UNIVERSITY OF GRONINGEN

THESIS HUMAN-MACHINE COMMUNICATION

Bilinguals Already Know: Reducing Internal Control Leads to Faster Task-Switching Performance.

Author:
Charlotte DE BLECOURT
(s2188813)

Supervisors:
dr. Jelmer BORST
dr. Andrea STOCCO

November 2, 2018

Contents

Abstract

Rapid Instructed Task Learning (RITL) involves rapid switching between instructions and stimuli. In a previous RITL experiment by Stocco and Prat (2014), bilingual and monolingual participants solved sets of arithmetic operations. Monolinguals executed novel sets $\pm$500 ms slower than practiced sets, while bilinguals executed both novel and practiced sets at the same pace. In other words, bilinguals may not have changed their execution strategy; they may already know how to efficiently switch between task instructions and information. Bilingualism increases an individual's linguistic demands. Bilinguals may therefore reduce internal control in exchange for faster information routing, resulting in more fluent language use at the expense of accuracy. This effect might spill over in non-linguistic task-switching, thus resulting in lower reaction times in task-switching paradigms. This hypothesis is in line with Abutalebi and Green's adaptive control hypothesis and Stocco et al.'s conditional routing model. To investigate this hypothesis, we looked at the computational steps that bilinguals and monolinguals take during a RITL trial. We created computational models of monolinguals and bilinguals for the RITL paradigm as seen in Stocco and Prat (2014), using the Adaptive Control of Thought–Rational (ACT–R) architecture. This allows us to observe RITL task performance as an adaptive step-by-step process. The "monolingual" model performs each step in the task separately. With experience, these steps merge into habitual sequences of actions, in which multiple computations are performed at once. The "bilingual" model has these compiled sequences by design, and internal control checks have been removed. This reduces execution time and eliminates learning effects. The models correctly reproduce the behavioral pattern observed in Stocco and Prat over a wide parameter space. These results suggest that the additional linguistic demands of bilingualism are met by reducing internal control in exchange for faster information routing between cortical areas, and that this effect carries over to non-linguistic rule-switching as well.

Keywords: ACT-R; Rapid Instructed Task Learning; Bilingualism

1 Introduction

Over 86% of the Dutch (Eurostat, 2018) and about a quarter of the American population (United States Census Bureau, 2013) speak at least one other language besides their native tongue. Increasing evidence suggests that multilingualism has additional benefits besides the ability to communicate outside one's native language. Bilinguals tend to outperform monolinguals in cognitive control tasks (e.g. Bialystok, Craik, Klein, and Viswanathan (2004) and Bialystok (2009)), especially in task-switching paradigms that are designed to measure executive function, such as the Simon task (Bialystok et al., 2004; Prior & MacWhinney, 2010).

Executive function is an overarching name for activities that involve inhibition, set shifting, and updating (see, for example, Miyake et al. (2000)). Bilingualism seems to train the latter two (Prior & MacWhinney, 2010; Carlson & Meltzoff, 2008). As both languages are always active and competing (MacWhinney, 1997), bilinguals constantly need to control the flow of information. This "training" improves the efficiency of task-switching, which is in turn observed as faster reaction times in task-switching paradigms (Prior & MacWhinney, 2010). This supports the idea that task-switching is less taxing for bilinguals (Stocco, Yamasaki, Natalenko, & Prat, 2014). Moreover, neuroimaging findings suggest that the bilingual experience trains specific brain circuits involved in flexible rule selection and application (Stocco & Prat, 2014; Stocco et al., 2014), which are an important aspect of executive function. The bilingual experience may therefore result in computational advantages that spill over in non-linguistic tasks. However, although the behavioral and neural effects have been studied, the mechanisms behind the computational advantages of bilingualism are unknown.

In the present paper, I will explain the computational advantage of bilingualism as exchanging internal control for faster information routing. First, I will explain the current theories behind the bilingual advantages in task-switching. Next, I will explain the advantage of using the Rapid Instructed Task Learning (RITL) paradigm to investigate task-switching behavior. We will test the difference in task-switching behavior in a RITL paradigm for bilinguals and monolinguals using two computational models created with the Adaptive Control of Thought–Rational (ACT–R) architecture. The models differ in two ways. The "monolingual" model performs ACT-R buffer manipulations separately and checks buffer availability before executing the task. The model is able to merge computational steps (known as *production rules*), and removes buffer queries in the process. The "bilingual" model already has these merged production rules and has no queries by design. These adaptions reduce response times, but may result in a lower accuracy.

1.1 Current Models of Bilingual Task-Switching

The current theories behind bilingual advantages in task-switching will be outlined in this section. The *conditional routing model* (Stocco, Lebiere, & Anderson, 2010; Stocco et al., 2014) suggests that signals usually travel over the cortex. When the regular cortico-cortical route cannot sufficiently keep up with external demands - our case, linguistic demands - , an alternative neural route will be sought through the basal ganglia. The other theory, Green and Abutalebi's cognitive control hypothesis (Green & Abutalebi, 2013), states that language control processes themselves adapt to demands from the environment. Note that these two theories do not contradict each other: they complement each other in explaining the external and internal circumstances that lead to faster task-switching in bilinguals.

1.1.1 The conditional routing model

According to the conditional routing model (Stocco et al., 2010, 2014), the basal ganglia re-route signals between cortical regions. Without basal ganglia intervention, the flow of signals across cortical networks is determined by the strength of cortico-cortical projections, which are shaped by practice. The basal ganglia actively shape behavior by prioritizing different signals and overriding preexisting cortico-cortical connections. With practice, intermediate steps through the basal ganglia are eliminated. This learning process results in a stable trade-off between task speed and automaticity.

In language, the conditional routing model comes in to play when linguistic rules increase in complexity (Stocco et al., 2014; Seo, Stocco, & Prat, 2018). For example, besides the regular conjugation of a verb, the English language has numerous irregular verbs. This challenges the established cortico-cortical pathways, because they require flexible activation of the right pathways under the right conditions. The rules and requirements get permanently stored with enough practice. The conditional routing model also connects the relationship between language and executive function: although they are two different aspects of cognitive functioning, they rely on the same set of computations through the basal ganglia circuitry.

Bilingualism provides an individual with a whole new set of rules that are being used separately from the other language. This increases demands on the basal ganglia to select appropriate rules and representations, and to switch between rules and representations depending on the intended language. Thus, second language proficiency may increase the basal ganglia's ability to exert control over established cortico-cortical connections as a side effect. Findings indeed suggest that bilinguals and monolinguals differ in re-routing when task demands are increasing. For example, bilinguals are more effectively modulating basal ganglia activity when

faced with changing task demands (Stocco & Prat, 2014) (see Figure 1, left): basal ganglia activity in bilinguals increases more in novel trials, while familiar trials do not affect BG activity. The basal ganglia do not show a significant difference in activity in monolinguals.

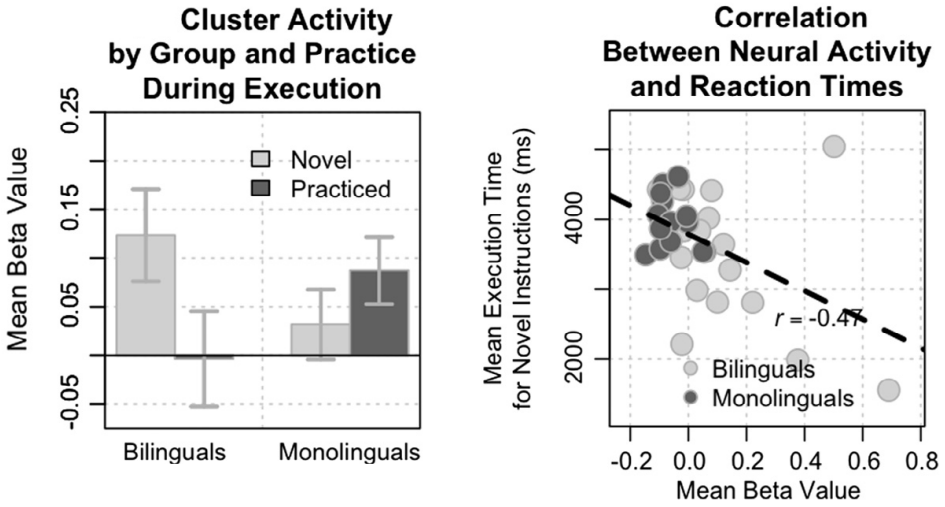

Figure 1: Left: Mean beta values (±SEM) during the Execution phase of the behavioral experiment in the left thalamus, left pallidus, and left putamen, divided by group (Bilinguals vs. Monolinguals) and practice (Novel vs. Practiced instructions). Right: Scatterplot illustrating the correlation between Rapid Instructed Task Learning execution times and activation difference (Novel – Practiced trials) in the left putamen cluster of the basal ganglia. Retrieved from Stocco & Prat (2014).

1.1.2 The adaptive control hypothesis

The adaptive control hypothesis (Green & Abutalebi, 2013) takes another approach by asserting that the demands of language control processes in bilingual speakers are higher than in monolingual speakers. Second language usage highly depends on context. For example, one might use two different languages at work (*dual language context*), or one language at work and one at home (*single language context*). As the context of the language usage changes, the demands for control of cognitive processes change along too. Each interactional context affects control processes in a different way. In particular, goal maintenance, interference control, response inhibition, and task (dis-)engagement are affected in single and/or dual language contexts (Abutalebi & Green, 2007; Green & Abutalebi, 2013).

The language control processes are tied to a wide network of brain regions (Green & Abutalebi, 2013; Abutalebi & Green, 2016). For the adaptive control hypothesis, these regions are divided into the *speech pipeline* and the *control network*. The basal ganglia are placed as a "gate" that control the information flow between the prefrontal cortex and posterior cortical regions (Crinion et al., 2006; Abutalebi & Green, 2016). The dorso-lateral pre-frontal cortex (DLPFC), anterior cingulate cortex (ACC), and the presupplementary motor area (pre-SMA) are particularly important in conflict monitoring and initiating speech in language switching (Luk, Green, Abutalebi, & Grady, 2012; Seo et al., 2018).

The non-verbal advantage of bilinguals could be a result of using control processes more often in a verbal context (Green & Abutalebi, 2013). Context-dependent control is exercised by a common mechanism, which is also active in non-linguistic situations. Thus, in situations where additional adaptive control is needed, bilinguals exhibit better adaptive responses compared to monolinguals, such as lower switch-costs (Prior & MacWhinney, 2010).

1.2 Rapid Instructed Task Learning

Although task-switching paradigms have shown a general advantage of bilingualism, they often cannot show the origin of the computational advantage. Tasks may not distinguish between executive function components or even the difference between task understanding and actual execution of instructions. This division is made in Rapid Instructed Task Learning (RITL; Cole, Laurent, and Stocco (2013); Stocco, Lebiere, O'Reilly, and Anderson (2012)), which involves high-paced task learning and execution. Figure 2 (A) shows an example RITL trial. Participants are presented with sets of instructions, a set of stimuli, and a probe. Each trial, the participant applies the set of instructions to the stimuli and answers accordingly. Although the main goal of the experiment remains the same, the instructions and stimuli change with each trial. This forces participants to quickly reconfigure their "mental template" of the task: they must apply new information to new instructions with every trial.

The trials can be separated into three phases. In the *encoding* phase, instructions are presented on the screen and remembered for further use. Instructions are familiar tasks, such as arithmetic and go/no-go paradigms (Stocco & Prat, 2014; Cole et al., 2013; Bialystok et al., 2004). The actual stimuli are presented in the *execution* phase. The instructions are applied to the stimuli. Next, the subject continues to the *response* phase, and communicates their answer. To separate the execution and response phases, participants may need to enter their answer in a separate screen (such as in Figure 2 (A), which includes a probe). A new trial starts after the response: the instructions change and the process starts over again.

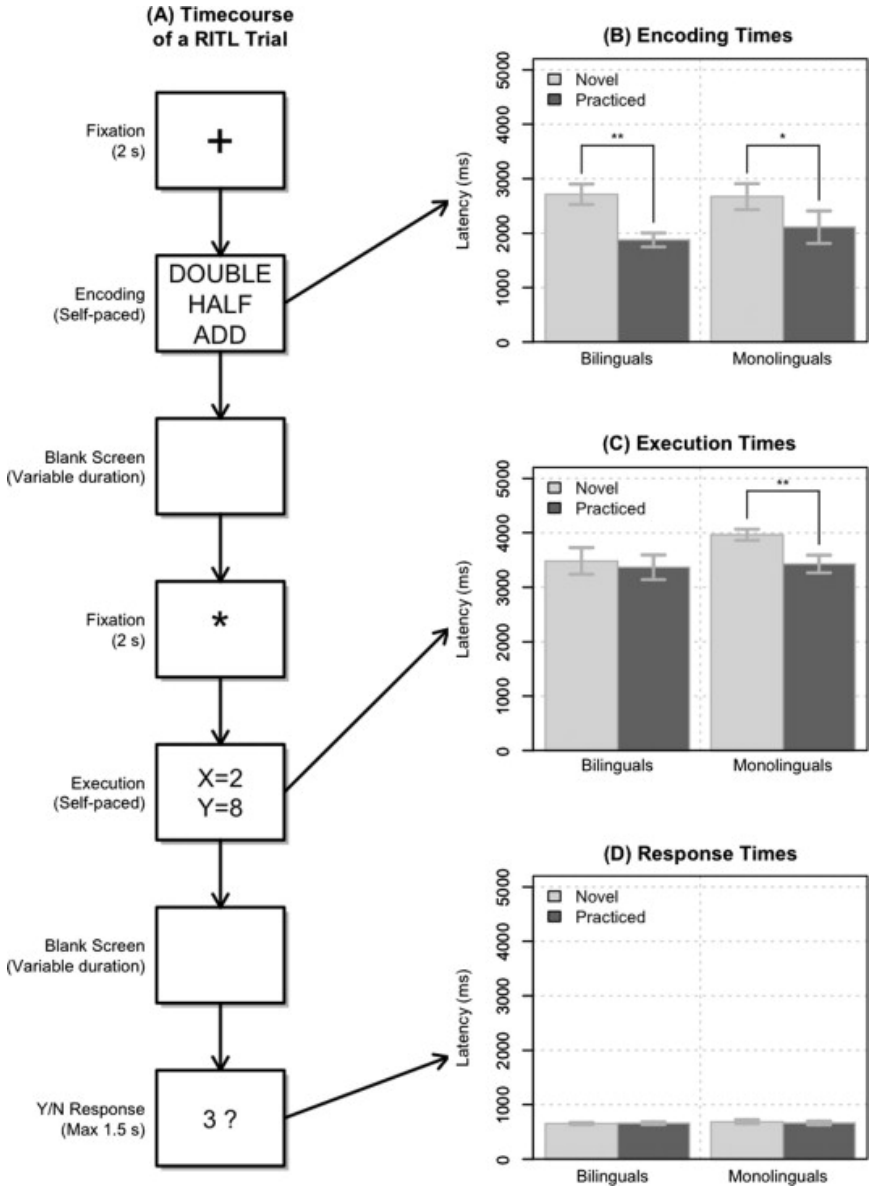

Figure 2: (A) Sample trial of the RITL paradigm used in by Stocco and Prat (2014). Participants started with the *encoding phase*, where instructions are presented describing the mathematical task to perform, such as "DOUBLE/HALF/ADD". During the *execution phase*, participants were then shown two input numbers, X and Y, on which the task specified by the instructions need to be applied. Finally, a numeric probe, such as "3 ?" was presented on the screen, and participants had to indicate whether the number was the result of their mental calculation. (B–D) Average reaction times of bilinguals and monolinguals during the Encoding (B), Execution (C), and Response (D) phases of the RITL paradigm. Bars represent means $\pm$ SEM. Line segments indicate statistically significant within-group differences: * $p < 0.01$, ** $p < 0.001$. Extracted from Stocco and Prat (2014).

1.2.1 Bilingual and Monolingual Performance on the RITL Task

Bilingual individuals must choose among various rules to achieve the same linguistic goal. For example, the standard rule for pluralization in English is to attach -s to the word: *human* becomes *humans*, *car* becomes *cars*, et cetera. However, in Dutch the suffix can be -en (*mens-mensen*) or -'s (*auto-auto's*). As bilinguals have multiple sets of linguistic rules, they are forced to flexibly maneuver around linguistic rule application, which supposedly gives rise to the greater flexibility in task-switching behaviors when faced with novel or changing rules (Prior & MacWhinney, 2010).

RITL paradigms may be a helpful tool to study the bilingual advantage, as they focus on rule application. They separate the memorization of rules from the application. Furthermore, because the exact rules and variables change with each trial, RITL paradigms focus more on task goals than content. This is similar to grammatical rule application: the overarching goal may remain the same, but the exact words and rules by which this is achieved differ within and between languages.

With this in mind, Stocco and Prat (2014) used a RITL paradigm to investigate the cognitive and neural mechanisms behind flexible rule application in advanced bilinguals. Bilinguals and monolinguals performed similarly on practiced tasks, but bilinguals were faster than monolinguals at executing novel tasks (see Figure 2 (C)). This finding supports the hypothesis that the bilingual advantage in executive functions is related to a better ability reconfigure rule-based behavior.

Stocco and Prat also collected fMRI data in that study, which showed that the basal ganglia of bilinguals and monolinguals react differently in the execution phase of the task (see Figure 1, left). The basal ganglia show higher neural activity to novel trials in bilinguals; the activity is reduced to almost zero once participants become familiar with the RITL instructions. For monolinguals, the activity remains low, regardless of set familiarity. This greater modulation in bilinguals supports the idea that better executive abilities are associated with improved neural adaptability (Prat, Keller, & Just, 2007). This becomes apparent in Figure 1 (right): shorter time and less resources are needed to apply novel rules.

1.3 Hypothesis

The difference between bilinguals and monolinguals is especially visible in task-switching paradigms that involve shifting between mental sets of rules (Prior & MacWhinney, 2010). More precisely, the results of Stocco and Prat show that bilinguals are faster than monolinguals when executing novel rules. They perform calculations for novel and practiced instructions at the same pace (see Figure 2 (C)). Bilinguals seem to already utilize a more efficient rule application strategy

than monolinguals during novel trials. As monolinguals achieve similar RTs for practiced trials, they may learned this computation strategy during practice.

This hypothesis also arises from both the conditional routing model and the adaptive control hypothesis described above. Monolinguals might need to exert more control when switching between RITL-rules and information. However, this mechanism would already be established in bilinguals, as they are used to switching between linguistic rules. Bilinguals may show greater neural adaptability in the basal ganglia than monolinguals: higher neural activity when executing novel trials, and reduced activity when executing practiced trials.

2 Models

Bilinguals and monolinguals may initially perform differently on Rapid Instructed Task Learning paradigms, but monolinguals seem to adapt their strategy to the demands of the execution phase of the task. A "bilingual" model and a "monolingual" model are presented in this section to show how these changes might come to be. The goal of the models is to show that the bilingual advantage observed in RITL-paradigms - and to a broader extend, other task-switching paradigms - is the result of a learned rule application strategy at the cost of internal checks. The mechanism that creates this strategy may cause the decrease in reaction times in monolinguals that Stocco and Prat observed in their RITL experiment (see Figure 2 (C)).

The full model code and simulation data can be found on `https://github.com/UWCCDL/ACTR_RITL`.

2.1 Task Description

The model task is based on the RITL procedures presented in Stocco and Prat (2014). Each trial consists of three parts: the *encoding* phase, *execution* phase, and *probe* (or *response*) phase (see Figure 4). Instructions are a combination of three operations, for example "DOUBLE", "HALF" and "ADD". The first two instructions are unary calculations ("DOUBLE", "TRIPLE", "INCREMENT", "DECREMENT", "THIRD", or "HALF") and need to be applied to X and Y respectively. The last operation is a binary operation ("TIMES", "DIVIDE", "ADD", or "SUBTRACT"), and applies to the results of the two previous calculations. The stimuli presented in the execution phase are two numbers between 1 and 9. The solution will always be between 0 and 50. The execution phase is self-paced. Finally, the participant indicates if the probe number corresponds to the solution.

Stocco and Prat (2014) presented eighty trials to each participant. Participants performed forty trials to familiarize themselves with the paradigm. The instructions and input numbers differed per trial, but could be duplicates. During a second session, which was administered within 24 hours, participants were presented with twenty previously trained instruction sets and twenty novel combinations. The trained instruction sets were two previously trained combinations, which were presented ten times each. The novel sets were randomly assembled instructions.

The task has been slightly modified for the models. They are first presented with twenty training trials; they did not process any distractors to keep more combinations of instructions available for the next phase. The practiced instructions were always identical (INCREMENT DOUBLE DIVIDE and TRIPLE INCREMENT ADD). The ACT-R baseline activation is the same for each individual instruction in declarative memory (see section 2.2.1), so the exact combinations of

practice instructions do not matter as long as they are different from the novel sets. The input numbers were still random numbers between 1 and 9. The testing phase immediately followed after the training phase. This phase consisted of forty trials again: twenty previously practiced instructions and twenty novel combinations. Again, the two trained sets were presented ten times each.

2.2 Model Implementation

2.2.1 Overview of the ACT-R architecture

The behavior on the RITL task was modeled in the the Adaptive Control of Thought-Rational (ACT-R) cognitive architecture (Anderson et al., 2004; Anderson, 2007) (see also `http://act-r.psy.cmu.edu/`). In the ACT-R architecture, cognition emerges through interaction between modules (see Figure 3a). For example, in a RITL task, the visual module processes the presented instructions, the retrieval module retrieves relevant mathematical information, the imaginal module manages the intermediate answers, the temporal module tracks time, et cetera. They work independently from each other, but can interact through buffers. Buffers can only contain one chunk at a time, which creates a bottleneck effect (Borst, Taatgen, & Van Rijn, 2010; Salvucci & Taatgen, 2008). Thus, the imaginal buffer may accept information from the visual buffer, but needs to remove its current content first. This bottleneck also extends to intermediate mental representations of the task (*problem states*, associated with ACT-R's imaginal module). If a task requires storage of intermediate results, the process of switching between representations will interfere with task execution. This is usually observed as delays in response time (Borst, Taatgen, & Van Rijn, 2010; Borst & Taatgen, 2007). In the case of a RITL-paradigm the bottleneck is in the storage and retrieval of the current task, intermediate results, and the remaining tasks.

ACT-R models have declarative knowledge (such as arithmetic facts) and procedural knowledge (mental operations) (Anderson, 2007). Declarative knowledge is represented as clusters (called *chunks*) of smaller features (called *slots*). For example, the presented RITL instruction set is one chuck, with three slots: each for every instruction. Both declarative and procedural knowledge have *activation*, which determines the likelihood that the chunk will be retrieved. The activation also determines how long it takes to retrieve a requested chunk.

The interaction between ACT-R modules is managed by the procedural buffer, which executes *production rules*. The procedural knowledge consists of the available set of production rules. The procedural module can only process the production rules in a serial manner (Salvucci & Taatgen, 2008; Borst, Taatgen, Stocco, & Van Rijn, 2010). These rules are programmable IF-THEN statements, that are

matched against buffer content. For example, IF the visual buffer contains an instruction set, THEN put the corresponding behavior in the retrieval buffer. Individual components of production rules are independent from task-specific knowledge that determines when task switching should occur. Individual components can therefore be separated and re-integrated into other tasks, which allows for flexible re-combination of previously learned behavior (Salvucci & Taatgen, 2008). When multiple rules follow each other, they may merge into one new rule. Explicitly memorized rules (or in our case, RITL instructions) are first stored in declarative memory before merging into task-specific production rules over time (Taatgen & Lee, 2003; Salvucci & Taatgen, 2008). For example, instead of retrieving the whole set of RITL instructions after performing the first operation of the set, the model may immediately skip to performing the next task, because it is already familiar with this set of RITL instructions. This process, called *production compilation* (Taatgen & Lee, 2003), reduces the use of the declarative memory by eliminating intermediate steps in the model. It making the process faster over time.

Each buffer is mapped onto brain areas associated to their task (Borst & Anderson, 2015; Anderson, Fincham, Qin, & Stocco, 2008) (see Figure 3). For example, the procedural buffer is associated with the caudate nucleus in the basal ganglia. The models will rely less on the declarative module and more on the procedural module as a result of production compilation (Salvucci & Taatgen, 2008). The implications of this process for the models and their representations in the brain are explained in section 2.2.4.

2.2.2 Model foundation

Figure 4 shows how the model flows through a trial. For both models, the RITL-instructions presented to a model (for example, DOUBLE HALF ADD) are one chunk with three slots. The declarative memory contains predefined chunks: ten arithmetic operations (such as DOUBLE for X*2, ADD for X+Y, et cetera) and 3778 facts about arithmetic operations rounded down to the nearest integer (i.e. 9 / 4 = 2). These predefined chunks have a relatively high baseline activation: this results in a quick retrieval of facts that are well-established in people.

The LISP-code can be found on `https://github.com/UWCCDL/ACTR_RITL`. A comparison between the bilingual and monolingual models' productions in the execution phase can be found in Appendix A.

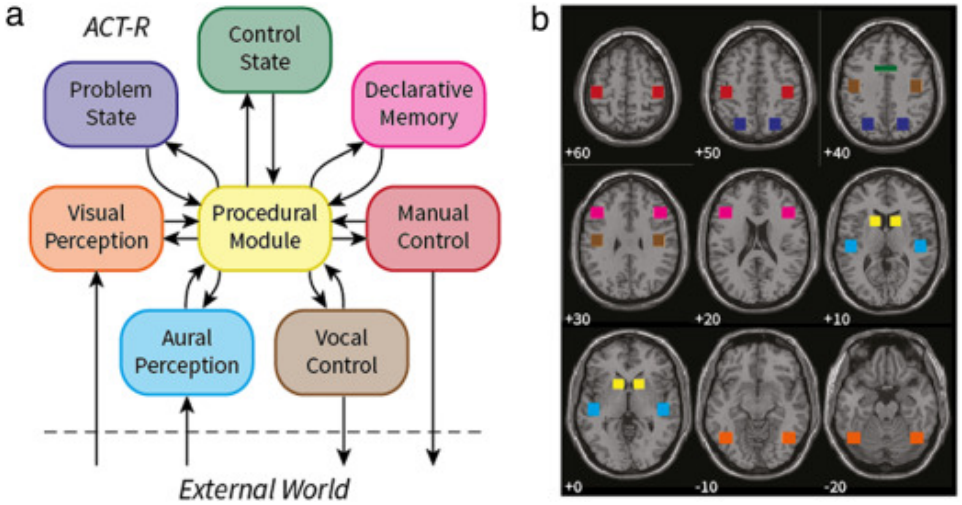

Figure 3: The ACT-R cognitive architecture (a) and its mapping to brain regions (b). The colors of the modules correspond to the colored squares in the brain. Retrieved from Borst & Anderson, 2015.

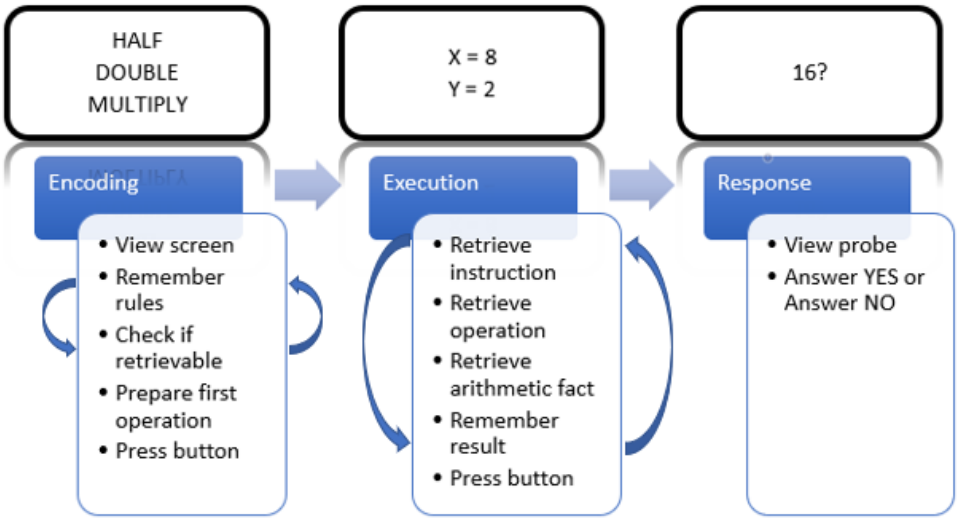

Figure 4: Overview of production flow through a trial. A set of three instructions (here: HALF DOUBLE MULTIPLY) is presented in the encoding phase. The models attempts to retrieve the instructions from declarative memory. If this is unsuccessful, the models take a new look and attempts again. Once the retrieval is successful, the instructions are copied in the imaginal buffer and first operation is prepared. The models press a button to continue. This process is identical in both the bilingual and monolingual model. During the execution phase, the models retrieve the instruction from the imaginal buffer, retrieve the appropriate operation (such as HALF = X/2) from declarative memory, and retrieve the arithmetic fact (8/2 = 4) from declarative memory. The result of the operation is stored in the imaginal buffer and the models continue with the next operation. These steps (retrieve instruction, operation, fact, and storage in buffer) are separate in the monolingual model, but have been merged in the bilingual model (see also Appendix A). Once all operation have been performed, the models press a button, and continue with the response phase. The final answer is compared with the answer in the imaginal buffer, and the model responses accordingly. This process repeats until all trials have been completed.

2.2.3 Encoding phase

The encoding phases of the monolingual and bilingual models are identical. The models attend the screen, place the content - a set of RITL instructions - in the imaginal buffer, and subsequently clear this buffer. This will place the instructions in declarative memory. Next, they attempt to retrieve the instructions from memory to check if the instructions have been stored properly. This cycle is guided by ACT-R's temporal buffer (Taatgen, Van Rijn, & Anderson, 2008). The temporal buffer starts to track time once the imaginal buffer has been cleared. The model will repeat the encoding process if more than $\pm$150 ms have passed before the instructions have been retrieved. The repetition increases the activation of the instructions chunk in declarative memory, which increases retrieval speed and likelihood. If the model successfully retrieves the instructions within $\pm$150 ms, the model will prepare for the first arithmetic operation while pressing a button to continue.

2.2.4 Execution phase

The models retrieve an operation (e.g. ADD means X + 2) and do calculations (e.g. 1 + 2 = 3) in similar fashion. The models retrieve results of the arithmetic operation from the declarative memory. Arithmetic facts and knowledge of operations are assumed to be well established in the declarative memory.

The bottleneck in intermediate storage becomes most apparent in the execution phase. As buffers contain only one chunk at a time, the models needs to swap information between buffers to store instructions and results to use them later on. The strategy of swapping intermediary information between buffers is the essential difference between the monolingual and bilingual models (see Table 1 and Appendix A). The bilingual model's strategy is more efficient from the start: it remembers an answer and prepares the next instruction in one production. The monolingual model needs to learn this strategy with practice.

The monolingual model initially separates rule application and variable binding: calculation (*execution* goals in Table 1), storing information (*update-scratchpad* goals in Table 1), and information retrieval (*retrieve-task* goals in Table 1) all have their own separate productions. These may merge into one action by allowing production compilation, which allows the monolingual model to learn a more efficient switching strategy between RITL rules and variables. The middle row of Table 1 shows the monolingual model's intermediate goals after production compilation.

The production rules of the bilingual model for the execution phase are based on the monolingual model's rules after production compilation: rule application and variable binding may happen simultaneously. For example, the production rule *update-scratchpad-y-start-binary* remembers the solution to the second arith-

metic problem while retrieving the remaining third problem. In other words, the bilingual model has already learned what the monolingual model can learn during task execution. For the bilingual model, the instructions and results remain active in working memory, while retrieving answer to arithmetic operations at the same time. The monolingual model does not have this ability initially, and needs to retrieve the instructions again after retrieving an arithmetic fact. The comparison between monolingual and bilingual functions in Appendix A show this: the monolingual model switches between the chunk that contains the three task instructions (*ritl-task*) and the intermediate results (*ritl-result*). The bilingual model retrieves task instructions immediately after solving the previous problem.

Internal checks can be removed after production compilation. They are buffer queries checking, for example, whether a buffer is busy processing new content or whether content has been placed in the buffer. Removing buffer checks speeds up information processing and creates a more flexible control mechanism. In theory, this creates the possibility of empty or unavailable buffers, leaving more room for errors during trials (Salvucci & Taatgen, 2008). This is not the case nor the goal of this model; the model goals have been strictly defined and guide the models through the process.

The connections between ACT-R modules and their associated brain areas are particularly interesting in the execution phase. Based on the results from Stocco and Prat (2014), it is expected that the bilingual model may especially rely on the basal ganglia when processing novel instructions: the procedural module should therefore show a pattern similar to Figure 1 (left). We also expect a difference between the bilingual and monolingual model in the imaginal module (or problem state) and the retrieval module (or declarative memory), which are respectively associated with the posterior parietal cortex and the ventrolateral pre-frontal cortex (Anderson et al., 2008). Initially, the monolingual model relies more on the the imaginal and declarative modules to switch between intermediate results, but this should reduce over time, as the monolingual model becomes more familiar with shifting between RITL instructions. Note that the procedural module activity is also affected by this process. As the bilingual model executes instructions efficiently regardless of trial familiarity, it will probably rely less on the imaginal and declarative modules than the monolingual model.

2.2.5 Response phase

The response phases are identical again. The models check the scratch pad for a result. If the result corresponds to the probe on the screen, the models answer 'yes' using the keyboard. If not, they answer 'no' using the keyboard.

Because the models have been designed to retrieve the right instruction set,

Table 1: Model goals during the execution phase of the monolingual model at the start, after the learning phase and for the bilingual model. Although the bilingual model was capable of learning, the model's productions did not change over time.

Monolingual Model	Monolingual Model After Learning	Bilingual Model
execution-x	execution-x	execution-x
retrieve-task-y		
execution-y	execution-y	execution-y
update-scratchpad-y	update-scratchpad-y	update-scratchpad-y
retrieve-task-binary		
execution-binary	execution-binary	execution-binary
update-scratchpad-binary	update-scratchpad-binary	update-scratchpad-binary
done	done	done

they won't make mistakes in this phase. Note that an actual accuracy of 1.0 is unrealistic. Bilinguals made slightly more mistakes than monolinguals in the Stocco and Prat experiment (respectively 87% and 92%), but this difference was not significant.

2.3 Simulations

2.3.1 Model parameters

Table 2 presents the range under which parameter effects were examined. The parameter's range refers to the range of values over which a full grid search was conducted to explore the model behavior and performance. For each unique combination of *alpha* α*, ans s, imaginal-delay, le f*, and *nu* ν, the model was run 100 times.

As explained in section 2.1, the models were presented with identical data sets of 60 trials during each run. Two combinations of instructions (always INCREMENT DOUBLE DIVIDE and TRIPLE INCREMENT ADD) are practiced 10 times each in random order. Immediately after these 20 trials, 40 trials are presented. Half of these consisted of practiced instructions and half of these are new instructions. The input numbers, which are presented in the execution phase, differed with each trial.

2.3.2 Model selection and fit

We used two approaches to select the most appropriate model to the data. First, we selected the model with ACT-R's default values, except for activation noise parameter *ans (S)*. This parameter controls random noise in activation of chunks in the declarative knowledge, and has no pre-defined default value. Furthermore, we also selected the parameters with the best fit to Stocco and Prat's (2014) experiment over the whole parameter space. The best fit to the results of the behavioral RITL experiment was found by identifying the unique set of parameter values that minimized the sum of squared errors between the mean responses times in the experiment by Stocco and Prat and the response times predicted by the models (Root Mean Square Error). See section 3.1 for the fitting process and the resulting model parameters. We found a different parameter set when maximizing the correlation with the behavioral experiment. The model based on RMSE was selected for further use.

The model with the best fitting parameters was also used to create predictions of various Regions-of-Interest (Borst & Anderson, 2017). This helps to further evaluate the validity of our modelling approach to rapid instructed task learning. The results are presented in section 3.2.

Table 2: Parameters investigated during simulations.

Name	Function	Range	Default
alpha α	Learning rate	0.0 - 0.5	0.2
ans s	Activation and utility noise	0.01 - 0.04	-
imaginal-delay	Processing requests to imaginal buffer	0.15 - 0.25	0.2
le f	Retrieval exponent factor	0.9 - 1.1	1.0
nu ν	Starting utility for a newly learned production	0.0 - 0.5	0.0

3 Results

3.1 Model Fit to Experimental Data

Two fitting methods were used to select the parameter set that creates the best model fit to Stocco and Prat's (2014) behavioral data. We used the root mean square error (RMSE), which was calculated by squaring the root of the difference between the mean models' response times and the mean behavioral reaction times. Next, the RMSEs of the encoding and execution phases are added and the minimal value is taken as the value with the smallest deviation from the behavioral data. The best model fit differs about 400 ms from Stocco and Prat's experiment when all parameters are considered, and 864 ms when using ACT-R's default values (Table 3). Note that the parameters *alpha*, *imaginal-delay*, *le*, and *nu* are all at the end of their range (although 0 is the default value for *nu*). See Table 4 for the response times of the best fitting model. See Figure 5 for a comparison between Stocco and Prat's data and the model's RTs.

The other fitting method is based on the Pearson correlation between model data and behavioral data. The mean RTs per parameter set are correlated to the mean behavioral data for each task phase. The Pearson's *r*s of each phase are added, making the range of this metric -2 to 2. The largest positive value is used for parameter fitting. Overall, the models correlated better to the encoding phase than to the execution phase of the behavioral experiment: the highest Spearman's *r* value for a single encoding phase was 0.998, but for a single execution phase was 0.511. In contrast to the RMSE, Spearman's *r* relies stronger on variability. However, because we use the means to calculate the correlation - in opposite to raw data - the variability in the input data is greatly reduced. This has consequences for the robustness of the resulting *r*.

The correlation-based parameter set was slightly different from RMSE-based set, but the resulting RTs were remarkably similar. The RSME-based model deviated 400 ms from the behavioral data; the added correlation was 1.314. The correlation-based method produced a model with a total RMSE of 490 ms and an added correlation of 1.356. The correlation with the behavioral data is marginally better, but as we attempt to reproduce the Stocco and Prat's data, a smaller absolute deviation from the behavioral data is more desirable. Thus, we prefer to use the RMSE-based parameters for further analyses.

Table 3: ACT-R model parameter values with smallest RMSE to the behavioral data of Stocco and Prat (2014). The first row contains the parameters with the smallest added Root Mean Square Error values. The second row shows the *ans* value with the smallest added RMSE; all other parameters were set to the ACT-R default value.

Alpha	Ans	Imaginal-Delay	le	nu	Error enc + ex (ms)
0.5	0.03	0.15	0.9	0.0	400
0.2	0.01	0.2	1.0	0.0	864

Table 4: Response times of the models with the parameter set that produces the smallest added Root Mean Squared Error. See Table 3 for model parameters.

Type	Language	Encoding	Execution
Novel	Bilingual	2745	3446
Practiced	Bilingual	1960	3398
Novel	Monolingual	2626	4186
Practiced	Monolingual	1661	3886

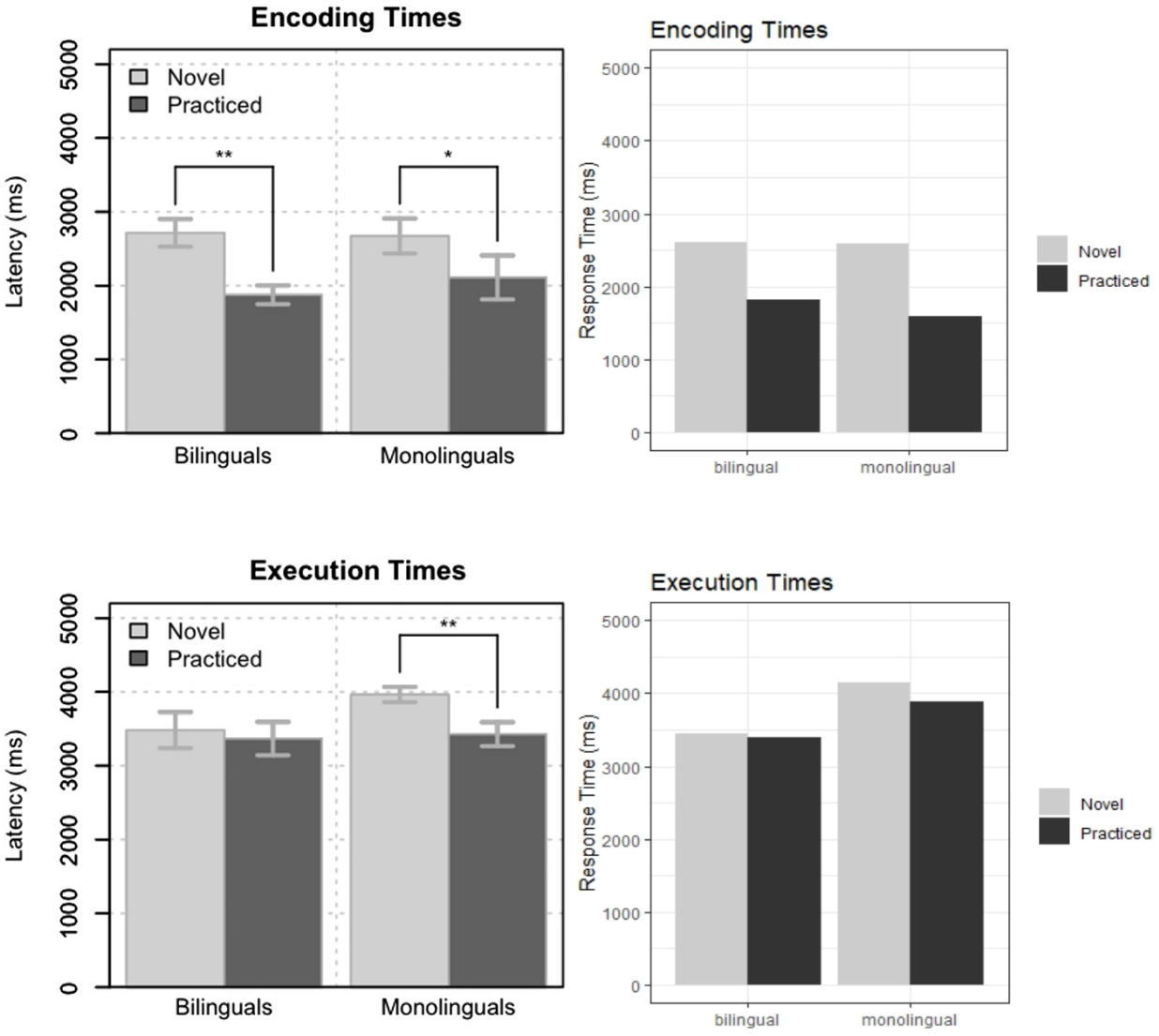

Figure 5: Comparison between encoding and execution results from Stocco and Prat (2014) (left) and model results based on RMSE-based parameter fitting (right). See Table 3 for parameter values.

3.2 Correspondence to Neural Correlates

ACT-R's modules are tied to specific brain areas (Figure 3). Following Borst and Anderson's (2017) tutorial, we created Blood Level Oxygen Dependent (BOLD) predictions of various Regions-of-Interest (ROIs) of the models with the best fitting parameters based on the RMSE. We extracted module activity of the declarative memory / retrieval module, problem state / imaginal module, manual module, visual module, goal module, and production module over the whole trial. The predictions are created by convolving module activity with a typical Hemodynamic Response Function. This results in a prediction of the BOLD response of the brain areas associated to the modules. The code and data can be found at `https://github.com/UWCCDL/ACTR_RITL/ACTR_BOLD`. The results are presented in Figure 6.

There is a clear difference between the monolingual and bilingual models when processing novel trials. The predictions for declarative memory, problem state, and the procedural module are important for our hypothesis. Differences in these modules start to emerge around five seconds into the trial - thus *after* the encoding phase. As expected, the monolingual model needs to use the declarative memory and problem state the most when processing new trials. The production module is most active in the monolingual model as well. The bilingual model activity in response to novel instruction sets is right between the monolingual model's response, and practiced instructions. The bilingual model activity in the procedural module also decreases earlier than the monolingual model, which could imply greater modulation. On the other hand, this could also be due to the bilingual model responding faster in general. The signal change in the manual module in response to novel trials suggests the latter: the decrease in activity starts earlier in the bilingual model.

The models react almost completely similar to practiced instructions. The goal module is the exception: it seems that the monolingual model is systematically less active, which might be attributed to differences in goal creation after production compilation. However, the overall signal change is very small - probably not significant.

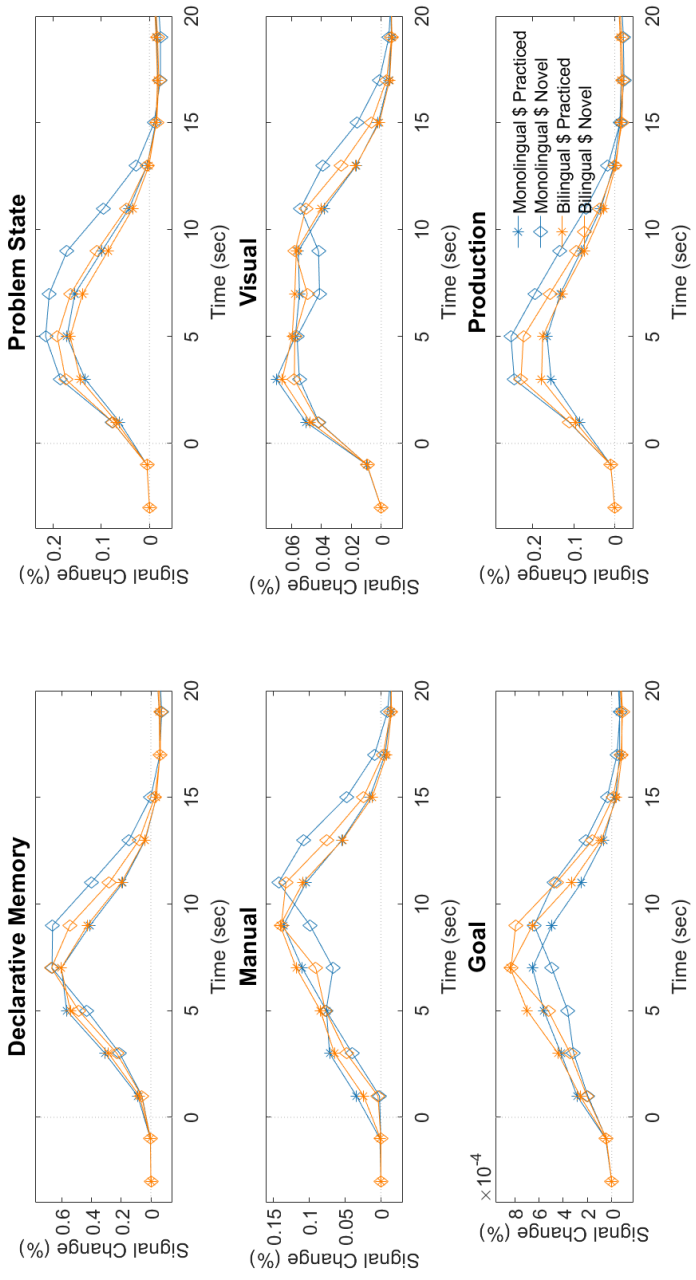

Figure 6: Model predictions for six ACT-R modules.

3.3 Model Complexity

Besides finding the right parameters that produce the experimental outcomes, the model principles may stand regardless of parameter values. This can be examined through parameter space partitioning (Pitt, Kim, Navarro, & Myung, 2006). We investigated the extend to which the models can reproduce the pattern of Stocco and Prat's (2014) study for the all simulations over the whole rage of parameters (see Table 2). We expect that both models need to learn an efficient way to encode various instruction sets. Thus, the models should always produce a higher encoding RT for novel instruction sets compared to practiced instruction sets (see Figure 2, B) regardless of the exact parameter settings. More importantly, the monolingual model should always produce a decrease in execution RT over the whole parameter space (Figure 2, C).

3.3.1 Encoding phase

The parameter space is created by aggregating all possible parameter combinations. Then, we compare the differences in reaction time between novel and practiced instruction sets in each phase and variation of the model. Practiced trials have a lower encoding RT than novel trials for 67.2% of all parameter combinations in the bilingual model. The monolingual model acts as predicted: practiced trials have a lower encoding RT in 99.8% of all parameter combinations. Further investigations showed that the monolingual model has lower encoding RTs than the bilingual model in 99.7% of all parameter combinations. This suggests that the monolingual model reduces the encoding RTs much stronger than the bilingual model, despite using the same encoding mechanism.

The parameters *imaginal-delay* and *le* influence the model response times in particular (see Figure 7). The imaginal-delay parameter controls how long it takes a request to the imaginal buffer to complete. *le* determines the latency by which chunks are retrieved. Both parameters influence the drop in RTs between novel and practiced instruction sets, but the influence of *le* is stronger.

3.3.2 Execution phase

The monolingual model accurately predicts a difference between novel an practiced trials in the execution phase: 100% of all parameter combinations produce a lower execution RT on practiced trials. It is unexpected that the bilingual model *also* produces lower response times on practice trials in 100% of the trials. Bilinguals executed practiced trials marginally faster in Stocco and Prat's experiment, but the RTs did not significantly differ from each other (see Figure 2). If we look at the distribution of the mean model RTs over the whole parameter space (Figure 8),

we see that the bilingual model performs practiced sets faster as well, but the difference is not as substantial as the monolingual model. The pattern is still similar to the Stocco and Prat experiment.

The parameters *imaginal-delay*, *le*, and *nu* influence the average response times in the execution phase. No particular parameter value results in the monolingual response times dropping to the bilingual level. The patterns that the parameters produce are different from the encoding phase. The imaginal delay parameter has no influence on the bilingual model, which suggests that the requests to the imaginal buffers are handled differently compared to the monolingual model. The monolingual model shows its own pattern when processing practiced trials, suggesting that there are still differences between the models when processing familiar instruction sets. On the other hand, the monolingual model shows the same pattern as the bilingual model to practiced trials for the *le* parameter. Parameter *nu* approximates Stocco and Prat's results best when set to the default value of 0.

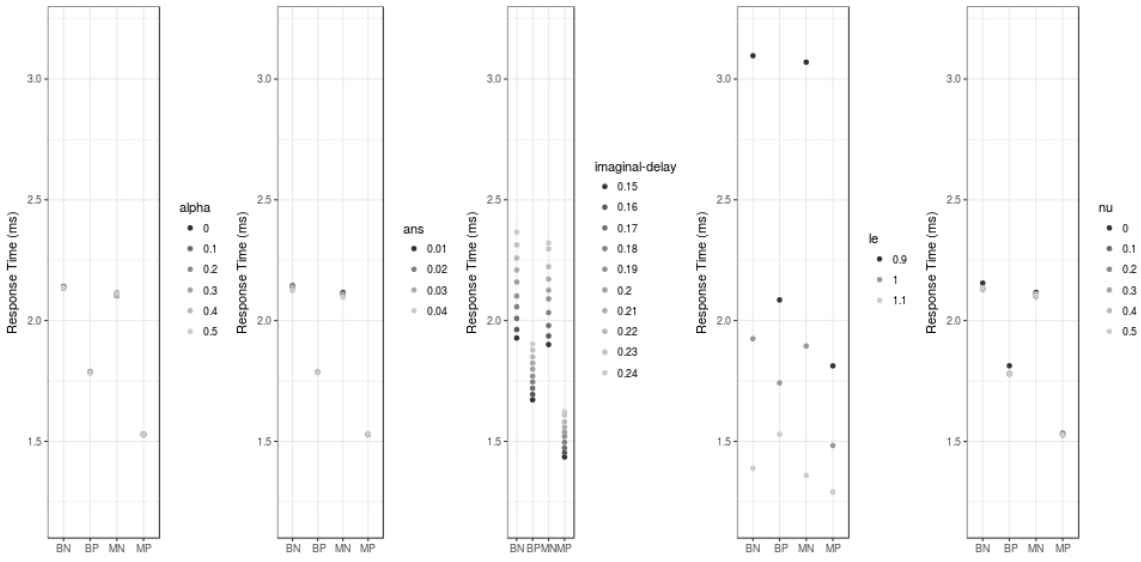

Figure 7: Mean response times (ms) to novel and practiced RITL instructions of the bilingual and monolingual models in the encoding phase, grouped by ACT-R parameters *alpha*, *ans*, *imaginal-delay*, *le* and *nu* respectively. B = bilingual model, M = monolingual model. N = novel trials, P = practiced trials.

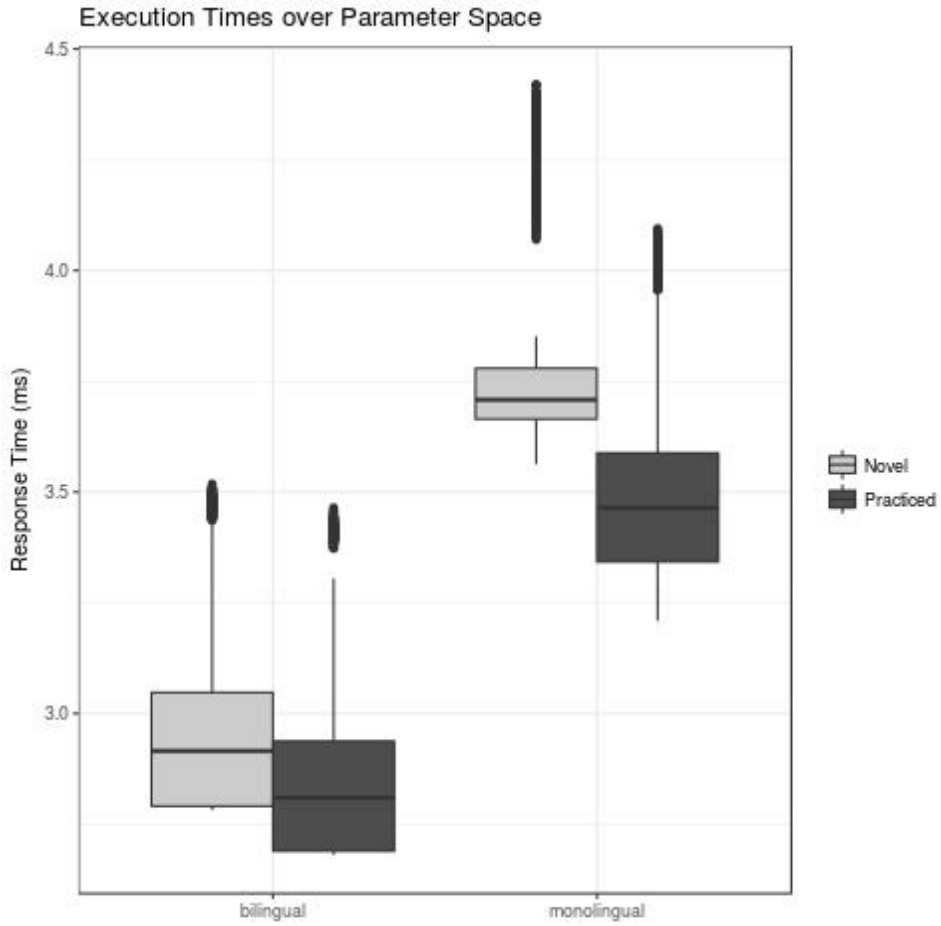

Figure 8: Boxplot showing the distribution of mean executions times of novel and practiced instruction sets over the whole parameter space for the bilingual and monolingual model.

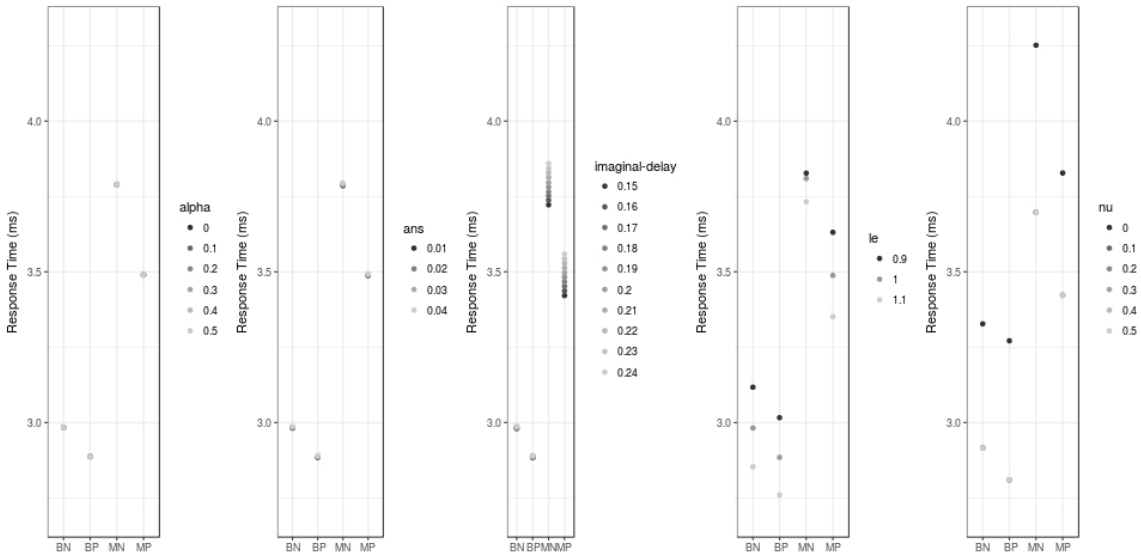

Figure 9: Mean response times (ms) to novel and practiced RITL instructions of the bilingual and monolingual models in the execution phase, grouped by ACT-R parameters *alpha*, *ans*, *imaginal-delay*, *le* and *nu* respectively. B = bilingual model, M = monolingual model. N = novel trials, P = practiced trials.

4 Discussion

The difference in task-switching behavior on RITL tasks between bilinguals and monolinguals may result from differences in internal control, leading to a more efficient information exchange in bilinguals. Two ACT-R models were created to test this hypothesis; the monolingual model initially has more internal checks. More specifically, the model actively retrieves the next task in a separate production, but the bilingual model retrieves the next sub-tasks while storing information from the previous one. Over time, the monolingual model learns to act the same (see Table 1). The models could accurately reproduce the behavioral patterns observed in Stocco and Prat (2014) (Figure 5). These findings persist over the whole parameter space (Figure 8). However, the monolingual model could not predict the exact reaction times of the behavioral experiment. Thus, it seems that the model mechanism is accurate, but parameters or - more likely - production rules need to be refined to replicate the reaction times. This is further supported by the BOLD predictions of the ACT-R production module (Figure 6).

4.1 Encoding Phase

Parameter fitting gave a relatively satisfying result for the encoding phase: the response times of the bilingual model fit well with Stocco and Prat's data (Figure 5), but the monolingual model seems to encode the practiced instructions a bit too fast. Both the Root Mean Square Error and correlation to the behavioral results are favorable in this phase. The parameter set with the smallest RMSE in the encoding phase is the same as the set that produces the smallest overall RMSE (see Table 3 and Figure 5, right). The best fitting models have only little variability, indicating that RITL instructions need to be well established in memory for the behavioral results to take place.

The parameters *imaginal-delay* and *le* are particularly important for fitting the encoding phase: they are at the end of the range (compare Table 2 and Table 3). These parameters are respectively related to imaginal module delay and latency of retrieval. This suggest that fast learning and retrieval have a strong influence on successful modeling of the encoding phase, which is not very surprising. After all, this phase consists of repeatedly retrieving and forgetting the presented set of RITL instructions. The simulation value for parameter *le* (retrieval exponent, determines latency by which chunks are retrieved) may have been too coarse (Figure 7). A *le* value between 0.9 and 1.0 may lead to a better approximation of Stocco and Prat's results.

The monolingual model reduces the encoding RTs stronger than the bilingual model over the whole parameter space (Pitt et al., 2006), despite using the same

encoding mechanism. We expected that the encoding RTs of practiced instructions sets would always be lower than novel instruction sets. Only the monolingual model behaves as expected: practiced instruction sets are encoded faster than novel sets in 99.8% of all unique parameter combinations. This is only 67.2% of parameter combinations in the bilingual model. In other words, the monolingual model can produce the desired pattern, but the bilingual model approximates Stocco and Prat's response times better.

In conclusion, the memorization mechanism works best for models that learn and retrieve quickly, which leads to only small variations in encoding response times. Although both models follow the same set of productions in the encoding phase, the monolingual model encodes practiced instructions a bit too fast, but consistently lower than novel instructions. Still, we cannot confirm that this is the way participants remembered the instructions. However, this mechanism mostly served to ensure that the instructions would retain in memory during the execution phase.

4.2 Execution phase

In Stocco and Prat's (2014) study, monolinguals responded around 500 ms faster on practiced trials while bilinguals responded at the same pace (Figure 5). We attempted to recreate this effect by using production compilation in a model that still had to learn to efficiently switch between the current task and intermediate results. Contrary to our expectations, our models always executed practiced instruction sets faster, regardless of model variety and parameter values. This reduction was only marginally smaller in the bilingual model (Figure 8). Further investigations attribute this finding to compilation of content specific productions (e.g. INCREMENT = X+1) and decrease in retrieval latency due to increased activation of practiced instruction sets. This slight difference is nevertheless in line with Stocco and Prat's results. However, the RTs of the monolingual model never dropped to the bilingual level (compare Figure 5 to Figure 8 and Figure 9). This suggests that the current mechanism used by the monolingual model is not enough. This is also supported by Figure 9: the differences between the bilingual and monolingual model by *imaginal-delay* parameter suggest that the the models still process familiar instruction sets differently. On the other hand, the monolingual model shows the same pattern as the bilingual model to practiced trials for the *le* parameter. Moreover, the *nu* parameter fits best with with Stocco and Prat's behavioral results when it is set to 0 - the default value. The parameter determines the starting utility for a newly learned production. This suggests that regular production compilation indeed plays a role in the computational mechanism behind performance on the RITL task. In short, the model mechanism can consistently produce execution time pat-

tern, but only approaches the monolingual response times we saw in Stocco and Prat (2014).

Bilinguals learn to handle the extra effort of processing more linguistic rules (Stocco et al., 2010, 2014; Prior & MacWhinney, 2010). Figure 1 suggests that the additional effort of processing novel instruction sets is met with additional activity in the basal ganglia in bilinguals. This image also explains why the mean activation of bilinguals is so diverse when executing novel instructions: Figure 1's right graph may show individual differences in basal ganglia adaption. The small, but constant, activation of the basal ganglia in monolinguals can be explained in a similar fashion in Figure 1: in both novel and practiced situations, monolinguals need to adapt in ways they haven't learned before. We observe this in the model behavior as well: the monolingual model creates many new individual productions, but the bilingual model only a few.

We predicted Regions-of-Interest activity of the models to further validate and investigate the model implications (Borst & Anderson, 2017). In Stocco and Prat's study, the basal ganglia of the bilingual participants showed most activity when executing novel instruction sets. The production module of our models, which is associated to the basal ganglia (Borst & Anderson, 2015), does not show the same pattern for novel trials as Figure 1 (right). The monolingual model shows more activity over the whole trial. The bilingual model returns earlier to the level of practiced trials, which can be attributed to greater modulation, but also to just being faster in general. The models also show differences between novel and practiced trials in the problem state / imaginal module and the declarative memory / retrieval module. Their associated brain areas are respectively the posterior parietal cortex and pre-frontal cortex (Figure 3) (Borst & Anderson, 2015). Although these areas have not been observed in Stocco and Prat's experiment, they do fit with the Adaptive Control Hypothesis (Abutalebi & Green, 2016, 2007), where they are associated with the selection of target stimuli among interfering information, such as second language usage.

Note that the building strategy of the models influences the models' performance. The bilingual model is, essentially, the final product of the learning process of the monolingual model. It is therefor not surprising that these models behave similarly on practiced trials. It's safe to assume that the learning process bilinguals experience during a life time of language-switching is much more elaborate than the learning process of monolinguals when solving RITL trials. However, for the sake of simplicity, it is easier to assume that bilinguals and monolinguals only differ in their use of control mechanisms. Stocco and Prat's participants were also sufficiently balanced. It seems unlikely that participants substantially differed on aspects related to ACT-R's parameters, such as internal noise, starting utility for a new RITL procedure, or (non-linguistic) learning rate.

4.3 Neural Correlates of Differences in Task-Switching

Language-related differences in task-switching have been found in the prefrontal cortex, the anterior cingulate cortex, and the basal ganglia (Stocco & Prat, 2014; Abutalebi & Green, 2016; Seo et al., 2018). According to the conditional routing model, the basal ganglia steer information towards the prefrontal cortex if the conditions change. For bilinguals, this means that they need to choose between multiple rules for different languages, which puts additional strain of the basal ganglia circuit. The basal ganglia could be trained better to switch between sets of rules, and this skill may be "flowing over" into non-linguistic tasks as well. This idea is further supported by the finding that the basal ganglia keep track of the target language during a bilingual RITL paradigm (Seo et al., 2018).

In the ACT-R architecture, the basal ganglia are related to the procedural module, which coordinates the interaction between other modules (Anderson et al., 2008, 2004) and is important for learning (Salvucci & Taatgen, 2008). The BOLD predictions of the procedural module in our monolingual and bilingual models (Figure 6) could not replicate the basal ganglia pattern similar to Figure 1. The prefrontal areas are associated with declarative memory (the retrieval module). The posterior parietal cortex with the problem state (the imaginal module). The problem state is a directly accessible intermediate representation of the current state of a task (Borst, Taatgen, & Van Rijn, 2010). The monolingual model needs to retrieve the presented novel instruction sets and intermediate results more often than the bilingual model; it has not learned yet how to efficiently switch between information (Salvucci & Taatgen, 2008). Thus, these two modules should show more activity during novel trials - a pattern similar to Figure 5. Figure 6 confirms these expectations. The declarative memory and problems state findings are in accordance with the Adaptive Control Hypothesis (Abutalebi & Green, 2016, 2007): their associated areas are part of the *control network*. Interestingly, the dorso-lateral pre-frontal cortex has been found to be active in bilinguals during RITL task execution (Seo et al., 2018). The activity in these modules of the bilingual model is similar regardless of set familiarity, suggesting that the bilingual model already "knows" how to use the imaginal module and, to a lesser extend, the declarative memory. Thus, the current study supports the hypothesis that bilinguals already know how to switch efficiently between task information. The results suggest that the model mechanisms of production compilation and removal of internal checks indeed contribute to the bilingual advantage in task-switching.

4.4 Further Investigations

The models could be further optimized to simulate the behavior of monolinguals to novel trials better and to re-create the basal ganglia activity of bilinguals to novel instruction sets. This could be achieved, for example, by investigating why parameters result in different RTs patterns in the encoding and execution phase (compare Figure 7 and Figure 9). Moreover, we suggest a more sensible *le* parameter range. The one for this study seemed too coarse.

The models invite us to investigate other behavioral differences between bilinguals and monolinguals as well:

4.4.1 Accuracy

Although the bilingual group in Stocco and Prat was faster in the processing of novel trials, they made slightly more mistakes. This effect was not significant, but we may explain this with our model as well: they can be attributed to the removal of imaginal buffer checks resulting in the inability to retrieve the right set of instructions. Earlier versions of the models reflected this tendency, but were ultimately discarded, as accuracy is not the main focus of this investigation. Nevertheless, other studies found this effect as well (Prior & MacWhinney, 2010), but the effect is not significant either. The models could be developed further and used to investigate this marginal difference in accuracy in the future.

4.4.2 Applying model principles to other tasks

The monolingual model could reproduce the drop in response times when internal checks are removed and production compilation (Taatgen & Lee, 2003) was allowed. This opens doors for other task-switching paradigms, especially ones that have been modeled in ACT-R already, such as a traffic control task (Taatgen & Lee, 2003), or a multitasking paradigm that challenges ACT-R's information bottleneck (Borst, Taatgen, & Van Rijn, 2010). Borst et al. (2010) let participants alternate between typing a 10-letter word and subtracting a number. Earlier studies showed that people can only keep one intermediate task representation, or *problem state*, at the time, creating a task-switching bottleneck. In practice that meant, for example, that participants had difficulty switching midway between typing the remaining letters of an obscured word. The behavioral results reflected this bottleneck: participants were slower when they had to maintain much information in working memory. The task has also been modeled successfully using ACT-R. The authors collected neuroimaging data during the task, and compared those to the "BOLD response" the model produces (Borst, Taatgen, Stocco, & Van Rijn, 2010; Borst & Anderson, 2015). The left SMA, which is associated to language switching (Abutalebi

& Green, 2007, 2016), showed more activation in the hard conditions of the tasks. The prefrontal cortex, related to ACT-R's declarative module, showed an effect of task difficulty, but the expected interaction effect as a result of encoding problem states was not present. The authors suggested that this region's contribution to the processing problem states was perhaps too weak to impact the BOLD signal, or the retrieval of intermediate problem states is controlled by a different region.

The task has some overlap with the RITL paradigm: intermediate steps need to be stored and retrieved. Furthermore, the neuroimaging result of this task and the RITL paradigm can be compared, and may give insight in task-switching and the influence of the bilingual experience. If a bilingual and monolingual participant pool would perform the task, we may infer that bilinguals are more used to switching between information, and thus perform better in both difficult conditions of the task. Moreover, we may infer that bilinguals are initially faster than monolinguals in continuing with and finishing the task, as no queries of the imaginal buffer or declarative buffer, where problem state information is stored. However, monolinguals may catch up later. Again, this effect might be the result of reducing internal control in exchange for flexibility.

References

Abutalebi, J., & Green, D. (2007). Bilingual language production: The neurocognition of language representation and control. *Journal of neurolinguistics*, *20*(3), 242–275.

Abutalebi, J., & Green, D. W. (2016). Neuroimaging of language control in bilinguals: neural adaptation and reserve. *Bilingualism: Language and cognition*, *19*(4), 689–698.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036.

Anderson, J. R., Fincham, J. M., Qin, Y., & Stocco, A. (2008). A central circuit of the mind. *Trends in cognitive sciences*, *12*(4), 136–143.

Bialystok, E. (2009). Bilingualism: The good, the bad, and the indifferent. *Bilingualism: Language and cognition*, *12*(1), 3–11.

Bialystok, E., Craik, F. I., Klein, R., & Viswanathan, M. (2004). Bilingualism, aging, and cognitive control: evidence from the simon task. *Psychology and aging*, *19*(2), 290.

Borst, J. P., & Anderson, J. R. (2015). Using the act-r cognitive architecture in combination with fmri data. In *An introduction to model-based cognitive neuroscience* (pp. 339–352). Springer.

Borst, J. P., & Anderson, J. R. (2017). A step-by-step tutorial on using the cognitive architecture act-r in combination with fmri data. *Journal of Mathematical Psychology*, *76*, 94–103.

Borst, J. P., & Taatgen, N. A. (2007). The costs of multitasking in threaded cognition. In *Proceedings of the eighth international conference on cognitive modeling* (pp. 133–138).

Borst, J. P., Taatgen, N. A., Stocco, A., & Van Rijn, H. (2010). The neural correlates of problem states: Testing fmri predictions of a computational model of multitasking. *PLoS One*, *5*(9), e12966.

Borst, J. P., Taatgen, N. A., & Van Rijn, H. (2010). The problem state: A cognitive bottleneck in multitasking. *Journal of Experimental Psychology: Learning, memory, and cognition*, *36*(2), 363.

Carlson, S. M., & Meltzoff, A. N. (2008). Bilingual experience and executive functioning in young children. *Developmental science*, *11*(2), 282–298.

Cole, M. W., Laurent, P., & Stocco, A. (2013). Rapid instructed task learning: A new window into the human brain's unique capacity for flexible cognitive control. *Cognitive, Affective, & Behavioral Neuroscience*, *13*(1), 1–22.

Crinion, J., Turner, R., Grogan, A., Hanakawa, T., Noppeney, U., Devlin, J. T., ... others (2006). Language control in the bilingual brain. *Science*, *312*(5779), 1537–1540.

Eurostat. (2018, 04 23). *Number of foreign languages known (self-reported) by sex.* (data retrieved from `http://appsso.eurostat.ec.europa.eu/nui/submitViewTableAction.do`)

Green, D. W., & Abutalebi, J. (2013). Language control in bilinguals: The adaptive control hypothesis. *Journal of Cognitive Psychology*, *25*(5), 515–530.

Luk, G., Green, D. W., Abutalebi, J., & Grady, C. (2012). Cognitive control for language switching in bilinguals: A quantitative meta-analysis of functional neuroimaging studies. *Language and cognitive processes*, *27*(10), 1479–1488.

MacWhinney, B. (1997). Second language acquisition and the competition model. *Tutorials in bilingualism: Psycholinguistic perspectives*, 113–142.

Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The unity and diversity of executive functions and their contributions to complex "frontal lobe" tasks: A latent variable analysis. *Cognitive psychology*, *41*(1), 49–100.

Pitt, M. A., Kim, W., Navarro, D. J., & Myung, J. I. (2006). Global model analysis by parameter space partitioning. *Psychological Review*, *113*(1), 57.

Prat, C. S., Keller, T. A., & Just, M. A. (2007). Individual differences in sentence comprehension: a functional magnetic resonance imaging investigation of syntactic and lexical processing demands. *Journal of cognitive neuroscience*, *19*(12), 1950–1963.

Prior, A., & MacWhinney, B. (2010). A bilingual advantage in task switching. *Bilingualism: Language and cognition*, *13*(2), 253–262.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological review*, *115*(1), 101.

Seo, R., Stocco, A., & Prat, C. S. (2018). The bilingual language network: Differential involvement of anterior cingulate, basal ganglia and prefrontal cortex in preparation, monitoring, and execution. *Neuroimage*, *174*, 44–56.

Stocco, A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination. *Psychological review*, *117*(2), 541.

Stocco, A., Lebiere, C., O'Reilly, R. C., & Anderson, J. R. (2012). Distinct contributions of the caudate nucleus, rostral prefrontal cortex, and parietal cortex to the execution of instructed tasks. *Cognitive, affective, & behavioral neuroscience*, *12*(4), 611–628.

Stocco, A., & Prat, C. S. (2014). Bilingualism trains specific brain circuits involved in flexible rule selection and application. *Brain and language*, *137*, 50–61.

Stocco, A., Yamasaki, B., Natalenko, R., & Prat, C. S. (2014). Bilingual brain

training: A neurobiological framework of how bilingual experience improves executive function. *International Journal of Bilingualism*, *18*(1), 67–92.

Taatgen, N., & Lee, F. J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. *Human Factors*, *45*(1), 61–76.

Taatgen, N., Van Rijn, H., & Anderson, J. R. (2008). Time perception: Beyond simple interval estimation. *Department of Psychology*, 60.

United States Census Bureau. (2013, 08). *Language use in the united states: 2011*. (data retrieved from `https://www.census.gov/prod/2013pubs/acs-22.pdf`)

A Execution productions in monolingual and bilingual models

Monolingual

```
(p update-scratchpad-x
        =goal>
                isa phase
                step execution-x
        =retrieval>
                isa arithmetic-fact
                result =ans
        =imaginal>
                isa ritl-result
==>
        *imaginal>
                isa ritl-result
                x =ans
        =goal>
                isa phase
                step retrieve-task-y
)

(p retrieve-task-y
        ?imaginal>
        state free
        ?retrieval>
                state free
        =imaginal>
                isa ritl-result
                task1 =first
                task2 =second
                task3 =third
        =goal>
                isa phase
                step retrieve-task-y
==>
        =imaginal>
        =goal>
                isa phase
                step execution-y
        +retrieval>
                isa ritl-task
                kind ritl-task
                task1 =first
                task2 =second
                task3 =third
)

(p calculate-y
        =goal>
                isa phase
        step execution-y
        =imaginal>
                isa  ritl-result
                y nil
        =visual>
                isa  ritl-inputs
                y  =y
        =retrieval>
                isa  ritl-task
                task2 =second
==>
        =goal>
        =visual>
        *imaginal>
                isa ritl-result
                task =second
        +retrieval>
                isa operation
                task =second
                type unary
)
```

Bilingual

```
(p update-scratchpad-x-start-y
        ?retrieval>
                state free
        =goal>
                isa phase
                step execution-x
        =retrieval>
                isa arithmetic-fact
                result =ans
        =imaginal>
                isa  ritl-task
                task2 =second
==>
        *imaginal>
                isa ritl-result
                x  =ans
        +retrieval>
                isa  operation
                task =second
                type unary
        =goal>
                isa phase
                step execution-y
)
```

Monolingual

```
(p update-scratchpad-y
        =goal>
                isa phase
                step update-scratchpad-y
        =retrieval>
                isa arithmetic-fact
                result =ans
        =imaginal>
                isa ritl-result
                y nil
                task1 =first
                task2 =second
                task3 =third
==>
        *imaginal>
                isa ritl-result
                y =ans
        =goal>
                isa phase
                step retrieve-task-binary
)

(p retrieve-task-binary
        ?imaginal>
                state free
        =imaginal>
                isa ritl-result
                task1 =first
                task2 =second
                task3 =third
        =goal>
                isa phase
                step retrieve-task-binary
        ?retrieval>
                state free
==>
        =imaginal>
        +retrieval>
                isa ritl-task
                kind ritl-task
                task1 =first
                task2 =second
                task3 =third
        =goal>
                isa phase
                step execution-binary
)

(p calculate-binary
        =goal>
                isa phase
                step execution-binary
        =imaginal>
                isa ritl-result
                result nil
        =retrieval>
                isa ritl-task
                kind ritl-task
                task3 =third
==>
        *imaginal>
                isa ritl-result
                task =third
        =goal>
        +retrieval>
                isa operation
                task =third
                type binary
)
```

Bilingual

```
(p update-scratchpad-y-start-binary
        ?retrieval>
                state free
        =goal>
                isa phase
                step update-scratchpad-y
        =retrieval>
                isa arithmetic-fact
                result =ans
        =imaginal>
                isa ritl-result
                task3 =third
                y nil
==>
        *imaginal>
                isa ritl-result
                y =ans
        +goal>
                isa phase
                step execution-binary
        +retrieval>
                isa operation
                task =third
                type binary
)
```

B Afterword

I would like to thank dr. Jelmer Borst for his support and helpful commentary during research and writing. I'd like to thank dr. Andrea Stocco for his guidance during this project. Special thanks go to all the people at the University of Washington's Cognition & Cortical Dynamics Laboratory for their warm welcome and support during my stay.