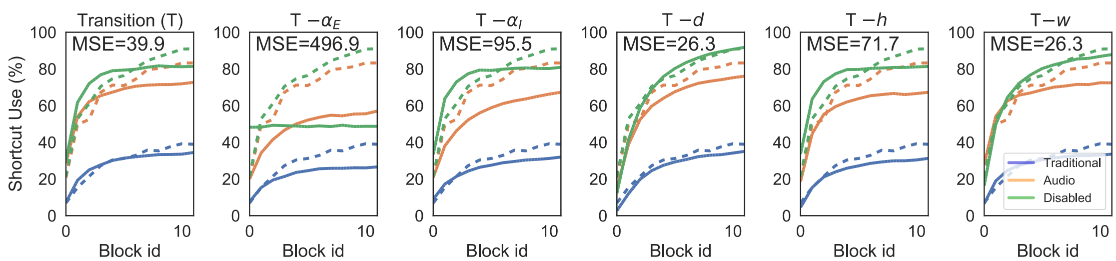


Fig. 8. Keyboard shortcut use (%) per block and interaction method for each model variant: observed participants' data are represented with dots. Synthetised data (solid line) where produced by aggregating 50 simulations per participant with individual parameters. MSE is calculated for each model variant.

#### 11.2.2 *Trial-by-trial: individual participant actions.*

Figure 9 shows one of the sequences of command executions for one participant using the audio method: The first row shows the participant data. For the definition of the transition (yellow box), we used the data of [6] where two experts annotated all the sequences of strategies from the Grossman et al. experiment [44]. The second and third rows illustrate the synthesized data from the Transition model and its variant (T$-w$) without the perseveration mechanism. From our observations, we found that the T$-w$ model better reflects the participant's transition than the Transition model both in terms of beginning and duration as well as in terms of variability of the strategies before, during and after the transition. The example of Figure 9 is representative of many sequences.

### 11.3 Discussion

In summary, (1) the results indicate that Implicit learning, explicit learning, Decay and Planing mechanisms play a role both in explaining and predicting the transition from menus to keyboard shortcuts; (2) the role of perseveration is less clear: while it significantly contributes to the goodness-of-fit (model fitting), the data produced by the models *without* perseveration better reflect participants' behavior (model simulation). Altogether, the models Transition and $T-W$ appear the most promising models for the transition from menus to shortcuts. Finally, (3) our results highlight the importance of studying model variants as well as combining model fitting and model simulation [80] to validate models of shortcut adoption.

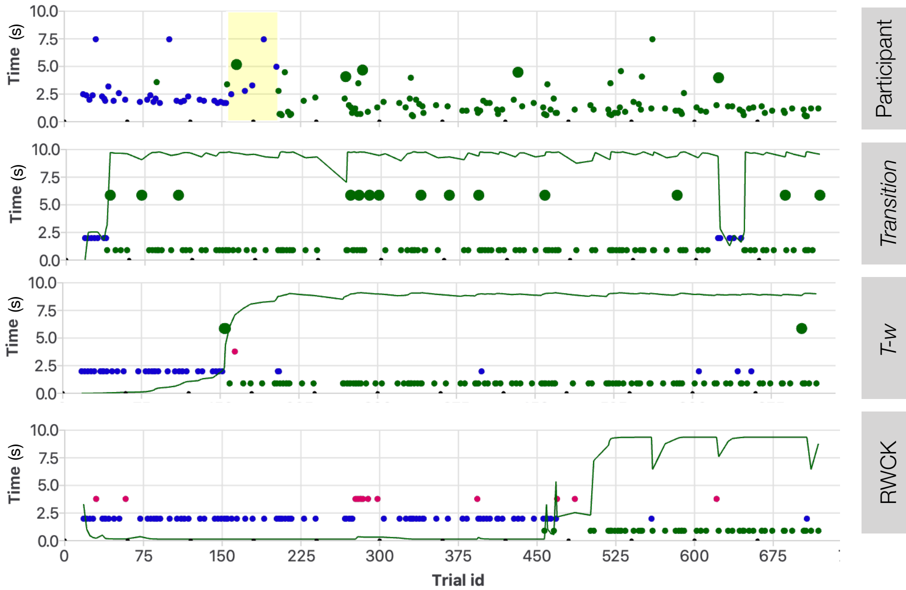


Fig. 9. Simulation comparisons ( Transition (T), T−$w$, $RWCK$) for a single command and a single user using the AUDIO interactive method. Each dot represents one command execution (144 executions for this command). The color encodes the strategy (Menu: Blue; Keyboard shortcut: Green; Learning: Pink); The height encodes the execution time; Finally, large dots indicate errors. For this participant (top row), the model T−$w$ (third row) better reflects the observed transition (indicated as a yellow box on the participant data). The green line indicates the probability of the agent to execute shortcuts. We observe that this one is much more stable for T−$w$ than the two other models because of the lack of the perseveration mechanism.

## 12 DISCUSSION AND FUTURE WORK

In this section, we summarize our main contributions regarding the design and the empirical evaluation of the Transition model. We then analytically evaluate this model in light of the criteria of Jacobs and Grainger [50] providing directions for future work. Finally, we discuss the opportunities of neuroscience research to model complex HCI tasks such as the transition from novice to expert interaction techniques.

### 12.1 Model of the transition from novice to expert interaction techniques

In this paper, we presented a new model, Transition, to predict the transition from novice to expert interaction techniques. One key aspect of our approach was to model the whole learning process of expert technique adoption, i.e to explain whether, when and how users make the transition. Another key aspect was to rely on the Reinforcement Learning framework appropriate to address learning and decision-making problems, where we considered three high-level strategies (Menu; Shortcut and Learning) as actions.

### 12.2 Empirical evaluation of the model

The Transition model has been empirically evaluated on the Grossman et al. database [44]. A key contribution of our work is the variety of approaches used to evaluate our model, increase transparency and avoid potential evaluation biases. First, despite the lack of dedicated models of shortcut adoption in HCI, we compared our model to three benchmark models in neuroscience (RW, CK and RWCK), which are widely used in decision-making tasks involving a learning process [101]. Second, we analyzed our model both in terms of goodness-of-fit and simulation. These two methods have been shown to be complementary as they can lead to different conclusions [80, 101]. Third, we analyzed our model at four levels of granularity [9]: Overall, interaction method, participant and sequence of actions. This is important to avoid the risk to over-interpret aggregated data. Fourth, we compared our model to variant models where each of the five involved mechanisms have been alternatively disabled to ensure all mechanisms are useful.

Altogether our results show that the Transition model outperforms the three benchmark RL models and the five mechanisms contribute to shortcut adoption.

### 12.3 Analytical evaluation of the model

We now critically discuss our model in light of the criteria of Jacobs and Grainger [50]:

*12.3.1 Plausibility and Explanation.* Transition is well grounded in the cognitive science and neuroscience literatures, both in terms of problem formulation and model design.

First, our *problem formulation* is in line with the computational view of human behavior [39, 66] where the users' strategies (policy) emerge from the user's goal (utility), their cognitive mechanisms and the task environment. Moreover, we acknowledge a hierarchical nature of decision-making [56], where we focus on the higher level of decision-making: users choose a strategy among Menu, Shortcut and Learning.

One promising direction for future work is to investigate alternative hierarchies of high-level decisions. For instance, some users might decide first whether they use shortcut or not. If not, they then decide whether they use the Menu or Learning strategy. However, it is not clear whether this would produce significantly different results than the present model. It would also be interesting to refine the action "Learning" to capture the degree of explicit learning, e.g. the time spent to learn the shortcut. Another direction is to adopt a mechanistic approach and to refine low-level decisions, i.e. how users select items in a menu or execute shortcuts. This will be important when focusing on execution time and error rate. Indeed, one limitation of our approach is that it does not cover intramodal performance improvement [26]. Our model currently assumes that execution time does not evolves over time and only depends on the used strategy. We plan to introduce some mechanisms such as visual search and pointing (Fitts' law) from existing models of menu performance [9, 21], and include speed of recall from memory [2, 96] to reflect how users behave *within* a menu as well as the effect of practice on execution time. We also plan to understand the nature of errors when using shortcuts. Currently our model over-estimates the number of errors. One direction would be to introduce a component for risk aversion as users might value more the certainty of correctly executing a command with menus than the uncertainty of the benefits of shortcuts [82].

Second, we designed the Transition model so that it combines five mechanisms: implicit and explicit learning, decay, planning and preservation. These five mechanisms are grounded in the cognitive science and neuroscience literatures. The comparison of Transition with some variants enabling/disabling each mechanism suggests that these five mechanisms play a role in the transition observed in participants.

Among these mechanisms, perseveration should require further investigation as its role is less clear. It has been demonstrated that perseveration is frequent in human choice behavior [101], but we have observed here differences between model fitting and simulation data. One possible reason is that our degree of perserveration is currently static. An alternative would be to introduces the choice kernel of the CK and RWCK models. This might provide more realistic simulations. Another direction is to add a model-free component. Indeed, several approaches in neuroscience and cognitive sciences combine model-free and model-based RL models. While this approach is more complex, it might help better describe users' behaviors [96]. Finally, it would also be interesting to study whether the use of shortcuts acquires the properties of behavioral habits after hours of practice, because computational neuroscience studies have pointed to a role of model-free RL mechanisms in the progressive acquisition of behavioral habits [29, 54, 96].

Several additional mechanisms could also be considered as future work. Among them, a promising direction is the transfer of learning between commands. Our model assumes that the evolution of the knowledge for a given command is *independent* of the other commands. We would like to investigate whether the successful adoption of a shortcut (for a given command) has an impact on the transition for the other commands. Future work should also investigate the ability of users to estimate command frequency and to include this estimate within the planning process.

*12.3.2 Descriptive adequacy.* Our model provides a good description of the observed data in comparison with the tested benchmark RL models both in terms of model fitting and model simulation. In absolute terms, our fitting scores might appear low. However, this is often the case when modeling decision-making problems due to the complexity of the task [101]. Moreover, we used state-of-the art model fitting methods that do not favor high fitting score but better reflect the adequacy with human behavior [101]. Indeed, we predicted trial-by-trial actions for each participant, i.e. we predicted more than 700 decisions per participant with a high variability within and between participants.

*12.3.3 Interpretability and Complexity.* Our predictive HCI model does not rely on "black box" machine learning such as deep learning [67]. Each parameter is associated to psychological mechanisms. Moreover, our approach shares some similarities with cognitive models (e.g. ACT-R), but is less complex as it relies on a well-established RL framework and has a limited number of parameters per participant. Finally, the model is easy to implement and test, i.e it does not require running millions of simulations such as regular RL models in HCI (e.g. [13, 21, 22, 38]). Considering the complexity of the task to predict in comparison with common HCI motor control tasks (e.g. Fitts law), we argue that our model has a low complexity.

*12.3.4 Generalizability.* Further work should investigate whether the model can characterize and predict the users' behavior with different interaction methods, modalities, populations or tasks. An analytical examination of our model already provides some hints. For instance, some interaction methods penalize menu selection time (e.g. HotKeyCoach [58]) or reduce the temporal cost of the Learning strategy (e.g. ExposeHK [71], KeyCue [92]). The equations 8, 9 and 10 inform that these strategies (i.e. increasing $T_M$ or reducing $T_L$) reduce the Q-Value of Menu in comparison with the two other strategies and thus favor shortcut adoption (Equation 2). However, several interaction methods are more complex to model such as those considered in this article. Indeed, their differences can not easily be represented with quantitative values and were represented as a nominal scale. Our long-term goal is thus to demonstrate our model can rely on a unique set of parameters *independent* of the interaction methods, i.e. the interaction methods are represented as a small set of variables, in order to test if the model gives plausible predictions when the techniques changed. We also plan to test whether the model can characterize the transition from menus to gesture shortcuts. This would probably require to consider additional types of decision (e.g. decision about the mapping between the command and the shortcut) to reflect the fact that gestures are generally easier to learn and recall than keyboard shortcuts [4].

### 12.4 Command selection and computational models

Beyond this work, this manuscript is also a call for computational models of command selection and in particular the transition from novice to expert interaction techniques. We argue that command selection is an important proxy to study HCI [7]. One whole interface can be too difficult to model because it involves so many different users' behaviors.

In contrast, a simple pointing task (which is already quite complex) hides fundamental aspects such as those related to learning or decision-making. Command selection appears to have the right level of complexity and thus especially appropriate for computational modeling. The main interactive components of command selection (menus, gestures, keyboard shortcuts, etc.) are quite well defined but involve many fascinating and challenging phenomena related to pointing, visual search, skill acquisition and decision-making, in particular, when considering the transition from novice (e.g. menus) to expert interaction techniques (e.g. shortcuts). However, we were not aware of a computational model to explain or predict how users switch from menus to shortcuts. This is surprising given the number of models of menu performance, e.g. [9, 21]. We believe that one of the reasons is that this transition involves a subtle interaction between learning and decision-making which is difficult to disentangle. This work contributes the first step in this direction and should encourage other researchers to investigate this challenging and fundamental HCI problem.

### 12.5 Neuroscience and Human-Computer Interaction

Neuroscience influences many fields such as economics, psychology, social sciences, marketing or information systems [86]. Recently, several authors also mentioned the potential of Neuroscience for HCI [75, 86]. In particular, in terms of *empirical* methods and tools to study interaction design [98]. For instance, by using fMRI, PET, EEG or GSR techniques to measure the effect of artifacts on the cognitive state (e.g. cognitive effects) of individual users or to identify cognitive conflicts in specific brain regions. In this article, we demonstrate the potential of importing approaches, models and evaluation methods from Neuroscience to HCI from a *theoretical* perspective.

First, neuroscience is strongly anchored in computational rationality [39], an emerging *approach* in HCI [21, 66]. Both fields address problems related to learning, decision-making or emotions with concepts of utility and reward through the Reinforcement Learning (RL) framework. However, neuroscience approaches can be beneficial to HCI. For instance, previous RL-based HCI models generally adopt a "machine learning" perspective of RL where the evolution of the Q-values does not have meanings (see section 2.2.4). In contrast, the dynamic of the Q-values is of importance in Neuroscience and reflects how the human or animal learns [101].

Second, several *models* have been proposed to study human behavior in Neuroscience (relying on the computational Rationality approach). We considered three of them: Rescorla-Wagner, Choice Kernel and their combination. However, more advanced models should be considered and transposed to HCI problems. In particular, an emergent class of models combining model-free and model-based RL approaches have been proved efficient to explain complex human behaviors. We plan to investigate such models, e.g. [96] in the context of the transition from novice to expert interaction techniques.

Third, Neuroscience has well-established *methods* to evaluate models of human behavior which are not common practice in HCI. For instance, it is common in HCI to consider population models (the same parameters for each participant), while we considered individual models (each participant has a different set of parameters) which is more appropriate when studying decision making [101] (see section 6.1.2). Moreover, our fitness function considers trial-by-trial actions rather than aggregated measures. While computationally more expensive, this better reflects users' behaviors. We also combined goodness-of-fit and simulation and performed post-analysis enabling/disabling mechanisms in order to increase the transparency of our results, which constitute gold standard nowadays in computational neuroscience [101]. We believe that these methods and others such as Model recovering can increase the validity, robustness and transparency of HCI computational models.

## 13 OPEN SCIENCE

We support adoption and further research efforts by providing an open code repository, with examples and instructions, on our project page: https://hci.isir.upmc.fr/project/model-of-transition/.

## 14 ACKNOWLEDGMENTS

We would like to thank Baptiste Caramiaux and Sylvain Malacria for their feedback on the early draft of this manuscript as well as the discussions with Olivier Sigaud, Antti Oulasvirta and Andrew Howes. This work was funded by Agence Nationale de la Recherche (grant number ANR-16-CE33-0023).

## REFERENCES

[1] Carlos Alós-Ferrer, Sabine Hügelschäfer, and Jiahui Li. 2016. Inertia and decision making. *Frontiers in psychology* 7 (2016), 169.

[2] John R Anderson. 1982. Acquisition of cognitive skill. *Psychological review* 89, 4 (1982), 369.

[3] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An integrated theory of the mind. *Psychological review* 111, 4 (2004), 1036.

[4] Caroline Appert and Shumin Zhai. 2009. Using Strokes As Command Shortcuts: Cognitive Benefits and Toolkit Support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. ACM, New York, NY, USA, 2289–2298. https://doi.org/10.1145/1518701.1519052

[5] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[6] Gilles Bailly, Emmanouil Giannisakis, Marion Morel, and Catherine Achard. 2018. Characterize the Transition from Menus to Hotkeys. In *Proceedings of the 30th Conference on l'Interaction Homme-Machine* (Brest, France) *(IHM '18)*. Association for Computing Machinery, New York, NY, USA, 30–41. https://doi.org/10.1145/3286689.3286699

[7] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *ACM Comput. Surv.* 49, 4, Article 60 (Dec. 2016), 41 pages. https://doi.org/10.1145/3002171

[8] Gilles Bailly, Jörg Müller, and Eric Lecolinet. 2012. Design and evaluation of finger-count interaction: Combining multitouch gestures and menus. *International Journal of Human-Computer Studies* 70, 10 (2012), 673–689. https://doi.org/10.1016/j.ijhcs.2012.05.006 Special issue on Developing, Evaluating and Deploying Multi-touch Systems.

[9] Gilles Bailly, Antti Oulasvirta, Duncan P. Brumby, and Andrew Howes. 2014. Model of Visual Search and Selection Time in Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, New York, NY, USA, 3865–3874. https://doi.org/10.1145/2556288.2557093

[10] 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* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 331–342. https://doi.org/10.1145/2501988.2502024

[11] Gilles Bailly, Thomas Pietrzak, Jonathan Deber, and Daniel J. Wigdor. 2013. Métamorphe: Augmenting Hotkey Usage with Actuated Keys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 563–572. https://doi.org/10.1145/2470654.2470734

[12] Robert W Baloh, Andrew W Sills, Warren E Kumley, and Vicente Honrubia. 1975. Quantitative measurement of saccade amplitude, duration, and velocity. *Neurology* 25, 11 (1975), 1065–1065.

[13] Nikola Banovic, Tofi Buzali, Fanny Chevalier, Jennifer Mankoff, and Anind K. Dey. 2016. Modeling and Understanding Human Routine Behavior. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 248–260. https://doi.org/10.1145/2858036.2858557

[14] Marc G Berman, John Jonides, and Richard L Lewis. 2009. In search of decay in verbal short-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 35, 2 (2009), 317.

[15] Michael D Byrne. 2001. ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies* 55, 1 (2001), 41–84.

[16] Xiang Cao and Shumin Zhai. 2007. Modeling Human Performance of Pen Stroke Gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1495–1504. https://doi.org/10.1145/1240624.1240850

[17] Stuart K. Card, Allen Newell, and Thomas P. Moran. 1983. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., USA.

[18] John M. Carroll (Ed.). 1987. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. MIT Press, Cambridge, MA, USA.

[19] Romain Cazé, Mehdi Khamassi, Lise Aubin, and Benoît Girard. 2018. Hippocampal replays under the scrutiny of reinforcement learning models. *Journal of neurophysiology* 120, 6 (2018), 2877–2896.

[20] Noshaba Cheema, Laura A. Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. 2020. Predicting Mid-Air Interaction Movements and Fatigue Using Deep Reinforcement Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376701

[21] Xiuli Chen, Gilles Bailly, Duncan P. Brumby, Antti Oulasvirta, and Andrew Howes. 2015. The Emergence of Interactive Behavior: A Model of Rational Menu Search. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 4217–4226. https://doi.org/10.1145/2702123.2702483

[22] Xiuli Chen, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. 2017. A Cognitive Model of How People Make Decisions Through Interaction with Visual Displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1205–1216. https://doi.org/10.1145/3025453.3025596

[23] François Cinotti, Virginie Fresno, Nassim Aklil, Etienne Coutureau, Benoît Girard, Alain R Marchand, and Mehdi Khamassi. 2019. Dopamine blockade impairs the exploration-exploitation trade-off in rats. *Scientific reports* 9, 1 (2019), 1–14.

[24] Andy Cockburn and Carl Gutwin. 2009. A predictive model of human performance with scrolling and hierarchical lists. *Human–Computer Interaction* 24, 3 (2009), 273–314.

[25] Andy Cockburn, Carl Gutwin, and Saul Greenberg. 2007. A Predictive Model of Menu Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. ACM, New York, NY, USA, 627–636. https://doi.org/10.1145/1240624.1240723

[26] 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. https://doi.org/10.1145/2659796

[27] Nathaniel D. Daw. 2011. *Trial-by-trial data analysis using computational models*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199600434.003.0001 Publisher Copyright: © The International Association for the study of Attention and Performance, 2011. All rights reserved.

[28] Nathaniel D. Daw. 2013. *Advanced Reinforcement Learning*. Elsevier Inc., 299–320. https://doi.org/10.1016/B978-0-12-416008-8.00016-4 Copyright: Copyright 2021 Elsevier B.V., All rights reserved.

[29] Nathaniel D Daw, Yael Niv, and Peter Dayan. 2005. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience* 8, 12 (2005), 1704–1711.

[30] Peter Dayan and Nathaniel D Daw. 2008. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience* 8, 4 (2008), 429–453.

[31] Laurent Dollé, Denis Sheynikhovich, Benoît Girard, Ricardo Chavarriaga, and Agnès Guillot. 2010. Path planning versus cue responding: a bio-inspired model of switching between navigation strategies. *Biological cybernetics* 103, 4 (2010), 299–317.

[32] Kenji Doya. 2008. Modulators of decision making. *Nature neuroscience* 11, 4 (2008), 410–416.

[33] R'emi Dromnelle, B. Girard, Erwan Renaudo, R. Chatila, and M. Khamassi. 2020. Coping with the variability in humans reward during simulated human-robot interactions through the coordination of multiple learning strategies*. *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (2020), 612–617.

[34] Rémi Dromnelle, Erwan Renaudo, Guillaume Pourcel, Raja Chatila, Benoît Girard, and Mehdi Khamassi. 2020. How to Reduce Computation Time While Sparing Performance During Robot Navigation? A Neuro-Inspired Architecture for Autonomous Shifting Between Model-Based and Model-Free Learning. In *Biomimetic and Biohybrid Systems*, Vasiliki Vouloutsi, Anna Mura, Falk Tauber, Thomas Speck, Tony J. Prescott, and Paul F. M. J. Verschure (Eds.). Springer International Publishing, Cham, 68–79.

[35] Paul Morris Fitts and Michael I Posner. 1967. *Human performance*. Brooks/Cole.

[36] Wai-Tat Fu and Wayne D. Gray. 2004. Resolving the paradox of the active user: stable suboptimal performance in interactive tasks. *Cognitive Science* 28, 6 (2004), 901–935. https://doi.org/10.1207/s15516709cog2806_2

[37] Wai-Tat Fu and Peter Pirolli. 2007. SNIF-ACT: A Cognitive Model of User Navigation on the World Wide Web. *Human–Computer Interaction* 22, 4 (2007), 355–412. https://doi.org/10.1080/07370020701638806 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/07370020701638806

[38] Christoph Gebhardt, Antti Oulasvirta, and Otmar Hilliges. 2020. Hierarchical Reinforcement Learning Explains Task Interleaving Behavior. *Computational Brain & Behavior* (2020), 1–21. https://link.springer.com/article/10.1007/s42113-020-00093-9

[39] Samuel J Gershman. 2020. Origin of perseveration in the trade-off between reward and complexity. *Cognition* 204 (2020), 104394.

[40] Emmanouil Giannisakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. 2017. IconHK: Using Toolbar Button Icons to Communicate Keyboard Shortcuts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 4715–4726. https://doi.org/10.1145/3025453.3025595

[41] Wayne D Gray and Deborah A Boehm-Davis. 2000. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied* 6, 4 (2000), 322.

[42] Wayne D. Gray and John K. Lindstedt. 2016. Plateaus, Dips, and Leaps: Where to Look for Inventions and Discoveries During Skilled Performance. *Cognitive Science* (2016), n/a–n/a. https://doi.org/10.1111/cogs.12412

[43] Wayne D Gray, Chris R Sims, Wai-Tat Fu, and Michael J Schoelles. 2006. The soft constraints hypothesis: a rational analysis approach to resource allocation for interactive behavior. *Psychological review* 113, 3 (2006), 461.

[44] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. 2007. Strategies for Accelerating On-line Learning of Hotkeys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. ACM, New York, NY, USA, 1591–1600. https://doi.org/10.1145/1240624.1240865

[45] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster Command Selection on Tablets with FastTap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2617–2626. https://doi.org/10.1145/2556288.2557136

[46] Masahiko Haruno and Mitsuo Kawato. 2006. Heterarchical reinforcement-learning model for integration of multiple cortico-striatal loops: fMRI examination in stimulus-action-reward association learning. *Neural networks* 19, 8 (2006), 1242–1254.

[47] Anthony J. Hornof and David E. Kieras. 1997. Cognitive Modeling Reveals Menu Search in Both Random and Systematic. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '97)*. Association for Computing Machinery, New York, NY, USA, 107–114. https://doi.org/10.1145/258549.258621

[48] R. A. Howard. 1960. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA.

[49] Poika Isokoski. 2001. Model for Unistroke Writing Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '01)*. Association for Computing Machinery, New York, NY, USA, 357–364. https://doi.org/10.1145/365024.365299

[50] Arthur M Jacobs and Jonathan Grainger. 1994. Models of visual word recognition: sampling the state of the art. *Journal of Experimental Psychology: Human perception and performance* 20, 6 (1994), 1311.

[51] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4 (1996), 237–285.

[52] Antti Kangasrääsiö, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Samuel Kaski, and Antti Oulasvirta. 2017. Inferring Cognitive Models from Data Using Approximate Bayesian Computation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1295–1306. https://doi.org/10.1145/3025453.3025576

[53] Mehdi Keramati, Amir Dezfouli, and Payam Piray. 2011. Speed/accuracy trade-off between the habitual and the goal-directed processes. *PLoS Comput Biol* 7, 5 (2011), e1002055.

[54] Mehdi Khamassi and Mark D Humphries. 2012. Integrating cortico-limbic-basal ganglia architectures for learning model-based and model-free navigation strategies. *Frontiers in behavioral neuroscience* 6 (2012), 79.

[55] Michel C. A. Klein, Nataliya Mogles, Jan Treur, and Arlette van Wissen. 2011. A Computational Model of Habit Learning to Enable Ambient Support for Lifestyle Change. In *Modern Approaches in Applied Intelligence*, Kishan G. Mehrotra, Chilukuri K. Mohan, Jae C. Oh, Pramod K. Varshney, and Moonis Ali (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 130–142.

[56] Etienne Koechlin and Christopher Summerfield. 2007. An information theoretical approach to prefrontal executive function. *Trends in cognitive sciences* 11, 6 (2007), 229–235.

[57] Nils Kolling, Marco K Wittmann, Tim EJ Behrens, Erie D Boorman, Rogier B Mars, and Matthew FS Rushworth. 2016. Value, search, persistence and model updating in anterior cingulate cortex. *Nature neuroscience* 19, 10 (2016), 1280–1285.

[58] Brian Krisler and Richard Alterman. 2008. Training Towards Mastery: Overcoming the Active User Paradox. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges* (Lund, Sweden) *(NordiCHI '08)*. ACM, New York, NY, USA, 239–248. https://doi.org/10.1145/1463160.1463186

[59] Gordon Paul Kurtenbach. 1993. *The design and evaluation of marking menus*. Ph.D. Dissertation. University of Toronto.

[60] David M Lane, H Albert Napier, S Camille Peres, and Aniko Sandor. 2005. Hidden Costs of Graphical User Interfaces: Failure to Make the Transition from Menus and Icon Toolbars to Keyboard Shortcuts. *International Journal of Human-Computer Interaction* 18, 2 (may 2005), 133–144. https://doi.org/10.1207/s15327590ijhc1802_1

[61] Eric Lee and James MacGregor. 1985. Minimizing user search time in menu retrieval systems. *Human Factors* 27, 2 (1985), 157–162.

[62] Katri Leino, Antti Oulasvirta, and Mikko Kurimo. 2019. RL-KLM: Automating Keystroke-Level Modeling with Reinforcement Learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 476–480. https://doi.org/10.1145/3301275.3302285

[63] Luis A. Leiva, Daniel Martín-Albo, Réjean Plamondon, and Radu-Daniel Vatavu. 2018. KeyTime: Super-Accurate Prediction of Stroke Gesture Production Times. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173813

[64] Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1998. Using Markov decision process for learning dialogue strategies. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, Vol. 1. IEEE, 201–204.

[65] Blaine Lewis, Greg d'Eon, Andy Cockburn, and Daniel Vogel. 2020. KeyMap: Improving Keyboard Shortcut Vocabulary Using Norman's Mapping. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376483

[66] Richard L Lewis, Andrew Howes, and Satinder Singh. 2014. Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in cognitive science* 6, 2 (2014), 279–311.

[67] Yang Li, Samy Bengio, and Gilles Bailly. 2018. Predicting Human Performance in Vertical Menu Selection Using Deep Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3173574.3173603

[68] Wanyu Liu, Gilles Bailly, and Andrew Howes. 2017. Effects of frequency distribution on linear menu performance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1307–1312.

[69] Abraham S Luchins. 1942. Mechanization in problem solving: The effect of Einstellung. *Psychological monographs* 54, 6 (1942), i.

[70] Alan Lundgard, Yiwei Yang, Maya L. Foster, and Walter S. Lasecki. 2018. Bolt: Instantaneous Crowdsourcing via Just-in-Time Training. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3173574.3174041

[71] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. 2013. Promoting Hotkey Use Through Rehearsal with ExposeHK. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. ACM, New York, NY, USA, 573–582. https://doi.org/10.1145/2470654.2470735

[72] Sylvain Malacria, Joey Scarr, Andy Cockburn, Carl Gutwin, and Tovi Grossman. 2013. Skillometers: Reflective Widgets That Motivate and Help Users to Improve Performance. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. ACM, New York, NY, USA, 321–330. https://doi.org/10.1145/2501988.2501996

[73] Brendon Matusch, Jimmy Ba, and Danijar Hafner. 2021. Evaluating Agents without Rewards. arXiv:2012.11538 [cs.LG]

[74] Kevin J Miller, Amitai Shenhav, and Elliot A Ludvig. 2019. Habits without values. *Psychological review* 126, 2 (2019), 292.

[75] Brad S. Minnery and Michael S. Fine. 2009. FEATURE Neuroscience and the Future of Human-Computer Interaction. *Interactions* 16, 2 (March 2009), 70–75. https://doi.org/10.1145/1487632.1487649

[76] A. Newell and P. S. Rosenbloom. 1993. *Mechanisms of Skill Acquisition and the Law of Practice.* MIT Press, Cambridge, MA, USA, 81–135.

[77] Daniel L Odell, Richard C Davis, Andrew Smith, and Paul K Wright. 2004. Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. *Proceedings of Graphics Interface - GI '04* (2004), 17–24. https://doi.org/10.20380/GI2004.03

[78] John P O'Doherty, Jeffrey Cockburn, and Wolfgang M Pauli. 2017. Learning, reward, and decision making. *Annual review of psychology* 68 (2017), 73–100.

[79] Richard C Omanson, Craig S Miller, Elizabeth Young, and David Schwantes. 2010. Comparison of Mouse and Keyboard Efficiency Effects of Practice. (2010), 600–604.

[80] Stefano Palminteri, Valentin Wyart, and Etienne Koechlin. 2017. The importance of falsification in computational cognitive modeling. *Trends in cognitive sciences* 21, 6 (2017), 425–433.

[81] S. Camille Peres, II Franklin P. Tamborello, II Michael D. Fleetwood, II Phillip Chung, and II Danielle L. Paige-Smith. 2004. Keyboard Shortcut Usage: The Roles of Social Factors and Computer Experience. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 48, 5 (2004), 803–807. https://doi.org/10.1177/154193120404800513 arXiv:https://doi.org/10.1177/154193120404800513

[82] Philip Quinn and Andy Cockburn. 2020. Loss Aversion and Preferences in Interaction. *Human–Computer Interaction* 35, 2 (2020), 143–190. https://doi.org/10.1080/07370024.2018.1433040 arXiv:https://doi.org/10.1080/07370024.2018.1433040

[83] Philip Quinn and Shumin Zhai. 2018. Modeling Gesture-Typing Movements. *Human–Computer Interaction* 33, 3 (2018), 234–280. https://doi.org/10.1080/07370024.2016.1215922 arXiv:https://doi.org/10.1080/07370024.2016.1215922

[84] Adrian E Raftery. 1995. Bayesian model selection in social research. *Sociological methodology* (1995), 111–163.

[85] Roger W Remington, Ho Wang Holman Yuen, and Harold Pashler. 2016. With practice, keyboard shortcuts become faster than menu selection: A crossover interaction. *Journal of Experimental Psychology: Applied* 22, 1 (2016), 95–106. https://doi.org/10.1037/xap0000069

[86] René Riedl, Adriane B Randolph, Jan vom Brocke, Pierre-Majorique Léger, and Angelika Dimoka. 2010. The potential of neuroscience for human-computer interaction research. *SIGHCI 2010 Proceedings* (2010).

[87] Joey Scarr, Andy Cockburn, Carl Gutwin, and Philip Quinn. 2011. Dips and Ceilings: Understanding and Supporting Transitions to Expertise in User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2741–2750. https://doi.org/10.1145/1978942.1979348

[88] Noah A Shamosh, Colin G DeYoung, Adam E Green, Deidre L Reis, Matthew R Johnson, Andrew RA Conway, Randall W Engle, Todd S Braver, and Jeremy R Gray. 2008. Individual differences in delay discounting: relation to intelligence, working memory, and anterior prefrontal cortex. *Psychological science* 19, 9 (2008), 904–911.

[89] Catherine Sibert, Wayne D Gray, and John K Lindstedt. 2017. Interrogating feature learning models to discover insights into the development of human expertise in a real-time, dynamic decision-making task. *Topics in cognitive science* 9, 2 (2017), 374–394.

[90] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction.* Vol. 1. MIT press Cambridge.

[91] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.

[92] Susanne Tak, Piet Westendorp, and Iris van Rooij. 2013. Satisficing and the Use of Keyboard Shortcuts: Being Good Enough Is Enough? *Interacting with computers* 25, 5 (2013), 404–416.

[93] Robert Tobias. 2009. Changing behavior by memory aids: A social psychological model of prospective memory and habit development tested with dynamic field data. *Psychological review* 116, 2 (2009), 408.

[94] Kashyap Todi, Gilles Bailly, Luis A Leiva, and Antti Oulasvirta. 2021. Adapting user interfaces with model-based reinforcement learning. *arXiv preprint arXiv:2103.06807* (2021).

[95] Pramod Verma. 2013. Gracoli: A Graphical Command Line User Interface. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (Paris, France) *(CHI EA '13)*. Association for Computing Machinery, New York, NY, USA, 3143–3146. https://doi.org/10.1145/2468356.2479631

[96] Guillaume Viejo, Mehdi Khamassi, Andrea Brovelli, and Benoît Girard. 2015. Modeling choice and reaction time during arbitrary visuomotor learning through the coordination of adaptive working memory and reinforcement learning. *Frontiers in behavioral neuroscience* 9 (2015), 225.

[97] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* 17, 3 (2020), 261–272.

[98] Jan vom Brocke, René Riedl, and Pierre-Majorique Léger. 2011. Neuroscience in Design-Oriented Research: Exploring New Potentials. In *Service-Oriented Perspectives in Design Science Research*, Hemant Jain, Atish P. Sinha, and Padmal Vitharana (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 427–439.

[99] 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.

[100] Mark E Walton, Timothy EJ Behrens, Mark J Buckley, Peter H Rudebeck, and Matthew FS Rushworth. 2010. Separable learning systems in the macaque brain and the role of orbitofrontal cortex in contingent learning. *Neuron* 65, 6 (2010), 927–939.

[101] Robert C Wilson and Anne GE Collins. 2019. Ten simple rules for the computational modeling of behavioral data. *Elife* 8 (2019), e49547.

[102] Windows-Sdk-Content. [n.d.]. Keyboard - Windows applications. https://docs.microsoft.com/en-us/windows/desktop/uxguide/inter-keyboard

[103] Jingjie Zheng and Daniel Vogel. 2016. Finger-Aware Shortcuts. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 4274–4285. https://doi.org/10.1145/2858036.2858355