

2017

## Exploring the Time-Based Resource-Sharing Model of Working Memory Through Computational Modeling

Joseph Glavan  
*Wright State University*

Follow this and additional works at: [https://corescholar.libraries.wright.edu/etd\\_all](https://corescholar.libraries.wright.edu/etd_all)



Part of the [Industrial and Organizational Psychology Commons](#)

---

### Repository Citation

Glavan, Joseph, "Exploring the Time-Based Resource-Sharing Model of Working Memory Through Computational Modeling" (2017). *Browse all Theses and Dissertations*. 1741.  
[https://corescholar.libraries.wright.edu/etd\\_all/1741](https://corescholar.libraries.wright.edu/etd_all/1741)

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

**EXPLORING THE TIME-BASED  
RESOURCE-SHARING MODEL OF WORKING  
MEMORY THROUGH COMPUTATIONAL  
MODELING**

A thesis submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science

By

JOSEPH GLAVAN  
B.S., Rensselaer Polytechnic Institute, 2013

2017  
Wright State University

WRIGHT STATE UNIVERSITY  
GRADUATE SCHOOL

APRIL 5, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Joseph Glavan ENTITLED Exploration of the Time-Based Resource-Sharing Model of Working Memory Through Computational Modeling BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

---

Joseph Houpt, Ph.D.  
Thesis Director

---

Scott Watamaniuk, Ph.D.  
Graduate Program Director

---

Debra Steele-Johnson, Ph.D.  
Chair, Department of Psychology

Committee on  
Final Examination

---

Ion Juvina, Ph.D.

---

Christopher Myers, Ph.D.

---

Robert E. W. Fyffe, Ph.D.  
Vice President of Research and Dean of  
the Graduate School

## Abstract

Glavan, Joseph. M.S., Department of Psychology, Wright State University, 2017. Exploration of the Time-Based Resource-Sharing Model of Working Memory Through Computational Modeling.

Working memory is the fundamental component of cognition that allows us to temporarily maintain information needed for concurrent processing. An existing theory from the literature, the time-based resource-sharing (TBRS) model, posits that working memory is a serial, rapidly switching, attentional refreshing mechanism. While others have sought previously to formalize the TBRS model into a computational process model, I go further, using ACT-R to model the influence of working memory on an entire task from end to end. I leverage ACT-R's existing base-level learning mechanism, typically used to model recency and frequency effects in long-term memory, to enact the attentional refreshing and temporal decay central to TBRS. I also use a novel combination of existing inhibition and association theories to implement a functional list representation. The model replicates trends in human memory spans and response times across six experimental conditions from a previously published study. These efforts reveal that areas not traditionally associated with working memory research directly, particularly item representation and response strategy, are necessary assumptions of any such process model despite being underspecified in TBRS and other theories. I discuss future experiments to further constrain these ancillary assumptions and conclude by proposing various directions for expanding the model in subsequent work.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Working Memory Capacity . . . . .	3
1.2	Existing Models of Working Memory . . . . .	6
1.3	Connection to Serial Memory . . . . .	11
1.4	Time-Based Resource-Sharing . . . . .	17
1.4.1	Predictions of TBRS . . . . .	18
1.4.2	Further Investigations of TBRS Assumptions . . . . .	21
1.5	Overview of ACT-R . . . . .	23
1.6	Compatibility of ACT-R and TBRS . . . . .	26
<b>2</b>	<b>Description of the Task</b>	<b>28</b>
2.1	Human Results . . . . .	30
<b>3</b>	<b>Description of the Model</b>	<b>33</b>
3.1	Declarative Memory . . . . .	33
3.2	Non-Declarative Learning . . . . .	40

3.3	Production Rules and Model Behavior . . . . .	43
<b>4</b>	<b>Parameters and Simulation</b>	<b>54</b>
4.1	Fixed Parameters . . . . .	54
4.2	Free Parameters . . . . .	57
4.3	Model Simulation . . . . .	59
<b>5</b>	<b>Results</b>	<b>61</b>
5.1	Model Evaluation Metrics . . . . .	61
5.2	Determining Parameterizations of Interest . . . . .	64
5.3	Parametric Effects on Components of Model Misfit . . . . .	90
5.3.1	Span . . . . .	90
5.3.2	RT . . . . .	91
5.3.3	TPT . . . . .	91
5.3.4	Accuracy . . . . .	91
5.3.5	Slope . . . . .	100
5.3.6	RMSE . . . . .	100
<b>6</b>	<b>Discussion</b>	<b>102</b>
6.1	Best Fitting Parameter Sets Based on Each Cumulative Metric . . . . .	102

6.2	Parametric Effects on Model Behavior . . . . .	104
6.3	Recommendations for Future Parameterization . . . . .	107
6.4	Shortcomings and Assumptions of the Model . . . . .	109
6.4.1	Reward and Utility Learning . . . . .	109
6.4.2	Metacognition and Feedback . . . . .	110
6.4.3	Declarative Long-Term Memory . . . . .	111
6.4.4	Serial Memory . . . . .	112
6.4.5	Other Methodological Restrictions . . . . .	115
6.5	Future Versions of the Model . . . . .	116
6.5.1	Refreshing Strategies . . . . .	116
6.5.2	Individual Differences and the Fan Mechanism . . . . .	119
6.5.3	Temporal Decay Versus Representation-Based Interference . . . . .	121
6.5.4	Non-Attentional Mechanisms of Maintenance . . . . .	122
6.5.5	Adaptive Maintenance . . . . .	122
6.6	Summary of Predictions . . . . .	123
6.7	General Impact . . . . .	126
<b>7</b>	<b>Concluding Remarks</b>	<b>128</b>
	<b>REFERENCES</b>	<b>139</b>

## List of Figures

1.1	ACT-R Cognitive Architecture . . . . .	25
2.1	Trial Timeline . . . . .	29
3.1	Model Organization . . . . .	44
3.2	Perception Subroutine . . . . .	44
3.3	New-List Subroutine . . . . .	45
3.4	Target-related Subroutine . . . . .	47
3.5	Distractor-related Subroutine (Parity) . . . . .	49
3.6	Target-related Subroutine (Location) . . . . .	50
3.7	Recall Subroutine . . . . .	51
3.8	Maintenance Subroutine . . . . .	53
5.1	Slope Error Example . . . . .	63
5.2	Total Error Functions . . . . .	66
5.3	Error Space (full) . . . . .	67

5.4	Error Space (subset) . . . . .	68
5.5	Total Error Functions (recolored) . . . . .	69
5.6	Total Error Cluster Parameter PDFs . . . . .	70
5.7	Total Error Cluster Parameter Frequencies . . . . .	72
5.8	Mean span of $\theta_0$ compared to the human data . . . . .	74
5.9	Mean RT of $\theta_0$ compared to the human data . . . . .	75
5.10	Mean TPT of $\theta_0$ compared to the human data . . . . .	76
5.11	Mean accuracy of $\theta_0$ compared to the human data . . . . .	77
5.12	Span/CL slope of $\theta_0$ compared to the human data . . . . .	78
5.13	Mean span of $\theta_\varphi$ compared to the human data . . . . .	79
5.14	Mean RT of $\theta_\varphi$ compared to the human data . . . . .	80
5.15	Mean TPT of $\theta_\varphi$ compared to the human data . . . . .	81
5.16	Mean accuracy of $\theta_\varphi$ compared to the human data . . . . .	82
5.17	Span/CL slope of $\theta_\varphi$ compared to the human data . . . . .	83
5.18	Mean span of $\theta_{RMSE}$ compared to the human data . . . . .	84
5.19	Mean RT of $\theta_{RMSE}$ compared to the human data . . . . .	85
5.20	Mean TPT of $\theta_{RMSE}$ compared to the human data . . . . .	86
5.21	Mean accuracy of $\theta_{RMSE}$ compared to the human data . . . . .	87

5.22	Span/CL slope of $\theta_{RMSE}$ compared to the human data . . . . .	88
5.23	Interaction of Inhibition, Similarity, and BLC on Span Error . . . . .	92
5.24	Interaction of Inhibition, Similarity, and BLC on Span Error . . . . .	93
5.25	Interaction of Inhibition, Similarity, and BLC on Span Error . . . . .	94
5.26	Interaction of Inhibition, Similarity, and BLC on Span Error . . . . .	95
5.27	Interaction of Inhibition, Similarity, and BLC on Span Error . . . . .	96
5.28	Interaction of Retrieval Latency Scaling Factor and BLC on RT Error . . . . .	97
5.29	Effect of Retrieval Latency Scaling Factor on TPT Error . . . . .	98
5.30	Effect of Retrieval Latency Scaling Factor on Accuracy Error . . . . .	99
5.31	Interaction of Inhibition and BLC on RMSE misfit . . . . .	101
6.1	Simulation of Anderson, Bothell, Lebiere, and Matessa (1998) . . . . .	118

## List of Tables

2.1	Existing Human Data . . . . .	31
3.1	Declarative Memory Chunk Examples . . . . .	34
3.2	Model Parameters . . . . .	36
5.1	Parameter values associated with best model fits . . . . .	73
5.2	Misfit measure correlations . . . . .	73
5.3	$F$ -tests . . . . .	89

## Acknowledgment

Many people supported my efforts with this thesis, and I am grateful to each of them, particularly those named below.

I want to first thank my advisor, Dr. Joseph Houpt, for his guidance and support throughout my time in graduate school. Dr. Houpt met with me frequently to ensure I was on track and to make this thesis as rigorous as it could be. I have learned and continue to learn a lot from you.

I want to thank my remaining committee members, Dr. Ion Juvina and Dr. Christopher Myers, for their mentorship. A major share of this work required developing a new ACT-R model. I owe Dr. Juvina a great deal of gratitude for his advice and suggestions during the project and because it was in his class that I first learned to use the ACT-R architecture. During an internship with the Repperger program I worked closely with Dr. Myers on applying computational modeling to training research, and it was during these advanced studies that I gained the intimate knowledge of ACT-R and similar architectures needed to make this thesis possible.

I want to thank Dr. Pierre Barrouillet and Dr. Valérie Camos for sharing data and ideas with me. From our first meeting, they have been very approachable and supportive of my work.

I want to thank Dr. Jack Harris for help with MindModeling@Home. Dr. Harris was always available to help diagnose bugs between my code and the MindModeling interface, which was vital to the success of my thesis.

I am very grateful to my friends and colleagues who have supported me in their own ways over the past few years.

Lastly, I want to thank my parents, whose love and support encourages me to be my best in life as well as in graduate school.

This work was supported in part by AFOSR Grant FA9550-13-1-0087 awarded to Dr. Houpt.

# 1 Introduction

Working memory (WM) may be the paramount element of cognition. Drawing its name from the analogy of the mind as a computer, its significance has been recognized from the very beginning of the cognitive revolution (Miller, 1956). Just as random-access memory (RAM) is vital to the efficient operation of a computer system, it has been argued that a similar kind of immediately accessible memory or workspace is necessary for a cognitive system to function. Working memory has received considerable additional attention in the cognitive psychology literature lately for its strength as a predictor of general fluid intelligence (Chow & Conway, 2015; Conway et al., 2005). Just as intelligence can vary widely across people, individual differences are also commonly observed in measures of WM. Latent variable analyses repeatedly demonstrate covariance between these two constructs as high as .60 (Conway, Cowan, Bunting, Theriault, & Minkoff, 2002).

Working memory is defined as the ability to temporarily maintain representations of information relevant to the immediate environment or necessary for the accomplishment of goals (Baddeley & Logie, 1999). It is commonly thought of as a short-term storage for intermediate processing results, like the sums and products in an algebra problem (Anderson, 2005). It may also serve as a buffer for the storage and retrieval of information from long-term memory (LTM; Atkinson & Shiffrin, 1968; Baddeley, 2000). Whereas information held in LTM is relatively permanent with respect to time, WM is temporary, retaining information on the magnitude of seconds. The name short-term memory (STM) can also be used to describe any such nonpermanent

information, but the term WM is generally reserved for only those mnemonic tasks that involve a concurrent processing activity (Diamond, 2013). It will be fundamental to this thesis however that recall never takes place in a vacuum: all mnemonic activity, whether in or out of the laboratory, proceeds under some degree of concurrent processing demand (i.e. cognitive load; Barrouillet, Bernardin, & Camos, 2004). For this reason, any such division between WM and STM may be a false dichotomy. I will introduce a model that advocates for such a WM/STM spectrum.

With its roots in the cognitive revolution as well, computational process modeling has grown in popularity as a tool for formally specifying theories. Computational process models go beyond verbal theories by instantiating the theory as a set of algorithms. By specifying the process in computer code, it quickly becomes apparent when hidden assumptions have been overlooked because the simulation simply will not run. Additionally, there is no ambiguity in computer code like there is in written language. This benefit has been used to unite related cognitive theories and classes of models into cognitive architectures, overarching systems of shared assumptions derived from empirical work and organized into compatible libraries of computer programs. One such cognitive architecture, Adaptive Control of Thought – Rational (ACT-R; Anderson, 2007; Anderson et al., 2004) will be a focus of this thesis.

Computational process modeling provides a stricter method for composing cognitive theories and allows for directly testing the theories because the models are computer programs capable of execution and generation of simulated data. Hypotheses are built into a model and one determines if the effects observed in human data can actually be produced by the assumptions of the model. This approach is even more useful in research situations where there is more than one plausible mechanism proposed to explain the data. Computational modeling can be used to pit these hypotheses against each other, head-to-head, while holding everything else constant.

Beyond which model explains the observed human data better, modeling can even be used to ascertain where indistinguishable theories diverge. Through simulation, it is possible to determine the full range of results a model is capable of producing, and the conditions where such theories no longer overlap spotlight the critical experiment that should be conducted next.

The greatest advantage of computational cognitive modeling, in addition to those outlined above, is that it does not require closed-form solutions, unlike other styles of mathematical modeling and verbal theorizing. Behaviors arising from many complex and dynamic interactions can be chaotic and impossible to fully predict with these models but not for computational ones. These types of phenomena are fully exemplified by real-world behavior, and understanding the behavior of everyday people in the real world is the ultimate goal of psychology. Therefore computational process models are crucial if we are to understand how the mechanisms underlying WM regulate human interactions with the environment in real-life situations. In this thesis, I divulge the development of such a model, capable of performing every aspect of a WM task, not just the memory component. It builds upon an existing framework for end-to-end process modeling, laying the foundation for future application to other areas of research. I will discuss how it integrates multiple elements from contemporary theories of WM, STM, and LTM; specifically, the areas of executive control, serial memory, and episodic memory. This effort reveals that WM may not be a simple buffer for storage, as in the RAM analogy, but an active process susceptible to the strains of competing attentional demands.

## **1.1 Working Memory Capacity**

The standard quantification associated with WM is its capacity (WMC), the amount of information maintained over a given period of time and under a given

cognitive load. Chow and Conway (2015) outline three unique sources of variance that contribute to individual differences in WMC, the first being the number of items able to be kept active simultaneously within the focus of attention. Cowan (2001) suggests this “pure capacity” to be around four items. In addition to the raw number of items simultaneously maintained, the quality, or resolution, of these items may be used to characterize WMC; however, the distinctiveness of its contribution to fluid intelligence remains unresolved (Chow & Conway, 2015; Fukuda, Vogel, Mayr, & Awh, 2010). Beyond the scope of attention, the top-down control of attention provides another source of variance in WMC. Reflecting one’s ability to disregard distractions, to concentrate on goal-relevant information, and to efficiently divert focus between tasks; this construct connects WMC with domain-general everyday cognition (see Conway et al. (2005) for a review of references to studies ranging from high-level activities such as reading, reasoning, and learning to low-level behaviors such as exogenous attentional capture and proactive interference).

The third operationalization of WMC, successful control of attention, is frequently measured using the complex span paradigm. In a complex span task, to-be-remembered items (denoted as “targets”) are presented in alternation with some number of “distractors.” Subjects are instructed to remember the targets while processing the distractors until they are prompted to recall the targets, often in the correct order and sometimes with the option to skip serial positions for which they cannot recall the correct target. Typically this routine is repeated with progressively more target-distractor series until the subject fails to meet some recall criterion, such as correctly recalling some percentage of the list. For example, in the reading span task (Daneman & Carpenter, 1980), subjects are presented with a series of  $n$  sentences, which they are instructed to read aloud, memorizing the final word of each sentence. After the series has been presented, subjects are prompted to recall the list of final words. If they continue to meet criterion, a series with  $n + 1$  sentences is presented. Other popular

complex span tasks include the operation span task (Turner & Engle, 1989), where subjects are required to evaluate arithmetic expressions between memorizing targets, and the counting span task (Case, Kurland, & Goldberg, 1982), where subjects count the number of objects in a display and then memorize the tally. The score obtained from a complex span task, indicative of the subject's WMC, is called a memory "span," and may be calculated in various ways, depending on how the experimenter wishes to treat the completeness, order, etc. of recall. It is often the average length of each target-distractor series the subject completed. Based on a meta-analysis of Kane et al. (2004), Conway et al. (2005) recommend using partial-credit scoring, where the score of a particular series is the percentage of that series correctly recalled, as opposed to all-or-nothing scoring. They demonstrate that, while both are still relatively high, the partial-credit scoring method exhibits greater within-task consistency. Because all-or-nothing scoring is a more coarse measure, it necessarily will exhibit greater within-individual variance, meaning the same person may score differently upon repeated administrations of the complex span task.

The quick pace and attentional demand of the distractors in a complex span task are intended to interfere with the subject's ability to employ a number of memory-aiding cognitive strategies (e.g. mnemonics, grouping/chunking, mental imagery, etc.). Ideally, the intense cognitive load of these tasks (e.g., sentence comprehension, equation solving, counting, etc.) entirely prevents any rehearsal from taking place, allowing the experimenter to measure the raw amount of information that survives from encoding to probed recall. However, because the traditional complex span task is self-paced, subjects, deliberately or not, may adapt to the load by pausing slightly before responding (Barrouillet et al., 2004). During such pauses they can engage in covert maintenance processes (Cowan, 1992; Cowan et al., 1994), inflating their effective WMC. Barrouillet et al. (2004) recognized that it may be impossible to experimentally eliminate all covert rehearsals in a traditional complex span task

because of variance in individuals' processing abilities. To remedy this, they proposed reducing the complexity of processing demanded by distractors (e.g. from equation solving to single-digit addition or from sentence comprehension to single-symbol articulation) and strictly controlling the onset and offset of targets and distractors. While the time available for maintenance cannot be eliminated, if the number of processing steps (e.g. retrievals) for a given distractor can be determined, then this time can at least be controlled. I refer to this general paradigm as continuous span tasks, adapted from the name Barrouillet et al. (2004) give to their second experiment (continuous operation span task).

## 1.2 Existing Models of Working Memory

Probably the most influential model of WM is the multicomponent model (Baddeley, 1986; Baddeley & Hitch, 1974; Baddeley & Logie, 1999). Baddeley and Hitch (1974) conducted a series of experiments where subjects were required to memorize a list of digits and then complete a verbal reasoning task. They observed that the size of the memory load had little effect on accuracy in the reasoning task while increasing response times. They also observed greater interference when the memory items were phonologically similar to the reasoning items, although concurrent articulatory suppression only had a minor effect on reasoning times. They concluded that while verbal reasoning must share some resources with STM, the sharing is likely limited to the degree of overlap between representations in the two tasks because they failed to find the sort of catastrophic interference they expected a single capacity model to predict. Accordingly, they proposed a model in which processing and maintenance are handled by two separate but interconnected systems. The phonological loop, operating on a verbal code, implements storage through articulatory rehearsal of items. This slave system is managed by a central executive system, which is responsible for processing. While supervision of the phonological loop is necessary for coordinating storage, it does

not exhaustively tax the central executive, explaining the relatively minor interference observed by Baddeley and Hitch (1974). Further work on the multicomponent model has emphasized dissociation between verbal STM and visuospatial STM, leading to the addition of another slave system to the model, the visuospatial sketchpad.

Contemporary to the multicomponent model is a capacity-sharing model. The trade-off model of WM (Case et al., 1982) posits the existence of a singular pool of strictly limited resources called *M space*. This resource space is required for both processing and storage. Case et al. (1982) used a counting span task to assess the size of subjects' M spaces, where subjects had to count the green dots (amongst distractors) on a series of cards and remember the sequence of counts. In adults, they manipulated counting speed by forcing one group to count using a pseudo number scheme. They observed a linear relationship between working memory span and counting speed and concluded that the slower counting speed in the pseudo number condition reflected worse processing efficiency than with normal numbers. This less efficient processing required more space in M space, which left less space available for storage and caused lower spans to be observed. Likewise, phonologically similar items require more resources to distinguish them from each other and cause lower spans.

Towse and Hitch (1995) interpreted Case et al.'s results differently. They argued that slower counting caused lower spans not because this inefficient processing stole additional resources (relative to the normal numbers condition) away from storage activities but because of temporal decay. They proposed that subjects switch between storage tasks and processing tasks and that while subjects are performing processing tasks, memory items decay as a function of time. Slower counting speeds contribute to longer periods between storage activities, thus allowing more decay to take place. Towse and Hitch evaluated this hypothesis using an alternative manipulation to the counting span task. In one condition of the task, the first card in the sequence took a

long time to count while the last card could be counted (relatively) quickly. In the second condition, the sequence was flipped so that the first card took less time to count and the last card took more time to count. Because storage is not required until after the first card has been counted, the two conditions have distinctly different retention periods but equal overall difficulty. Working memory spans were observed to negatively correlate with retention period length, and because the same series of cards were used in both conditions (only the order was manipulated), the authors concluded that temporal decay, not limited processing resources, was responsible for limited working memory capacity. These results were later found in other complex span tasks (e.g., reading span, operation span; Towse, Hitch, & Hutton, 1998) and taken as further evidence for the task-switching model.

The capacity-sharing hypothesis did not disappear with the rise of the task-switching model. Anderson, Reder, and Lebiere (1996) proposed an addition to the ACT-R theory, applying findings from research on the fan effect (Anderson, 1976; Pirolli & Anderson, 1985) to WM. According to the fan effect, the more items a cue is associated with, the less reliable it is at predicting any of them. In this same way, they argued that activation from a given context must be divided among all the items in the context. Contexts with fewer elements are able to spread more activation to each item, making them more accessible. Anderson et al. (1996) conducted an experiment where subjects had to memorize a list of digits and then solve algebra problems. In some of the experimental conditions, subjects were required to substitute one or two of the digits from the list into the algebra problem. They found effects of the number of symbols in the equations and the length of the memorized list. A single capacity model, which modeled the activation of an item as  $1/(d + s)$ , where  $d$  is the number of symbols in the equation and  $s$  is the number of items in the list, fit the human data better than a separate capacity model, which modeled the activation of equation elements as  $1/d$  and the activation of list items as  $1/s$ . Interestingly, their model

did not utilize temporal decay or rehearsal, significant components of later ACT-R models.

More recently, Barrouillet and Camos (2001), in addition to describing various other possible confounds in Towse and Hitch's (1995) counting span task, pointed out a significant flaw in the task-switching conclusion. While it is true that retention periods are different in each condition because storage activities do not begin until after the first card has been counted, the cognitive load concurrent with the retention period is also different because it too does not begin until after the first card has been counted. Thus Towse and Hitch (1995) may not have truly ruled out the trade-off model and other resource-sharing accounts of working memory. To better control temporal dynamics so as to better study the effects of processing difficulty on working memory performance, Barrouillet et al. (2004) introduced the aforementioned continuous span paradigm, which contributed substantially to the development of their model.

The time-based resource-sharing (TBRS) model of WM (Barrouillet et al., 2004; Barrouillet, Bernardin, Portrat, Vergauwe, & Camos, 2007) was introduced to account for the shortcomings of limited resource trade-off models (Case et al., 1982) and task-switching models (Towse & Hitch, 1995) when task time and difficulty are tightly controlled. It is a member of a class of models that assume a central bottleneck (Pashler, 1984). The TBRS model proposes that active maintenance of items is achieved through rapid, and possibly covert, switching of attention between processing and maintenance. The major claim of TBRS is that the ratio of time devoted to maintenance relative to the time spent processing is the main factor responsible for determining the amount of information able to be retained, rather than the raw time allocated to either alone. Contrary to trade-off models, which suggest task difficulty constrains WMC, or other capacity-sharing models (Anderson et al., 1996), which suggest the number of items to be retained directly limit WMC, TBRS predicts

that (potentially vastly different) tasks with varying levels of difficulty or number of elements can produce equal WM spans if they are equated in processing time relative to the total time of the task (Barrouillet et al., 2007). Likewise, the time between targets, the hypothesized determinant of task-switching models, was shown by Barrouillet et al. (2004) to only affect WM spans when total task time was kept constant; when the ratio of the two is controlled, this effect disappears. The conclusion of the TBRS theory, that limited attentional resources are shared on a temporal, rather than global, basis (hence the name time-based resource-sharing), reveals that WM is a dynamic sequence of items trading their time in the focus of attention rather than the static partitioning of a limited capacity space.

Unlike task-switching models (e.g., Towse & Hitch, 1995; Towse et al., 1998), which only allow maintenance to take place between sets of distractors, the TBRS model allows maintenance to take place whenever the central bottleneck is free. This includes delay intervals, within sets of distractors, and even during the processing of a distractor if it is waiting on peripheral systems (Barrouillet & Camos, 2015). Refreshing, retrieval, and other sources of attentional capture are not instantaneous; they take time to fulfill. Although maintenance can only be executed serially due to the central bottleneck, temporal decay affects all memory traces simultaneously. This exchange of rates ensures that only a limited number of items may be kept accessible by the cognitive system, a hallmark of WM.

Oberauer and Lewandowsky (2011), themselves critics of some of TBRS's base assumptions, developed a connectionist model, TBRS\*, to evaluate whether such a rapidly switching serial mechanism could produce the observed effects on WM span. Their model borrows heavily from Burgess and Hitch (2006; discussed in more detail in the next section) and refreshes targets by strengthening associations between items and positional markers (represented as a group of nodes) through Hebbian learning

until some criterion strength is achieved or maintenance is interrupted by a processing episode.

Although TBRS\* successfully reproduces the effects predicted by the verbal TBRS theory and observed in empirical studies, yielding proof-of-concept support for time-based resource-sharing, it suffers from one significant defect. TBRS\* does not fully model the processing of distractors; rather it samples a random duration from the distribution of response times determined post-hoc by experimental observations. Refreshing targets is then delayed for the interval sampled. A comprehensive model of WM should not abstract away distractor processing when the critical aspect of the TBRS theory is that the interaction of maintenance and processing is what affects WMC. Particularly when studying real-world tasks where processing is more complex and more variable, it is paramount that we have an integrated model of WM that is able to fully simulate the person-environment system. My model does this, conducting simultaneous maintenance of targets and processing of distractors to predict all measures of performance.

### **1.3 Connection to Serial Memory**

In many WM experiments, including the continuous span task I use to validate my model, subjects are required to recall the targets presented and in the order in which they are presented. Thus, WM is not responsible for merely maintaining multiple items individually but as a list of items. List memory, or serial memory, is its own area of study and seemingly overlooked in theories of WM (cf. Farrell, 2012). However, as noted by Oberauer and Lewandowsky (2011) when developing TBRS\*, computational process modeling makes apparent the need to explicitly specify the representation of items and how their order is encoded. To satisfy this requisite, I next provide an overview of contemporary theories and models of serial recall.

Three techniques for encoding order in list or serial memory are commonly considered in the literature: positional coding, primacy gradients, and chaining. With positional coding, an item is tagged with its ordinal position (e.g., first, second, third, etc.) when it is first encoded at the time of presentation. The cognitive system then loops through the ordinal tags, using them as cues for retrieving the associated target. Similarity gradients or random noise can be used to introduce positional errors in recall. This is the approach taken by Burgess and Hitch (2006; and therefore Oberauer and Lewandowsky, 2011) as well as Huss and Byrne's (2003) ACT-R model of the phonological loop (likely following the example of Anderson et al., 1998). Primacy gradient models (e.g., Farrell, 2012; Page & Norris, 1998) assume some decreasing function of activation across list items, which may originate from association with the beginning of the list or derived from the number of items already in the list at the time of encoding. The item with the highest activation is retrieved, and to avoid repeatedly retrieving the same item, already retrieved items are suppressed until the full list has been iterated. Chaining models (e.g., Kieras, Meyer, Mueller, & Seymour, 1999; Lewandowsky & Murdock, 1989; Solway, Murdock, & Kahana, 2012) assume that adjacent items in a series are directly associated with each other, akin to a linked list in computer science. Retrieval of one item cues retrieval of the next item, and list structure emerges from local, inter-item connections rather than from assuming some global order.

While each of the methods for encoding order introduced above has their merits, none are without a fatal flaw. Positional coding can be argued to simply move the problem of order from items to positions. In order to know which position with which to encode a new item, the total number of items currently in the list must be maintained in addition to the list itself. The list could instead be iterated (read: rehearsed in full) when a new item is presented to determine its appropriate position for encoding, but this would require a significant amount of time, during which the

new item would still need to be stored somewhere. Additionally, if an item's position is used as the cue to retrieve the next position in the list, then positional coding reduces to chaining. Furthermore, because the positions are chained rather than the items themselves, additional retrievals must be made to get the item from its positional code.

Primacy models may face criticism for using the well-known primacy effect observed in serial recall data as a fundamental assumption rather than a product of its mechanisms. The nature of its suppression mechanism is also underspecified — why does such a mechanism know to relinquish its suppression only after rehearsing the final item of the list, which may change throughout the experiment? Oberauer and Lewandowsky (2011: Appendix A) provide an analysis of the shortcomings of primacy models in the context of a TBRS model. Primacy models reconstruct their activation gradient as items are rehearsed and require rehearsing the full list before the old gradient is updated. Partial list rehearsals, which the TBRS theory proposes to take place during brief pauses in processing, cause uneven allocations of activation across the list, polluting the gradient and contaminating the encoded order. Moreover, the distractor items in a complex span task are intended to interrupt maintenance activities, making it very unlikely that there will be an idle interval long enough to rehearse the list in its entirety (Oberauer & Lewandowsky, 2011).

Simple chaining models suffer the most obvious faults: like any chain, they are only as strong as their weakest link. If the model fails to retrieve an item, then it has no cue with which to retrieve the next item. The remaining items that follow in the list may still have activation and so may not be lost from memory per se, but they have become disconnected from the list and thus are no longer accessible, at least for this iteration. The next rehearsal loop through the list may successfully retrieve the previously failed item, but the items that follow did not get rehearsed previously and

now may have decayed out of memory. Lists that allow repeated items may also cause trouble for chaining models because these items would become linked to multiple items, making the correct next item ambiguous. Models that allow associations among more than two items (becoming more like a network than a simple chain) and/or unique identities for each item can overcome these challenges; however, such models may still have difficulty accounting for various error patterns in serial recall (Henson, Norris, Page, & Baddeley, 1996).

Burgess and Hitch's (2006) connectionist model of serial recall, which Oberauer and Lewandowsky (2011) adopt as their engine for serial representation in TBRs\*, uses positional coding. It sidesteps the problem of positional order by presupposing a continuously changing context signal instantiated by a population of nodes, possibly realized by neurons acting as temporal oscillators (Brown, Preece, & Hulme, 2000). When items are first encoded, their association to the context signal is strengthened. Rehearsal and recall are achieved by replaying the context signal, like a film, and retrieving the item most highly activated by the reestablished context signal. Under this framing, it is the association of an item with its position that is maintained, not the item itself. This idea reflects the binding hypothesis (Oberauer et al., 2007) of WM: items are not directly lost from memory; rather they lose their binding to the correct retrieval cue. I have not found a source that satisfactorily explains how the context signal is maintained so that it can be reinstated at a later time.

Anderson et al.'s (1998) model of list memory utilizes an alternative to Burgess and Hitch's conceptualization of positional coding that circumvents the problems previously described. Their ACT-R model directly encodes items with their ordinal position, which is justified because of the paradigm they chose to use. In their immediate serial recall experiments, subjects are visually presented with a number of boxes equal to the number of items in the upcoming list. Each item is presented

one at a time in its respective box. During recall, the cursor moves between boxes as the subject enters their responses. Because the boxes are always present on screen, subjects are heavily biased toward associating each item with its respective spatial location. However, this is not the case in typical complex span tasks where items are presented more temporally disparate and in the same location. For this reason, Anderson et al.'s method of positional coding may not generalize conclusively to all instances of serial recall.

More recently, researchers have begun to consider whether items are encoded with their temporal position in the experiment rather than with their ordinal position in the list. Unsworth and Engle (2006) propose that one of the factors contributing to individual differences in complex span task performance is subjects' ability to effectively use temporal-context as a cue for retrieval. They argue that while other cues (e.g., semantic, phonological, etc.) may facilitate rehearsal, complex span tasks are inherently episodic memory tasks, and the only statistically meaningful cue for an item in such a task is the temporal context in which it is encoded. They support their argument with error patterns in reading span and operation span tasks; however, their analysis is based on the classic interpretation of a complex span task — that it is simply an immediate serial recall task modified to prevent all rehearsal. They apply their account of temporal-contextual cues to retrieval at recall only; they do not consider any intra-task refreshing or rehearsal, covert or otherwise. This is problematic because their complex span tasks are self-paced, which allows the subject to insert pauses for rehearsal while processing the distractor task. For these reasons, their results alone may not be enough to substantiate their conclusions.

Fortunately, further support for temporal-contextual associations encoding target order can be found elsewhere in the literature. Looking at simple serial recall research, recall of target items tends to cluster around their correct position when all responses

are taken into account (Estes, 1972). This finding is commonly attributed to positional coding of items; however, such an evaluation confounds the temporal position of items, as recalled items also tend to cluster around the position of the previously recalled item (Solway et al., 2012). Recalling an item in its correct position necessarily implies that it is recalled in the next position from the previously (correctly) recalled item. Conditioning their analyses on the item recalled after a first-order transposition error, Solway et al. (2012) found that anticipation errors, recalling an item in an earlier position, were most commonly followed by another anticipation error — most often the item that follows the transposed item. They showed that a forward chaining model better supported their results than Burgess and Hitch’s (2006) positional coding model. Contrasting these findings, Farrell, Hurlstone, and Lewandowsky (2013) repeated Solway et al.’s (2012) analyses on 19 other previously conducted serial recall experiments and found that anticipation errors were more often followed by filling-in the skipped over item rather than consecutive anticipation errors. They note a number of differences in the ways the analyzed studies were conducted, any of which may explain the discrepancy in conclusions. Solway et al. (2012) examined studies where subjects learned much longer lists over multiple trials and were allowed to skip items. Unlike Farrell et al.’s (2013) reviewed studies, which used letters and digits as the targets, they used words as the to-be-remembered items, which may be more readily formed into phrases, an inherently chain-like structure. Farrell et al.’s (2013) findings are more applicable to complex span studies because they use similar list-lengths and instructions. More convincingly, they showed that Farrell’s (2012) model of short-term and episodic memory is able to reproduce the results from both papers, as well as intermediate studies.

Farrell’s (2012) model is driven by the notion that humans constantly parse their continuous experiences into temporally related, discrete episodes. Applied to serial recall tasks, it assumes that subjects spontaneously group proximate items into subsets.

These subsets are cued by a temporal context — similar to Unsworth and Engle’s (2006) account — and once retrieved can be used to access their constituent elements. Relative association to the first and last items in the subset encodes the order of items within a subset using a primacy gradient. This relative association may be derived from neurally-instantiated temporal oscillators (Brown et al., 2000). The temporal grouping model has been shown to account for a variety of findings common to simple span tasks and provides a unified theory of immediate free recall performance and immediate serial recall performance (Farrell, 2012; Spurgeon, Ward, Matthews, & Farrell, 2015). Interestingly, Farrell (2012) modeled the data from Unsworth and Engle (2006) and found that lower WMC could be explained by an increased likelihood to form smaller subsets, the errors in recall arising from more opportunities to retrieve the wrong group context.

Stepping back, it appears that an interesting cycle of conflict has emerged: Solway et al. (2012) suggest that chaining models produce temporal-grouping effects that positional models cannot, Farrell et al. (2013) demonstrate that a primacy gradient model is superior to chaining models, and Oberauer and Lewandowsky (2011) argue that a primacy gradient cannot be used to model TBRS but positional coding can be. To rectify this impasse, I propose a compromise that borrows from each approach and is explained in detail in the model description section. In brief, my model will encode each item with the temporal context during which they are presented and then use associative chaining between these encoded contexts to retrieve one item after another. Other mechanisms will produce effects similar to primacy and suppression.

## 1.4 Time-Based Resource-Sharing

Since the work of Barrouillet and Camos (2001), Barrouillet et al. (2004), and Barrouillet et al. (2007) refuted pure capacity-sharing models (e.g. Case et al., 1982)

and pure decay-based models (e.g. Towse & Hitch, 1995), TBRS has become a leading model of WM. With its expansion into domain-specific investigations of verbal and visuospatial WM, TBRS is stronger and as comprehensive as the well-known multicomponent model (Baddeley, 2000, 2012; Baddeley & Hitch, 1974; Camos & Barrouillet, 2014). For these reasons, I use TBRS as the core of my model.

TBRS is derived from three key assumptions (Barrouillet et al., 2004, 2007; Barrouillet & Camos, 2015). The first assumption is that both processing and maintenance require the same limited resource: attention. The second assumption is the existence of a central bottleneck or limited focus of attention. Only one item may occupy the central bottleneck at any given time. The third assumption is an interaction between active refreshing and passive decay. The memory trace of an item receives activation when it is brought into the focus of attention; meanwhile, the memory traces of all items not currently being refreshed experience temporal decay. The process of active maintenance may be explicit (e.g. articulatory rehearsal) or implicit (e.g., covert retrieval, attentional refreshing). The need to continuously refresh memory traces in order to prevent their loss by decay, when coupled with the limitations of a central bottleneck, implies that the working memory system includes a serially rapid-switching mechanism, constantly switching between processing and maintenance to balance the temporal needs of each.

### **1.4.1 Predictions of TBRS**

Further study of the TBRS model's specification yields a strong, quantifiable, prediction of working memory capacity. The odds that an item is still accessible at some point later are directly related to the amount of retention time spent in the focus of attention. Observable WMC is therefore a function of the collective time available for maintenance activities. Tasks that require executive control obstruct the central

bottleneck, reducing the time available for maintenance during the task period and placing a hard constraint called “cognitive load” on task-dependent WMC. When this obstruction time is decomposed into individual processing events, cognitive load can be formulated as

$$CL = \frac{\sum_{i=1}^N a_i}{T} \quad (1.1)$$

where  $a_i$  reflects the latency of the  $i^{th}$  process and  $T$  is the total task time (Barrouillet et al., 2004). If the number of processes  $N$  is known, or controlled as it is in a continuous span task, Equation 1.1 can be simplified by using mean processing times:

$$CL = \frac{\bar{a}N}{T} \quad (1.2)$$

Note that cognitive load is a dimensionless ratio of times ranging from 0 to 1. Using response times as approximations of processing times and using Equation 1.2, WMC has repeatedly been observed as a linear function of cognitive load (Barrouillet et al., 2004, 2007; Barrouillet, Portrat, & Camos, 2011):

$$S = k(1 - CL) \quad (1.3)$$

where  $S$  is the empirically observed span,  $CL$  is cognitive load, and  $k$  is the participant’s raw capacity that may vary based on qualities of the memory targets (e.g., word length, frequency, etc.). The parameter  $k$  is also the capacity of an individual that hypothetically would be observed when there is no cognitive load present (i.e. all executive processes serving maintenance). Such a situation might be considered an ideal simple span task (i.e. in the absence of higher-level retrieval strategies like chunking, narrative production, etc.). Furthermore, recognizing that  $T$  also can be

blocked into  $N$  bins, Equation 1.2 becomes

$$CL = \frac{\bar{a}N}{\tau N} = \frac{\bar{a}}{\tau} \quad (1.4)$$

the proportion of time spent processing to time available for processing. Rearranging and substituting into Equation 1.3 yields

$$\frac{S}{k} + \frac{\bar{a}}{\tau} = 1 \quad (1.5)$$

where  $0 < k$ ;  $0 \leq S \leq k$ ;  $0 < \tau$ ;  $0 \leq \bar{a} \leq \tau$

Equation 1.5 expresses a perfect, time-based, limited resource trade-off: the proportion of WMC available for maintenance and the proportion of time necessary to process the concurrent cognitive load must sum to one.

Continuous span tasks have been used to factorially manipulate the individual components of cognitive load (Equation 1.2), and appropriate trends in span have been observed that simultaneously violate predictions of trade-off and task switching models while supporting predictions of the TBRS model (Barrouillet et al., 2004, 2007). A meta-analysis of 14 different experimental conditions (Barrouillet et al., 2007; Barrouillet, Portrat, & Camos, 2011) involving a variety of executive processes such as updating, inhibition, response selection, and retrieval found that Equation 1.3 accounts for an impressive 98% of the variance observed in span scores. Interestingly, the  $k$  parameter, the raw WMC, is commonly found to be around just over 8, well within Miller’s (1956) magical number ( $7 \pm 2$ ). These strong empirical confirmations of the model’s quantitative predictions provide strong evidence for time-based resource-sharing.

### 1.4.2 Further Investigations of TBRS Assumptions

The predictions of the TBRS model are well supported, which is why the most recent research has focused on its premises. The first assumption, that processing and maintenance share a common resource, has received the most attention. Vergauwe, Barrouillet, and Camos (2010) showed that verbal and visuospatial processes, which the multicomponent model (Baddeley & Hitch, 1974) assumed to rely on separate resources (Baddeley & Logie, 1999) or separate systems (Baddeley, 2000), can still interfere with each other as if sharing a limited resource if temporal factors are carefully controlled using a continuous span task. However, verbal processing interfered even more with verbal storage than can be attributed to domain-general resource sharing, suggesting that there may exist an additional system contributing to verbal capacity (such as the multicomponent model's phonological loop; Barrouillet & Camos, 2010; Vergauwe et al., 2010). Further investigations (Camos, Mora, & Barrouillet, 2013; Mora & Camos, 2013) found that effects such as phonological similarity and word length only affect this additional capacity for verbal information. The portion of interference from domain-general executive processes did not vary across tasks, suggesting these attentional and non-attentional systems are separate. Camos, Mora, and Oberauer (2011) have even shown that humans can adaptively favor one system over the other depending on the attentional and phonological demands of the concurrent task. As the TBRS model has grown to accommodate these separate systems, Camos and Barrouillet (2014, see also Barrouillet & Camos, 2015) have found it increasingly useful to discuss TBRS in the terms of a cognitive architecture. While useful for organizing a growing theory, no such architecture has yet been computationally implemented. My model is the first step in this direction.

The second assumption of TBRS, that attention is applied within the limited scope of a bottleneck, and its implication that refreshing is a serial process has also

been challenged. Portrat and Lemaire (2014) used TBRs\* (Oberauer & Lewandowsky, 2011), to simulate data from Barrouillet et al. (2007). They found that the switching rate required for the model to fit the unpublished serial position curves of the data was 10 ms, a speed they noted was implausibly short. They further demonstrated how TBRs\* with a more reasonable switching rate but focus size of one item cannot produce a recency effect on the last item of a serial position curve because the last item does not get refreshed frequently enough. When the focus of attention was increased to simultaneously hold up to four items it had the dual benefits of reintroducing the recency effect and bringing the best-fitting switching rate up to 80 ms. That being said, it is worth noting that TBRs\* uses a different list representation and refreshing scheme than my model will use, and it remains to be seen whether Portrat and Lemaire’s conclusions are specific to TBRs\* or applicable to TBRs in general.

The third assumption of TBRs, that memory traces decay as a function of time, is at odds with another hypothesis in the literature. Representation-based interference (Lewandowsky & Oberauer, 2009) proposes that processing a piece of information immediately weakens the memory traces of previously stored information because of interference between the items’ episodic representations. Barrouillet, Portrat, Vergauwe, Diependaele, and Camos (2011) demonstrated that there is little evidence for representation-based interference when temporal factors are carefully controlled. Interference affects representations with greater variation equally as much as representations with a high degree of similarity under the same time-structure.

While Lewandowsky and Oberauer (2009) hypothesize that the actual duration of the processing episode does not affect memory strengths, they suggest that memory traces undergo an accumulating process of reconstruction between processing episodes. Essentially, faster processing would result in stronger memory traces because the time available for reconstruction is greater, not because traces undergo less decay.

To evaluate these dueling hypotheses, Barrouillet, De Paepe, and Langerock (2012) manipulated the processing time in a complex span task while keeping the time between processing episodes constant. They found further support for the temporal decay hypothesis when longer processing episodes elicited poorer recall performance.

## 1.5 Overview of ACT-R

To facilitate my goal of developing a computational model of TBRS that can be applied to real-world scenarios, such as driving (Salvucci, 2006) or human-machine teams (Demir et al., 2015), I have chosen to construct my model within the Adaptive Control of Thought – Rational (ACT-R) cognitive architecture (Anderson, 2007; Anderson et al., 2004). A cognitive architecture is both a theory of human cognition as well as a framework for developing cognitive models. ACT-R is implemented in the programming language LISP, which allows its models to be computationally evaluated. This is especially useful when developing large, complex and dynamic models. Many researchers worldwide support the architecture through empirical studies in multiple areas of cognitive science including memory (Anderson, 2007; Anderson et al., 1998; Lebiere & Lee, 2002), learning (Anderson, 2005; Anderson & Betz, 2001; Blessing & Anderson, 1996; Janssen & Gray, 2012; Thomson & Lebiere, 2013), language processing (Ball, 2004, 2013; Budiu & Anderson, 2004; Reitter, Keller, & Moore, 2011), perception and action (Cao, Qin, Zhao, & Shen, 2015; Halverson & Gunzelmann, 2011; Harrison & Trafton, 2010; Taatgen, Van Rijn, & Anderson, 2007; Tamborello & Byrne, 2006), problem solving (Altmann & Trafton, 2002; Guhe, Pease, & Smaill, 2009; Reitter, Juvina, Stocco, & Lebiere, 2010; Taatgen, Huss, Dickison, & Anderson, 2008), and decision-making (Dickison & Taatgen, 2007; Marewski & Mehlhorn, 2011; Thomson, Lebiere, Anderson, & Staszewski, 2015). The use of a common framework ensures that new models enjoy a theoretical foundation supported by existing research while the architecture benefits from its expansion into and evaluation under new

paradigms. ACT-R provides a platform for integrating the task-specific elements of my model with elements from other tasks, building toward a universal theory of cognition (i.e. cognitive supermodels; Salvucci, 2010, 2013). The ability to generalize to different tasks and situations is especially important when modeling WM because WM influences processing in so many areas of cognition.

The structure of ACT-R is modularized so that certain cognitive processes are localized within specialized units, called modules (e.g., the Goal module keeps track of the current goal state, the Visual module controls visual attention, etc.). These modules (Figure 1.1) are theorized to process information in parallel, performing many calculations quickly and independently of any processing occurring in the other modules. The modules communicate with each other through buffers. Buffers can only hold one piece of information, called a chunk, and thus form bottlenecks in the otherwise parallel architecture. Chunks are generic containers for sets of features, called slots. A specialized module, the procedural module, performs a sort of central processing role. The procedural module maintains a set of condition-action pairs called production rules, the total of which represents procedural memory. If the contents of the model's buffers match the conditions of the production rule, then the actions (i.e. buffer manipulations) of the production rule are executed by the procedural module. The procedural module can only fire one production rule at a time, providing the architecture with one final additional bottleneck. Because ACT-R contains perceptual and motor modules based on well-established theories of perception and action (Byrne & Anderson, 2001; Meyer & Kieras, 1997), it is able to model task performance from the start of the trial to the end of the trial.

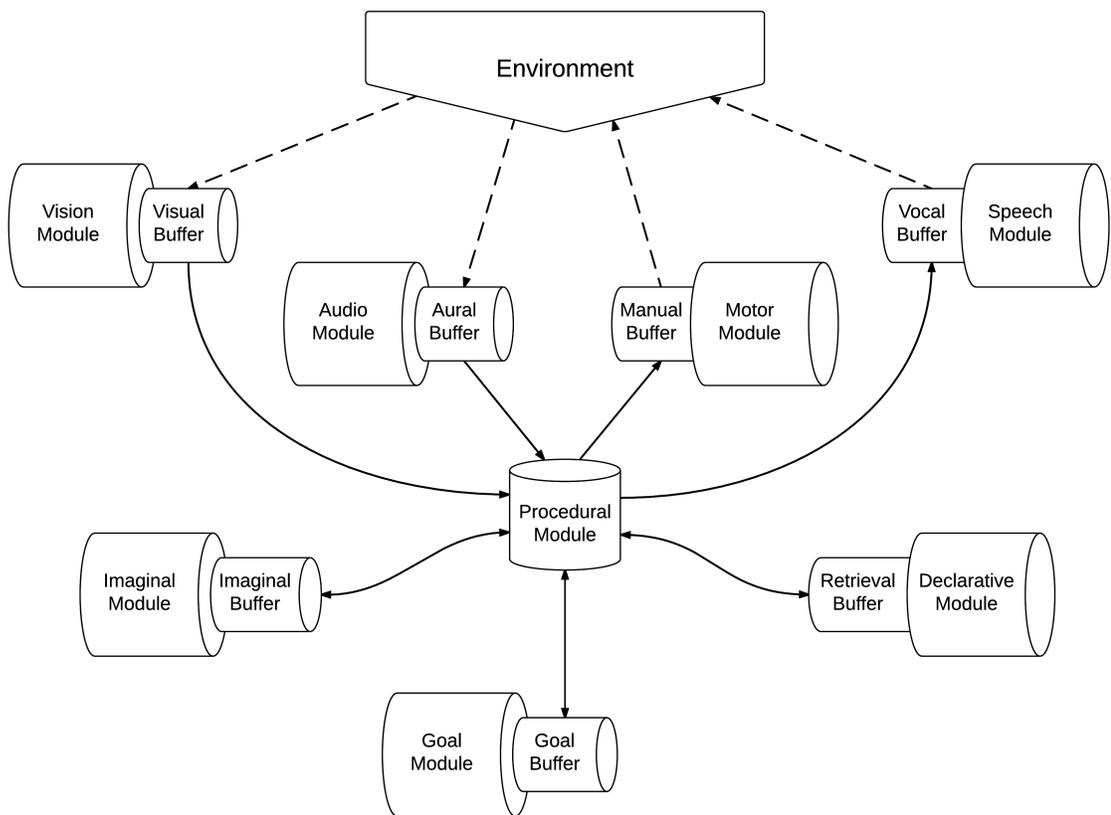


Figure 1.1: Schematic representation of the ACT-R cognitive architecture demonstrating the interaction of modules through buffers.

## 1.6 Compatibility of ACT-R and TBRS

In addition to the previously discussed advantages of ACT-R, I chose to use ACT-R because it conveniently shares or is highly compatible with all of the TBRS model's specifications. Recall the first assumption of TBRS: executive functions (e.g., information processing and maintenance) require some limited resource called attention. In an ACT-R model, information processing and maintenance take place in modules at the request of the procedural module. The procedural module directs attention by placing chunks in the buffers of modules. As such, the modules' buffers are considered to be the model's foci of attention. The size of each attentional focus is limited to the capacity of the buffer — one chunk. The rate at which attention can be redirected is also limited in ACT-R. This rate, the production-firing rate, is a parameter of ACT-R that controls how often the procedural model fires a new production. Only one production may be fired at a time, and only one attention redirection (buffer manipulation) can happen for each module when a production fires. These limits on attention constitute a central bottleneck and consequentially implement the second assumption of TBRS.

The analogy of buffers as attentional vessels is explicit in the visual module of ACT-R. Visual attention is shifted when the procedural module submits a request to the visual module with a particular visual location. The visual module then moves its “eyes” to fixate on that location and places (encodes) a visual representation of the object at that location into its buffer. In the same way, the declarative (memory) module, at the request of the procedural module, takes cues in the form of feature-slot values and shifts its attention to the most highly activated chunk that sufficiently matches the cues. The module then puts that chunk into its buffer (i.e. into memory's focus of attention). The activation that governs both a chunk's probability and speed of recall is quantified with ACT-R's activation equation. One specific portion of this

equation, base-level learning (Equation 1.6), embodies the third assumption of TBRS, that items in the focus of attention are refreshed while the others decay with time. Activation increases as a function of the number of times  $n$  a chunk has been retrieved (the frequency effect) and decays as a function of time since each retrieval (the recency effect). When a chunk is retrieved (attention is shifted to it) and subsequently cleared from a buffer, activation increases because the number of retrievals in Equation 1.6 has increased. When a chunk is not the focus of attention (not in the buffer), its activation decays as a function of  $\delta$ , a model parameter.

$$B_i = \ln \left( \sum_{j=1}^n t_{ij}^{-\delta} \right) \quad (1.6)$$

The TBRS hypothesis that WM rapidly switches between maintenance and processing activity is implemented by the modular structure of ACT-R. During the secondary task, there are small time intervals where the declarative module is not required by the task itself. One such interval is the time after a response has been decided and motor processes are executing a button press. As long as the declarative module is not currently busy, the procedural module is ready to fire, and a certain production matches the current context (the contents of the buffers); that production can direct the declarative module's attention to targets of the recall task, independent of the processing currently occurring in the other modules. In these ways, ACT-R is capable of implementing a time-based, attentional resource-sharing model of working memory.

## 2 Description of the Task

To validate the model, I chose to use an existing, published study rather than design and conduct a new study myself. This provides a less controversial benchmark for the model to pass than a new study because the human data will already have been peer-reviewed. I chose the continuous span task from the third experiment of Barrouillet et al. (2007) because it demonstrates temporal effects on WM spans across two different tasks administered at varying presentation rates. Fitting these two manipulations, comprising six experimental conditions, is more challenging and forces my model to be more comprehensive than would a simpler experiment with only a single manipulation. Participants were presented with a series of to-be-remembered consonants interspersed with a fixed number of distractor elements (Figure 2.1). At the end of each series, the word “Recall” was presented, prompting the participant to say aloud the target consonants in the order they were presented. Each series was repeated three times with new, randomly chosen consonants. As long as the subject was able to correctly recall at least one of the three series at a particular level, the experiment would continue with a new set of three series, each with one more target than the prior set of three series. The experiment ended once the subject failed to correctly recall all three series in a set or after they completed the seventh set. In Barrouillet et al. (2007), 97 participants were divided into six between-subjects conditions (2 types of distractor task  $\times$  3 levels of number of inter-letter distractor elements). In both types of distractor tasks, a number (in Arabic numeral form) would be presented either above or below a horizontal centerline. In the parity condition, subjects were required to respond by pressing the ‘f’ key if the number was odd and the ‘j’ key if the

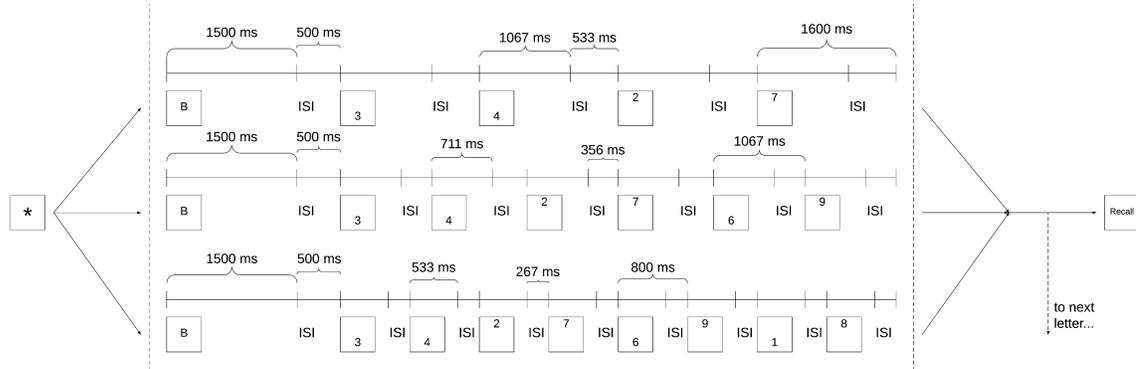


Figure 2.1: Timeline of the three distractor-pacing conditions: four, six, and eight; from the top-down, respectively. Parity and location judgment conditions only differed with respect to instructions. Each series started with the presentation of an asterisk, followed by a target letter and then a set of distractors (numbers). After the last number was presented, if every item in the list had been presented, the word “Recall” would be presented. Otherwise, a new target letter would be presented and the portion between the vertical dashed lines would be repeated.

number was even. In the spatial location condition, subjects were required to respond by pressing the ‘f’ key if the number appeared below the centerline and the ‘j’ key if the number appeared above the centerline. Although instructed to respond differently depending on the condition to which they were assigned, subjects in both conditions were presented with exactly the same stimuli. Target letters were always presented for 1,500 ms followed by a 500 ms inter-stimulus interval (ISI) and then 6,400 ms of distractor items. The rate at which distractors were presented, however, depended on the experimental condition (Figure 2.1). Four, six or eight distractors were presented for 1,067 ms, 711 ms or 533 ms, respectively, with corresponding ISIs: 533 ms, 356 ms or 267 ms. Span scores were calculated by taking the sum of correctly recalled series divided by three (essentially all-or-nothing unit scoring; Conway et al., 2005). Before beginning the experiment proper, participants completed 96 practice distractor items in their assigned condition, followed by two practice series of the continuous span task.

## 2.1 Human Results

TBRS predicts that cognitive load, not task difficulty or task duration, affects observed WM span. (Barrouillet et al., 2007) demonstrate this in Experiment 3 by manipulating the type of distractor task and pacing while holding duration constant. They report group means for WM spans, distractor response times (RT), and “total processing times” (TPT), which are the sum of RTs per string of stimuli. I relay these results in Table 2.1. Barrouillet et al. (2007) do not analyze the accuracy of the parity/location judgments (reported as 91% and 97% respectively) beyond checking that a criterion of 80% was surpassed. For the other dependent variables, I provide a qualitative summary of their statistical analyses; for specific values, I invite the reader to reference the original manuscript.

Mean WM spans were poorer in the parity condition than the location condition (4.48 and 5.23, respectively). They were also found to decrease as the number of inter-letter distractors increased: 5.36, 5.05 and 4.15; marginalized for four, six, and eight stimuli, respectively. No significant interaction was observed. Mean RTs were significantly greater for the parity condition than the location condition (554 ms and 411 ms, marginalized respectively), reflecting the relative difficulty of the tasks. Mean RTs were also found to decrease, without interaction, as the number of inter-letter distractor items increased (556 ms, 469 ms and 422 ms for four, six, and eight stimuli, respectively), which the authors suggest may indicate the presence of a speed-accuracy trade-off coping strategy. Similarly, mean TPTs were significantly greater for the parity condition than the location condition (3,147 ms and 2,351 ms, respectively) and increased with the number of inter-letter distractor items (2,198 ms, 2,774 ms and 3,275 ms for four, six, and eight stimuli, respectively).

After including TPT as the covariate in an ANCOVA, Barrouillet et al. (2007)

Table 2.1: Results from experiment 3 of Barrouillet et al. (2007)

Number of Distractors	Type of Distractor Task							
	Parity				Spatial Location			
	Span	RT	TPT	Span	RT	TPT	Mean Span	Mean RT
4	5.16 (0.78)	628 (117)	2467 (400)	5.56 (0.75)	484 (61)	1928 (233)	5.36	556
6	4.58 (1.23)	551 (53)	3251 (316)	5.52 (0.62)	387 (41)	2297 (239)	5.05	469
8	3.69 (0.63)	483 (32)	3724 (218)	4.60 (0.82)	361 (39)	2827 (266)	4.15	422
<i>M</i>	4.48	554	3147	5.23	411	2351	4.85	—

**Note.** Response time measures (RT and TPT) are reported in milliseconds. Standard deviations reported in parentheses.

found that task type no longer significantly predicted mean span, suggesting the effect of task is produced solely by the task’s temporal demands and not its difficulty. By contrast the inclusion of TPT also reduced the main effect of the number of distractors, but this effect remained significant, which they suggest may reflect the presence of switching costs.

To illustrate these results, Barrouillet et al. (2007) regressed group mean spans on TPT divided by the total time available per string of stimuli. Assuming response times are representative of the interval during which the central bottleneck is blocked, this ratio is a proxy for cognitive load (see Equation 1.4). The slopes ( $-7.82$  and  $-7.68$  for the parity and the location tasks, respectively) and intercepts ( $8.04$  and  $7.84$  for the parity and location tasks, respectively) of the two regression lines were very similar, further supporting TBRS. These regressions relating span to cognitive load will be important for my model to reproduce because they characterize TBRS’s main prediction (Equation 1.3).

## 3 Description of the Model

The model follows a general pattern of behavior to implement TBRS. It prioritizes addressing experiment-generated stimuli, whether a target to encode, a distractor to interpret and to respond, or a probe to recall. While not engaged in such processing, the model defaults to making repeated memory retrievals to simulate attentional refreshing. The specifics of these modes of cognitive behavior are best discussed in terms of declarative memory and non-declarative considerations.

### 3.1 Declarative Memory

The declarative knowledge structure employs five types, or classes, of chunks: a generic goal chunk to maintain the current goal state, stimulus chunks to encode the semantic representation of a stimulus, target chunks to encode the to-be-remembered episodes (consonants), number fact chunks to reflect parity knowledge of digits, and response rules to represent the task instructions. Examples of specific members of these chunk-types are provided in Table 3.1. I do not explicitly model the instructions phase of the experiment and assume that response rules are already present and accessible in declarative memory. Similarly the model is assumed to have prior knowledge of letters, words, and numbers; and can readily recognize them. Only the target chunks do not exist in memory at the start of the experiment; the model generates them after interpreting a particular stimulus as an item to be remembered.

The retrieval of chunks from declarative memory is governed by the model's

Table 3.1: Declarative Memory Chunk Examples

Type	Name	Slot	Value	Slot Description
<i>goal</i>	parity-goal	<i>condition</i> <i>which-list</i> <i>state</i> <i>location</i> <i>response</i>	parity "1000" get-resp-rule above "f"	The current task (parity or location judgment) Unique identifier of the current list of targets General state variable used to control behavior Location of the current stimulus Which/whether a response has been made
<i>stimulus</i>	one	<i>string</i> <i>type</i>	"1" number	Semantic representation Category of stimulus
<i>stimulus</i>	b	<i>string</i> <i>type</i>	"b" letter	Semantic representation Category of stimulus
<i>stimulus</i>	asterisk	<i>string</i> <i>type</i>	"*" special	Semantic representation Category of stimulus
<i>target</i>	TARGET1-0	<i>string</i> <i>parent</i> <i>list</i> <i>episode</i>	"c" start "1000" 1.941	Semantic representation Used to indicate the head of a list; null for the list's remaining members Unique identifier of the list to which this target belongs Representation of when the target was first encoded
<i>num-fact</i>	nf-two	<i>string</i> <i>parity</i>	"2" even	Semantic representation The number's parity
<i>resp-rule</i>	resp-below	<i>condition</i> <i>response</i>	"below" "j"	Possible parity/location of an item Appropriate response for the above condition

**Note.** Slot values in quotes indicate character strings. All other values are numerical or chunks themselves.

activation equation:

$$A_i = \ln\left(\sum_{j=1}^n t_{ij}^{-\delta}\right) - \ln\left(1 + \left(\frac{t_{in}}{\gamma_s}\right)^{-\gamma_d}\right) + P_i + \beta + \varepsilon \quad (3.1)$$

where  $A_i$  is the activation of chunk  $i$ ,  $t_{ij}$  is the time elapsed since the  $j^{\text{th}}$  presentation of chunk  $i$ ,  $n$  is the number of times chunk  $i$  has entered memory, and  $\varepsilon$  is logistically distributed noise independently generated for each retrieval attempt. I override the default partial-matching component  $P_i$  to implement my own formulation of temporal association:

$$P_i = \begin{cases} 0 & \text{if chunk } i \text{ matches the request perfectly} \\ -\infty & \text{else if chunk } i \text{ is not a target chunk} \\ \xi - \eta \cdot \ln\left(1 + \frac{\varepsilon_i - \varepsilon_{\text{requested}}}{\omega}\right) & \text{else if chunk } i \text{ is a target chunk} \end{cases} \quad (3.2)$$

where  $\varepsilon_i$  is the value in target chunk  $i$ 's episode slot (when it first entered memory) and  $\varepsilon_{\text{requested}}$  is the value in the episode slot of the retrieval cue. A full list of the model parameters is provided in Table 3.2.

The first logarithmic component of the model's specific activation equation (Equation 3.1) is the previously discussed base-level learning component (Equation 1.6). This element embodies the core mechanism of WM. When a chunk is retrieved or first encoded, a new trace to that particular memory is established. After the chunk is cleared from the focus of attention, the active connection of that specific trace to the present is severed and it decays away. The accessibility of the chunk at some later time is the sum of its surviving traces.

The partial-matching component  $P_i$  effectively only acts on the retrieval of target chunks; non-target chunks are quantitatively penalized in a way that simulates symbolic matching (i.e. zero mismatch penalty for identical chunks and infinitely

Table 3.2: Model Parameters

	Name	Symbol	Value(s)
Fixed	Utility learning rate	$\alpha$	0.2
	Utility noise	$s_U$	1
	Base-level learning decay	$\delta$	0.5
	Activation noise	$s_A$	0.3
	Base-level inhibition scaling	$\gamma_s$	1
	Temporal association scaling	$\omega$	1
	Temporal association constant	$\xi$	0
	Retrieval threshold	$\tau$	0
-----			
Free	Reward	$R$	[1, 5, 9, 13, 17]
	Base-level inhibition decay	$\gamma_d$	[2, 3, 4, 5, 6]
	Base-level constant	$\beta$	[1, 3, 5, 7, 9]
	Episodic selectivity	$\eta$	[2, 3, 4, 5, 6]
	Latency exponent	$f$	[0.1, 0.4, 0.7, 1.0, 1.3]
	Latency factor	$F$	[0.1, 0.3, 0.5, 0.7, 0.9]

large mismatch penalty for unlike chunks). Partial-matching as implemented in this model (Equation 3.2) provides the structure of the memorized list through a penalty proportional to the log-time between target encodings. A chunk is maximally similar to itself (hence 0 penalty) while it is increasingly less similar to chunks encoded at increasingly different times, producing greater negative associations. The temporal association constant  $\xi$  is used to counterbalance framing partial-matching as a penalty (i.e. choosing to subtract activation from dissimilar items rather than adding activation to similar items). Note that when the temporal association scaling parameter  $\omega$  is set to 1, chunks are compared by the difference in their absolute times of encoding, but as  $\omega$  approaches the target presentation rate, the comparison approaches the difference in their ordinal positions (i.e. positional coding). In this way, the model takes advantage of the ecology of the task: the serial order of the targets is already established by their presentation order; the model must only preserve it.

The second logarithmic component of Equation 3.2 corresponds to base-level

inhibition (Lebiere & Best, 2009). The current form of this mechanism in ACT-R evolved from repeated attempts to resolve the longstanding need for a method of repetition suppression in the architecture. ACT-R originally enacted a form of suppression using FINSTs, which in Pylyshyn (1989) indexed object positions for tracking but in ACT-R are used to keep track of previously attended items, where recently attended visual stimuli or recently retrieved memories are tagged so that they can be ignored during later requests. However, such an all-or-nothing mechanism, similar to the nonspecific suppression mechanism in primacy models, is too strong, and it is unclear why or how a cognitive system would know when to release the FINST. Juvina and Taatgen (2009) adapted the classic ACT-R FINST mechanism to decay continuously with time in a manner functionally equivalent to the current base-level inhibition mechanism. They use decaying FINSTs to explain between-trial effects in the Stroop paradigm while providing a thorough refutation of suppression-free accounts. Base-level inhibition has also been used to explain sequential effects in task-switching experiments (Grange & Juvina, 2015; Grange, Juvina, & Houghton, 2013).

Temporal inhibition provides the model with a means for traversing the list during maintenance. Consider the case when an item from the target list has just been retrieved for refreshing and now the next item must be retrieved. Rather than estimating the context of the next target item and then using that as its retrieval cue, as in Burgess and Hitch (2006), the model simply uses the immediately accessible context of the just retrieved item, relying on the fact that the next item in the list is temporally proximate to this context. Base-level inhibition heavily penalizes very recently retrieved items, preventing the just retrieved chunk, which matches the retrieval request perfectly, from being retrieved again. Without it, this strategy would never be able to rehearse more than one item.

It is worth noting that the model is able to account for forward, ordered traversal of the target list without using an explicit primacy gradient (cf. Page & Norris, 1998; Solway et al., 2012). The associations constituting the structure of the list (Equation 3.2) are bidirectional, meaning that the target item at position  $x$  in the list equally cues the items at positions  $x - 1$  and  $x + 1$  (assuming constant presentation rate). In fact, because earlier items in the list were presented earlier in the task, the item at position  $x - 1$  will necessarily have had more opportunities to be refreshed than the item at position  $x + 1$ . Thus it is likely to have respectively greater activation, and therefore more likely to be retrieved, due to base-level learning (Equation 1.6). However, for this same reason, the item at position  $x - 1$  is likely to have been retrieved recently before the item (at position  $x$ ) now being used as the retrieval cue, causing it to be penalized by inhibition. The item at position  $x + 1$ , on the other hand, likely has not been retrieved since the last refreshing cycle through the list, causing it to experience minimal inhibition. Provided the parameters of the model are sufficiently calibrated for base-level inhibition to overcome base-level learning, items will be retrieved one after the other in consecutive order. Although inhibition encourages forward traversal of the target list, it does not strictly enforce it. If for any reason an item were to be skipped over, causing a transposition error, the skipped item (at position  $x - 1$ ) would not undergo inhibition and be free to compete with the item at position  $x + 1$ . The aforementioned greater relative activation of the preceding item would likely cause it to be retrieved over the successive item, which may provide an explanation for the “fill-in” effect (Surprenant, Kelley, Farley, & Neath, 2005).

The two remaining terms in Equation 3.1 are the base-level constant  $\beta$  and transient activation noise  $\varepsilon$ . The base-level constant is a simple constant parameter used to counter the penalties of base-level inhibition and partial-matching. The transient noise component  $\varepsilon$  is independently sampled from a logistic distribution (intended to simulate a Gaussian distribution with  $\mu = 0$  and  $\sigma^2 = \frac{\pi^2}{3}s_A^2$ , where  $s_A$  is

a model parameter; see Table 3.2) each time the activation is computed for a chunk. The noise term not only allows for errors and failures to be made during retrieval, it contributes to the variability in retrieval latencies. The time it takes to retrieve a chunk  $RT_i$  is dependent on its activation  $A_i$  and computed using the following:

$$RT_i = \begin{cases} F \cdot e^{-(f \cdot A_i)} & A_i \geq \tau \\ F \cdot e^{-(f \cdot \tau)} & A_i < \tau \end{cases} \quad (3.3)$$

Equation 3.3 simulates the memory search process. It assumes that the cognitive system attempts retrieval for a particular length of time before giving up if a matching chunk has not been found yet. The time it takes for the search to retrieve a chunk is an exponential function of its activation, scaled by the latency factor parameter  $F$  and the latency exponent parameter  $f$  (Table 3.2). If no chunk matches the retrieval request with activation greater than some threshold  $\tau$ , then  $\tau$  is used in place of activation to compute the latency of the failed retrieval. In my model, if the model fails to retrieve a chunk in the service of responding to a distractor, then it is forced to guess. If it fails to retrieve a chunk while engaged in maintenance (refreshing), then it simply tries again to refresh any target chunk.

Lastly, because chunks like the response rules, letters, and numbers representations are long-term memories and not transient like the target chunks, I needed to stabilize their activations at some value above threshold to ensure that they are nearly always retrievable (or else the model may forget how to read!). One way to do this in ACT-R is to artificially create a number of past references to the chunks used in base-level learning (Equation 1.6). ACT-R's *set-base-levels* command is designed to achieve this by creating some modeler-specified number of evenly spaced references back to some modeler-specified time. I first estimated the maximum duration of an experimental session by adding up the fixed durations of every trial and estimates of recall latencies

from preliminary model runs. Then I calculated the 95<sup>th</sup> percentile of the retrieval noise distribution to determine how much activation corresponds to reliably above the retrieval threshold. Simulating the *set-base-levels* command in R (R Development Core Team, 2008), I found values (number of references: 1250; first creation-time: -1,000,000 seconds) that produced a relatively flat level of activation at my threshold for twice the approximate experiment duration. Fine-grained adjustment of this activation during fitting is controlled by the base-level constant ( $\beta$ ) parameter.

I disabled ACT-R's spreading activation mechanism, which allows associations between a chunk and the model's buffer contents to influence retrieval, for two reasons. The first reason is simplicity. The only type of association employed by my model is between the temporal features of target items, and it is easy enough to implement this using partial-matching, which is already required to account for retrieval errors. The second reason is because spreading activation invokes ACT-R's fan mechanism. I wanted to avoid using the fan mechanism, which divides source activation amongst associations, at all costs because, as discussed earlier with respect to Anderson et al. (1996), this could be interpreted as a capacity-sharing constraint on WMC, and the body of work supporting TBRS has refuted this class of capacity-sharing models.

### **3.2 Non-Declarative Learning**

The above discussion of declarative memory chunks and the mathematical formulas determining their accessibility formalizes the knowledge of the model. The non-declarative aspects, which conduct the remaining facets of cognitive behavior, are realized through the production matching, compilation and utility learning mechanisms.

I briefly described the process of production rule matching when introducing ACT-R. At regular intervals, typically 50 ms, the procedural module engages in

what is termed conflict resolution. During conflict resolution, the production rules whose conditions are compatible with the current state of every module's buffer are identified. If only one production matches the current buffer context it is fired, and if no productions match, then the procedural module remains idle until a change in some buffer triggers another round of conflict resolution. However, if multiple productions match then the production with the greatest utility (after adding some random noise, independently generated each conflict resolution from an approximated normal distribution with  $\mu = 0$  and  $\sigma^2 = \frac{\pi^2}{3} s_U^2$ , where  $s_U$  is the utility noise parameter (Table 3.2); see previous note regarding  $s_A$ ) is selected. Utility values are assigned to each production when the model is first defined, and they are updated whenever the model is rewarded according to the following:

$$U_i(k) = U_i(k - 1) + \alpha[R_i - U_i(k - 1)] \quad (3.4)$$

Equation 3.4 is a simple difference-learning rule that adjusts  $U_i$ , the utility of production  $i$ , at each reward event  $k$  to approach the expected reward for firing that production at a rate controlled by the learning rate parameter ( $\alpha$ ; Table 3.2). The amount of reward ( $R$ ; Table 3.2) is a free parameter and is awarded after the model responds to a trial. When a reward is triggered, every production  $i$  that fired since the last reward was received is awarded  $R_i$ , the initial amount  $R$  minus the time (in seconds) since it was fired. This aspect, combined with Equation 3.4, ensures that productions that lead to greater payoffs more quickly are selected more frequently.

Utility learning allows the model to learn which productions to fire when. Production compilation, the other form of procedural learning in ACT-R, allows the model to learn new productions through the fusion of existing productions. Essentially, this mechanism manifests the transitive relation of two productions that repeatedly

fire in close succession: if one production causes the model to transition from state A to state B ( $A \rightarrow B$ ), and another causes the model to transition from state B to state C ( $B \rightarrow C$ ), the model may learn a third production which transitions directly from state A to state C ( $A \rightarrow C$ ). Specifically, any time two eligible productions fire in succession, a compilation of the two is created. The first time a compiled production is created it is initialized with zero utility. Every additional time it is created its utility is updated according to Equation 3.4 with reward equal to the current utility of its first parent. Thus it takes repetition for a compiled production to achieve enough utility to compete with its parents, but once utility noise leads to its selection, it should accomplish its goal faster than its parents and begin to receive more reward than them. A series of checks within the ACT-R source code ensure that only “safe” productions are compiled (productions that will not create new bugs or discontinuities in the model). Along these lines, only buffer manipulations that produce predictable transitions may be compiled out. In the previous example, if the first production made a request to the vision module that was then checked in state B and used in the second production to transition to state C, these two productions cannot compile because the state of the environment obtained by the vision module is external and not predictable. Retrieval requests, however, are internal and dependent on the state of the model; therefore, when the model makes such a request it can be reasonably sure of what it will get back. This manifestation of production compilation is most relevant to my model, specifically in regard to response selection. For example, one production may request the correct response to an odd parity stimulus, and another production may initiate pressing the “f” key after retrieving the “respond-odd” response rule. Through production compilation, the model may learn to skip this retrieval and press the “f” key directly after determining that the stimulus is odd. Such learning may improve performance by reducing response latencies and by decreasing cognitive load by eliminating declarative retrievals.

Utility learning and production compilation are hypothesized to each contribute to the correlation between distractor pace and RT. Because there is less time available to respond in the faster conditions, fewer productions can lead to correct responses within the time allotted. These are the only productions that will be rewarded; therefore, the reduced time available to respond in these conditions strongly influences RT. Furthermore, because the total time per target-distractor series is constant in all conditions, the faster conditions contain more distractors. These additional processing episodes are additional opportunities for production compilation, which should decrease RTs in these conditions.

### 3.3 Production Rules and Model Behavior

The model's production rules, comprising its procedural knowledge, are broadly organized into six subroutines. Figure 3.1 demonstrates that the *perception* subroutine (Figure 3.2) receives the highest priority (by setting the initial utility of these productions to an arbitrary large value). I made this assumption because many demands of the task are stimulus driven and because, with regard to evolution, attending to changes in one's environment has great utility for survival. Whenever a stimulus appears on the display, the model fires productions which move its visual attention to that object. The object is then visually encoded, that is it is interpreted by retrieving its semantic representation from declarative memory. Depending on which kind of stimulus was presented (i.e. the value in the *type* slot of the stimulus chunk retrieved; see Table 3.1), the model advances to one of the four intermediate-priority subroutines.

After stimulus encoding, the first subroutine to which the model may proceed is the *new-list* subroutine (Figure 3.3). In Experiment 3 of Barrouillet et al. (2007), the presentation of an asterisk in the center of the screen indicates the beginning of a

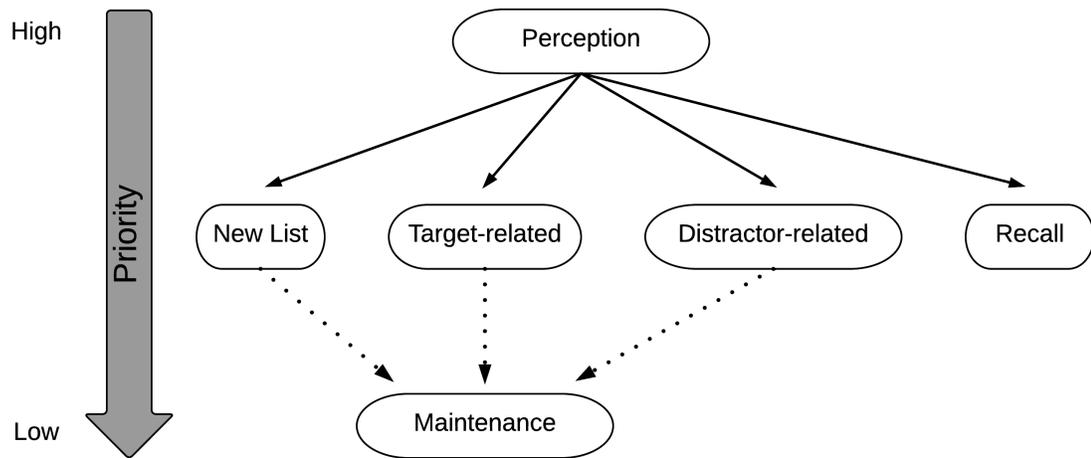


Figure 3.1: Schematic depicting the high-level organization of the model's subroutines.

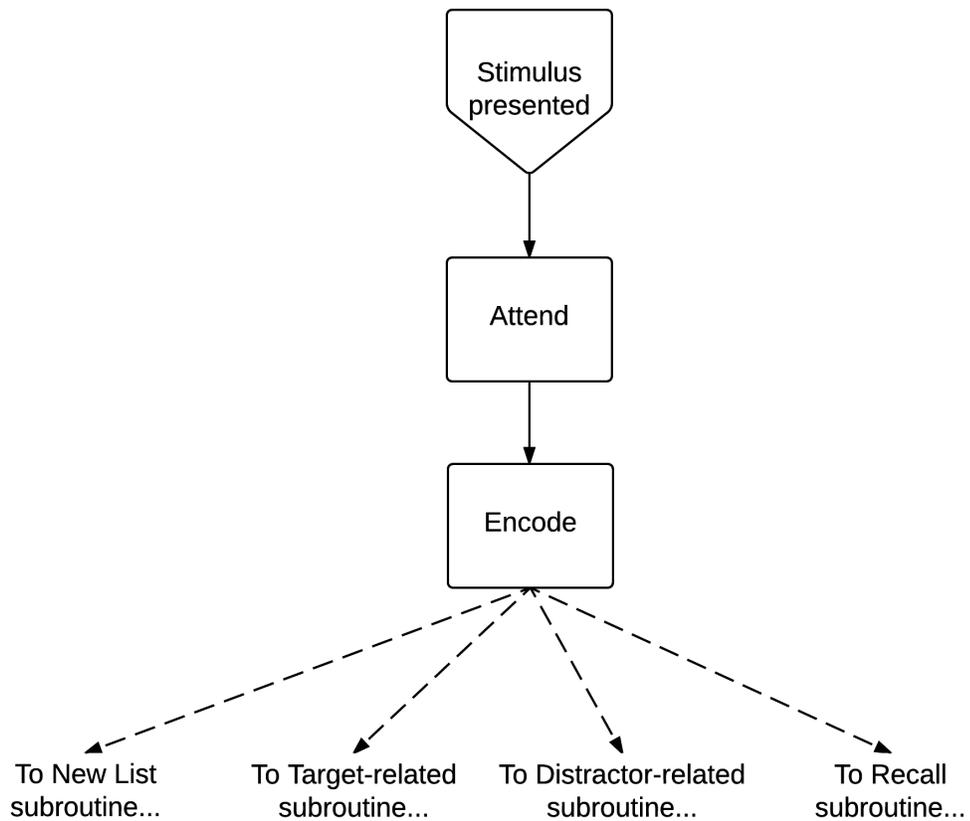


Figure 3.2: Schematic representation of the *perception* subroutine.

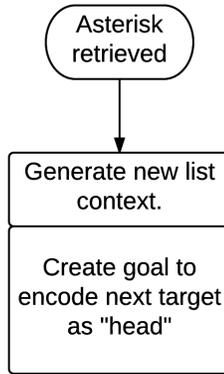


Figure 3.3: Schematic representation of the *new-list* subroutine.

new series of targets (i.e. a new list to remember). When this happens, the model generates a new context (implemented as a unique character string; see the *which-list* slot in the *goal* chunk-type in Table 3.1) to associate with the new list, and it sets a variable in the goal buffer-chunk to encode the next target presented as the first item in the list.

When the stimulus retrieved is of type *letter*, it is a new target for the model to commit to the current list. The *target-related* subroutine (Figure 3.4) recruits the imaginal module, which is used in ACT-R to add new chunks to declarative memory, to create a new *target* chunk (see Table 3.1), representing the episodic memory of the item’s presentation. The new *target* chunk includes its semantic representation, the list to which it belongs (the context in the goal buffer at the time of encoding), and the time at which it was encoded. This last feature, which is used to compute the similarity between two items in Equation 3.2, is artificially obtained to reduce the parameter space and simplify the coding process. A more rigorous approach would be to acquire the temporal representation of the item using the temporal module (Taatgen et al., 2007), a relatively newer addition to the ACT-R architecture. This would not change behavior of the current model, but may be useful for future versions (see Serial Memory in Discussion). Depending on whether the *new-list* subroutine

recently marked this letter as the first in the list, the new chunk may also include a slot indicating it as such. After the new *target* chunk has been fully encoded, which takes time (default 200 ms), it is cleared from the imaginal buffer and properly enters declarative memory so that it can be refreshed with the rest of the list items.

The most complex subroutine handles responses to distractors. Unlike the other intermediate-priority subroutines, the *distractor-related* subroutine may initiate before identifying the type of the stimulus. Distractors are the only stimuli that appear outside the center of the screen. Furthermore, they never appear in the center of the screen, making visual location a sufficient feature for identifying the stimulus as a distractor. This is a safe assumption in the task modeled (Barrouillet et al., 2007, : Experiment 3) because distractors were presented in the context of a horizontal midline so that subjects only had to make a relative comparison rather than an absolute spatial judgment. This assumption would most likely not hold in Barrouillet et al.'s Experiment 2 (not modeled) where distractors were presented without a reference line and overlapping so that the separation between potential locations was less than the size of the stimuli. Importantly, the relative stimulus location is available to the cognitive system (via the movement of visual attention) before the stimulus is encoded and its semantic representation retrieved. Hence, the model knows a distractor stimulus has been presented and is able to respond before even identifying which number was presented. The model may guess (i.e., respond before considering the appropriate response-rule set forth in the task instructions), but this is likely a poor strategy in the parity condition (Figure 3.5) because the number must be identified to first determine its parity and then its corresponding response (three retrievals). However, in the spatial location condition (Figure 3.6]) neither the number's identity nor any additional property needs to be retrieved to be accurate; the critical information needed to retrieve the response rule is available as soon as the model's "eyes" move (one retrieval). This does not mean that humans never identify

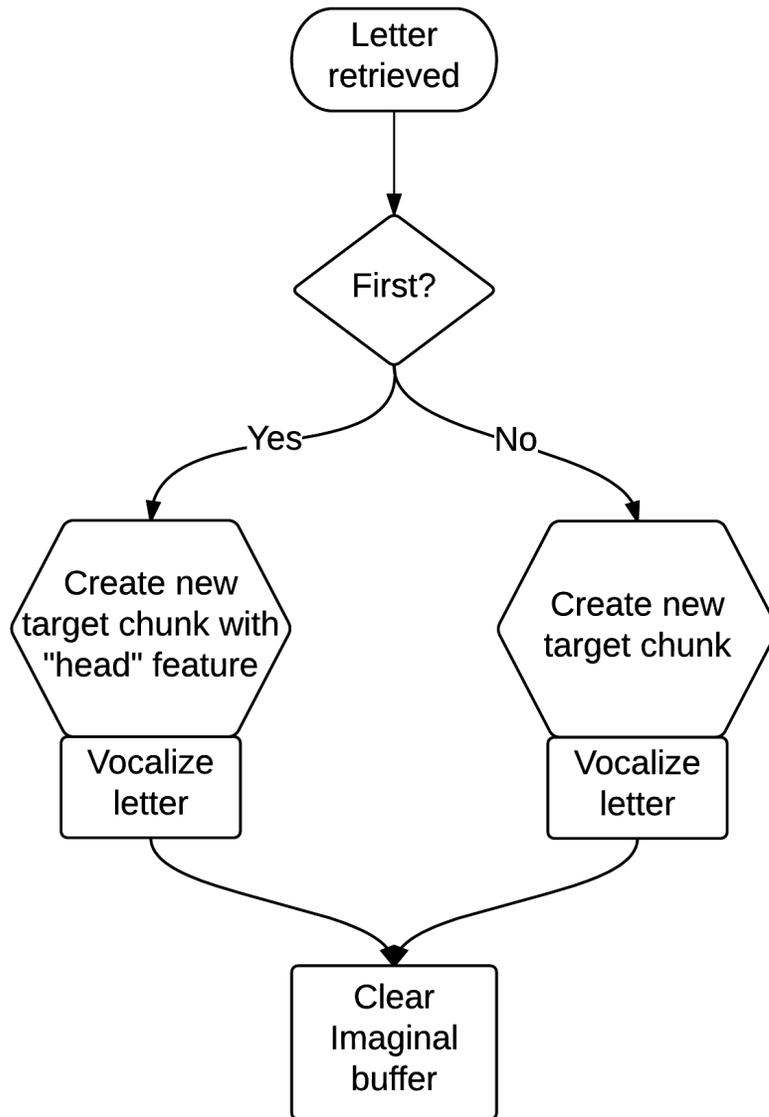


Figure 3.4: Schematic representation of the *target-related* subroutine.

the number before retrieving the appropriate response to its location, and the model can choose to do so. The disparity in the number of task-required retrievals causes spatial location judgments to be faster than parity judgments (Barrouillet et al., 2007).

Barrouillet et al. (2007) observed a speed-accuracy tradeoff in their data so I designed the *distractor-related* subroutine to be similar to Peebles and Bothell (2004) approach to modeling the speed-accuracy tradeoff. It combines a fast but less accurate strategy with a slow but more accurate strategy. The general idea is that at each point in the processing chain from stimulus to response (Figures 3.5 & 3.6) the model may either collect more information, thus reducing the uncertainty in its response, or it may guess, reducing its response latency. The mechanism of utility learning affords the model to learn the best rate at which to select one strategy over the other, but anytime the model guesses incorrectly it is forced to follow the slow but accurate strategy (i.e. it is prevented from guessing) on the next trial (Peebles & Bothell, 2004). This reflects Manly, Davison, Heutink, Galloway, and Robertson's (2000) finding that people slow down after making a mistake and helps the model to learn the correct path. For simplicity, I have not built into the model any means for retrieving the wrong information or otherwise making an unintentional mistake, although it would be possible to do so. Any incorrect response made by the model is caused by guessing incorrectly or failing to respond within the time limit of the trial.

Barrouillet et al. (2007) provided their subjects with feedback to their parity/location judgments during training but not during testing. In order for the strategy-learning approach described above to work, the model needs to know if its response was correct or incorrect. To achieve this in the testing phase where explicit feedback was not provided, the model engages in "metacognition". Anytime the model guesses it initiates its response and then continues down the processing tree (red paths in Figures 3.5 & 3.6) until it retrieves the correct response rule. It then evaluates the

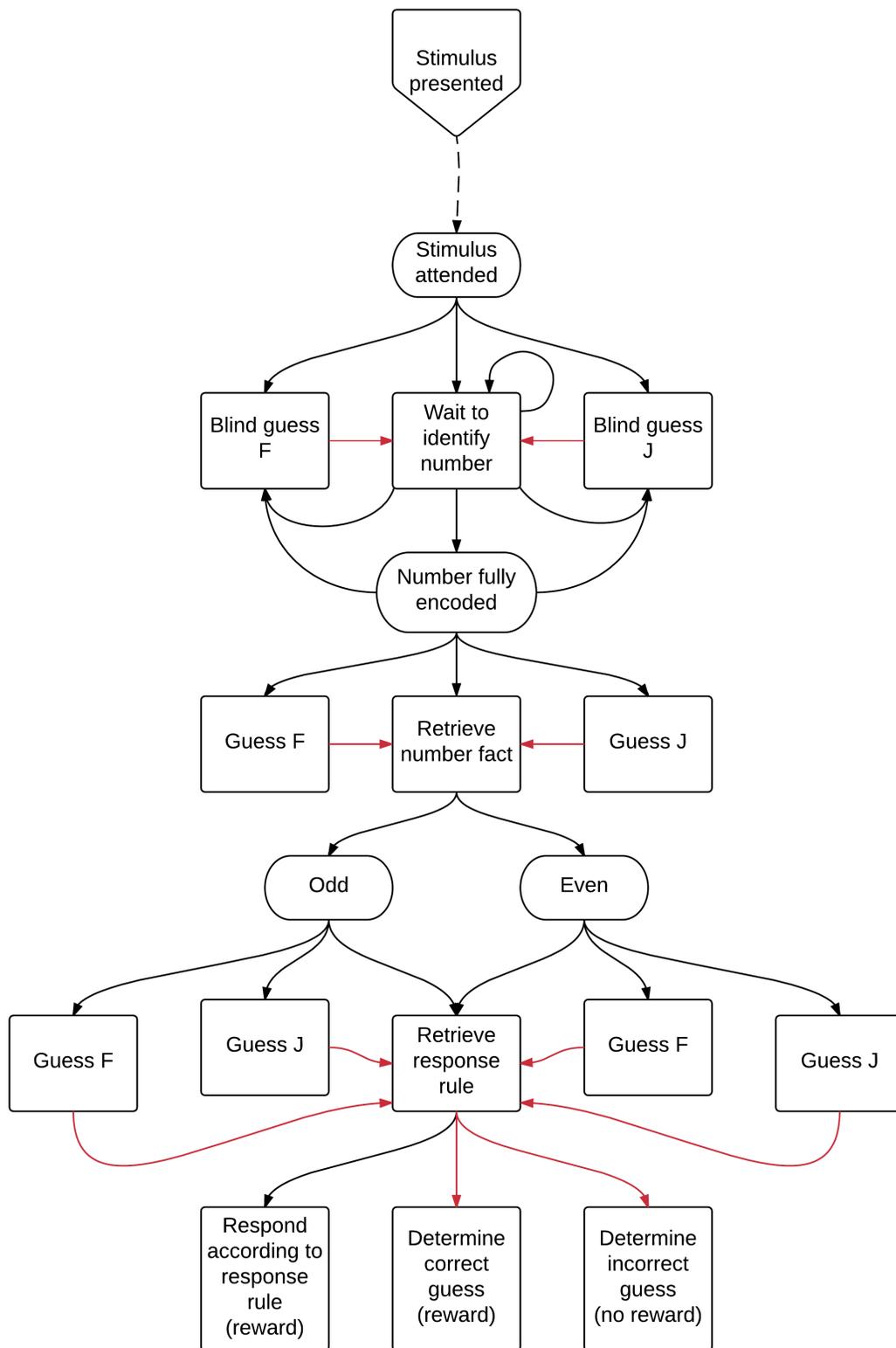


Figure 3.5: Schematic representation of the *distractor-related* subroutine for the *parity* condition.

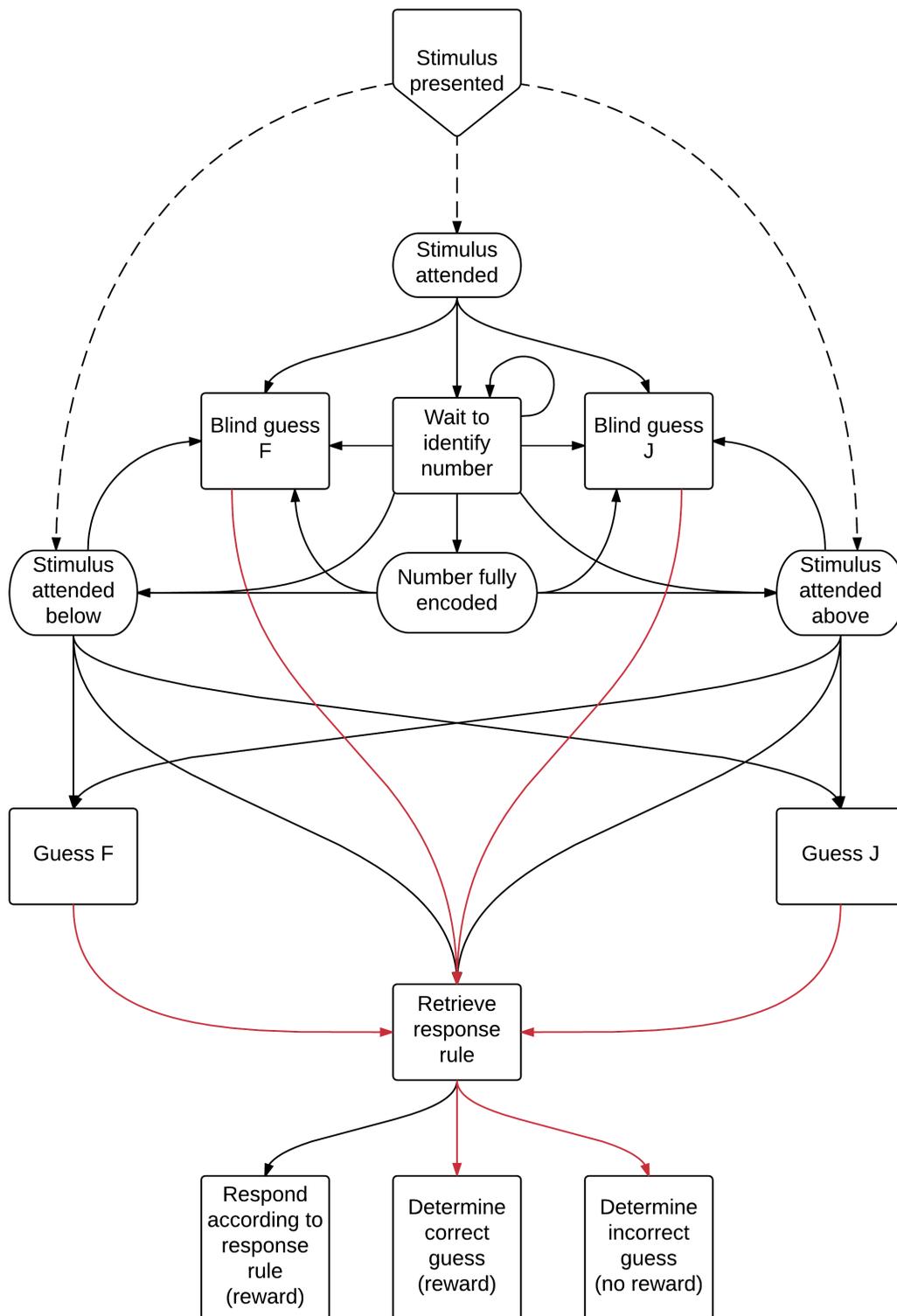


Figure 3.6: Schematic representation of the *distractor-related* subroutine for the *location* condition.

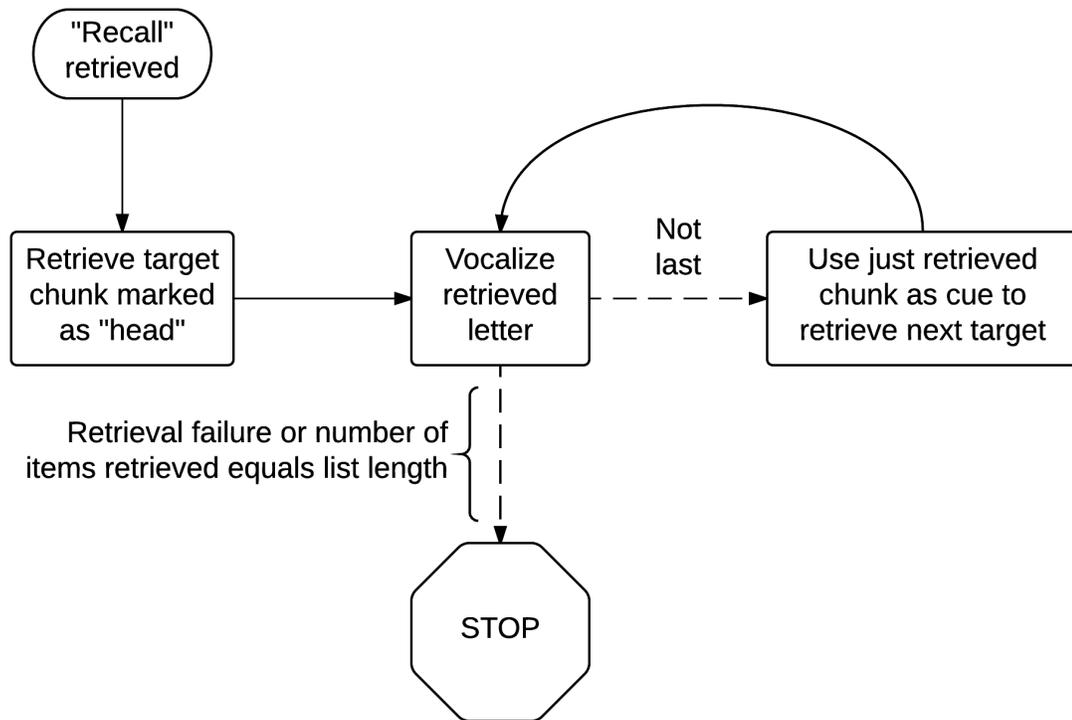


Figure 3.7: Schematic representation of the *recall* subroutine.

response it made against the retrieved rule to determine if it guessed correctly or not. Correct guesses receive the same amount of reward as responding without guessing (i.e. only after retrieving the response rule), while incorrect guesses receive zero reward.

The fourth stimulus-dependent subroutine (Figure 3.7) controls behavior in the recall portion of the experiment. After perceiving the word “Recall”, the model attempts to retrieve the first item in the list by requesting a target from the current list-context that is marked as the first item (*parent slot = start*). After retrieving an item, the model enters a loop where it vocalizes what it just retrieved and then uses the retrieved item as the cue for the next retrieval. The model halts after vocalizing as many items as in the true list or after failing to retrieve an item.

With the exception of when the *recall* subroutine is active, if at any point in the experiment the model’s retrieval and visual buffers are empty and the declarative

and vision modules are not busy (i.e. the central bottleneck is open), the model begins refreshing its target memory traces. The *maintenance* subroutine (Figure 3.8) is quite simple and consists of only two productions. The first, firing when the above conditions are met, makes a generic retrieval request for any *target* chunk with the current list-context. This avoids the modeling problem of deciding whether maintenance should always start with a certain item, such as the first item in the list or the last item refreshed. It is also another opportunity for inhibition to adapt the retrieval process by guiding attention toward items that have not been refreshed lately and thus avoiding wasting limited refreshing time on items that most likely do not need to be refreshed immediately. The second maintenance production, firing after a *target* chunk has been retrieved, uses the just retrieved chunk as an additional cue for requesting another *target* chunk with the current list-context. The second production repeats itself for as long as the central bottleneck remains open, continuing the refreshing loop until the declarative module is recruited by a higher priority routine. Typically, this happens when a new stimulus is presented, prompting the *perception* subroutine to determine the identity of the percept. The *maintenance* subroutine can be thought of as the “default” mode of the system, while the *perception* subroutine represents episodes of exogenous attentional capture, and the remaining subroutines reflect endogenous, task-driven redirection of attention.

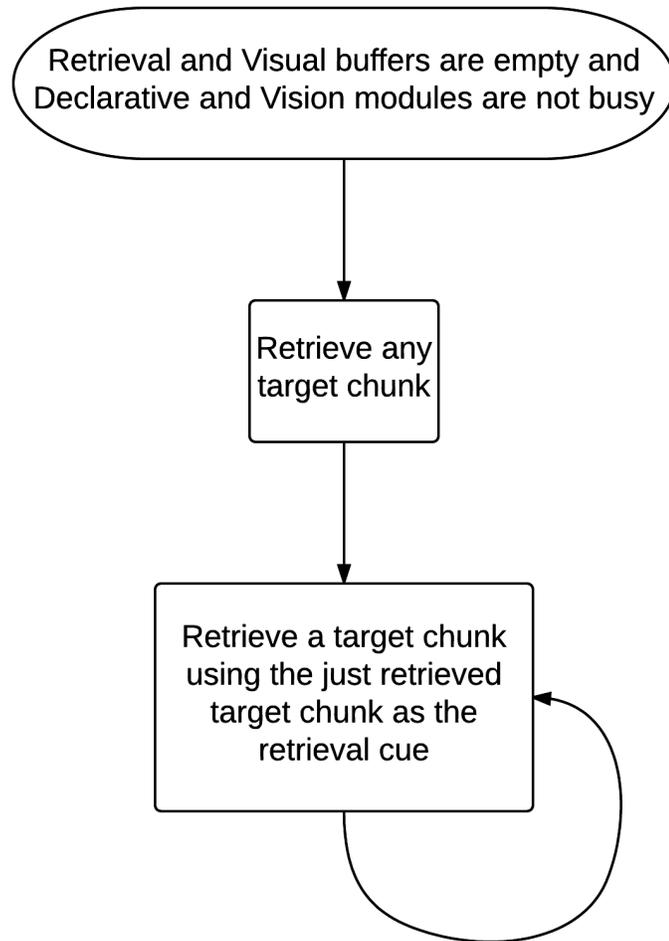


Figure 3.8: Schematic representation of the *maintenance* subroutine.

## 4 Parameters and Simulation

The model contains 14 different parameters (Table 3.2), 8 of which can be reasonably fixed at a priori values. The primary reason for fixing these parameters is tractability — each free parameter exponentially increases the size and complexity of my simulation. Superfluous free parameters also subvert the model by making it excessively flexible. Accordingly, the six remaining parameters will be fitted because it is essentially unavoidable.

### 4.1 Fixed Parameters

The following fixed parameters are discussed in the order of degree by which they have existing, accepted default values within ACT-R. Parameters with arbitrarily fixed values are justified last. The utility learning rate  $\alpha$  is kept at its default value of 0.2. Ahn, Busemeyer, Wagenmakers, and Stout (2008) conducted a quantitative comparison of various reinforcement learning models of decisions within the Iowa Gambling Task and the Soochow Gambling Task. The learning rates they estimated to best fit the human data for each respective task, 0.17 and 0.22, suggest this default value is reasonable.

The utility noise parameter  $s_U$  and base-level decay parameter  $\delta$  do not have true default values because their default value is to turn off their respective mechanisms (noisy conflict resolution and base-level learning); however, they do have conventional values within the ACT-R community. Utility noise is commonly set to 1 (Anderson, 2007) so that conflict resolution approximates Luce’s choice rule (Luce, 1959). Anderson

(2007) notes that “In the ACT-R community, 0.5 has emerged as the default value for the [base-level] decay parameter  $\delta$  over a range of applications.” I use these values for correspondence with the existing literature.

Similarly, the default value of the activation noise parameter  $s_A$  is to turn off activation noise within ACT-R. Activation noise plays a significant role in the production of errors by the model, yet there is less agreement on its value within the community. Therefore, I consulted a database of the parameter values used in submitted ACT-R models maintained by the Center for Adaptive Behavior and Cognition located at the Max Planck Institute for Human Development Berlin. I chose to fix activation noise at the database’s median value of 0.3 (An online database for ACT-R estimated parameters, n.d.).

Three parameters do not have common values because they are not attributes of the standard ACT-R release, either by others’ extension or by my own. For example, base-level inhibition is a relatively new addition to the architecture (it is included with the software distribution but not loaded by default when ACT-R is initialized), and convergent values for its two parameters are not available. By systematically plotting the base-level inhibition function for various values of each parameter, it is apparent that if inhibition is typically only employed over a relatively narrow temporal window, then it is possible to achieve similar inhibition penalties through a tradeoff in parameters. Therefore, I sought to fix one of these parameters. In preliminary development, I found that the decay parameter  $\gamma_d$  seemed to have the larger impact on the model’s behavior, while the scaling parameter  $\gamma_s$  did not notably affect model performance. For simplicity, I fixed the base-level inhibition scaling parameter at 1 so that absolute time since the last retrieval is the critical variable used in computing inhibition.

Similar to base-level inhibition, the temporal association parameters in Equation 3.2

have no standard values because I created the equation specifically for this model. It is not obvious to what the temporal association scaling parameter  $\omega$  in particular should be set. This parameter may trade-off with other temporal association parameters in a manner that is impossible to identify without the constraint of serial position data (to which I do not have access). The inclusion of  $\omega$  in the model is more for completeness than anything else, so I fix it at 1 and use absolute time as the critical variable in order to be consistent with base-level learning and base-level inhibition. An alternative to using  $\omega = 1$ , corresponding to similarity between temporal contexts, would be to set this scaling parameter to the target presentation rate, thereby defining association as similarity between list positions. I avoid this approach because of my previously discussed reservations regarding positional encoding and leave further investigations into the proper value of this parameter to future studies.

The temporal association constant  $\xi$  was originally included in the model to counter the partial-matching penalty by providing a means for raising heavily penalized items back above the retrieval threshold without impacting the activation of so-called LTM chunks. However, nonzero values of  $\xi$  lead to a discontinuity in Equation 3.2 in its current form. Assuming  $\omega = 1$ , when the difference between the retrieval cue and a target chunk's temporal context is less than  $e^{\xi/n} - 1$ , Equation 3.2 yields nonnegative "penalties" when  $\xi > 0$ , actually providing a boost in activation over identical chunks. When  $\xi < 0$ , the opposite is true, and the association penalty never approaches zero. One solution to this would be to no longer treat partial-matching as a penalty and allow identical chunks to receive additional (specifically  $\xi$ ) activation, but this would be a drastic departure from the way partial-matching is traditionally used within ACT-R. Another potential solution would be to further modify partial-matching to equal  $\min_i(0, P_i)$ , where  $P_i$  is still Equation 3.2. The addition of such a minimum function would solve the nonnegative penalty problem but introduce new issues. Mathematically, this modification introduces an elbow such that penalty no longer

smoothly approaches its maximum. Psychologically, this function would cause all targets encoded within  $e^{\xi/\eta} - 1$  seconds of each other to be interpreted as temporally identical. While this is an approach I do not wish to take currently, it may be an interesting route for future work regarding people’s ability to temporally discriminate items. Because these concerns may be of interest to the reader, I leave Equation 3.2 as it is but fix the temporal association constant  $\xi$  at 0.

The final fixed parameter of the model is the retrieval threshold  $\tau$ , which I set to 0. In addition to reducing the size of the parameter space and helping to constrain possible parametric trade-offs, fixing the retrieval threshold at zero in particular also provides a couple convenient interpretations to certain aspects of the model. First, there is simplicity in designating chunks with positive activation as retrievable and those without as inaccessible. Second, examination of Equation 3.3 reveals that when  $\tau = 0$ , the latency factor parameter  $F$  becomes immediately interpretable as the latency of a failed retrieval, or how long the model is willing to attempt to retrieve something.

## 4.2 Free Parameters

Six model parameters were explored through simulation. In order to better understand the relationships between these parameters and performance, a coarse grid search was used, evaluating the model at each parameter combination. This is in contrast to an adaptive search, which only tries to determine the best parameter combination at arbitrary scale. Such a method may identify the “best” combination of parameters without furthering any understanding of how and why the model works.

The reward parameter  $R$  governs the payoff productions receive during utility learning. Because incorrect guesses are always awarded zero reward, this parameter effectively represents the difference (in expected utility) between choices that lead

to correct responses and those that lead to incorrect responses. Greater values of  $R$  should cause the model to learn to give correct responses more often. Because the correct response pattern will be learned sooner with greater  $R$ , these productions should be repeated more, leading to more compilation. The best value for the reward parameter will generate the improved RTs observed in the conditions with more distractors, where there is more opportunity for learning, while still allowing for the slower responses observed in the conditions with only four distractors.

The base-level inhibition decay parameter  $\gamma_d$  controls how heavily recently retrieved chunks are penalized. Larger values of this parameter will cause larger penalties than smaller values, counteracting the activation built up by earlier targets through repeated refreshing. It is hypothesized that some minimal amount of inhibition is required for the model to function properly, but past a particular level increased  $\gamma_d$  will begin to cause other parameters, such as the base-level constant to increase in order to trade-off with its immense penalty. The base-level constant  $\beta$  primarily serves to counteract the base-level inhibition and partial-matching penalties. It is also used to tune the activation of LTM chunks such as number facts and response rules. Greater values of  $\beta$  globally increase the activation of all chunks, increasing the chances that it is above threshold and reducing retrieval latency. Because the base-level constant is framed as a counter to penalties, it is hypothesized to potentially trade-off most highly with other parameters. In particular, larger values of  $\beta$  allow increased base-level inhibition and temporal association gradients, which in turn allow for increased discrimination amongst items.

The episodic selectivity parameter  $\eta$  defines the linear strength of the logarithmic similarity gradient across the temporal contexts of target chunks. Together with the base-level inhibition decay parameter, this parameter controls the accuracy of the iteration of the target list during refreshing. Specifically, this parameter is hypothesized

to play a significant role in determining the rate of anticipation errors, or the rate at which the item in position  $x + 2$  is mistakenly retrieved in place of the item at position  $x + 1$ . Larger values of  $\eta$  lead to more reliable retrieval of the immediate items in a list, while smaller values effectively increase the similarity between all items and increase the influence of base-level activation.

The latency exponent parameter  $f$  controls the sensitivity of retrieval latency to activation. Larger values are more sensitive, while smaller values are less sensitive. In the extreme,  $f = 0$  causes retrieval latency to always equal the latency factor parameter  $F$ , regardless of activation (Equation 3.3). The latency factor parameter  $F$  controls the linear magnitude of the retrieval latency. Larger values unilaterally increase the latency of all retrievals. As mentioned previously, when  $\tau = 0$ , the latency factor is also interpretable as the maximum amount of time the cognitive system is willing to spend attempting retrieval. Together, these two parameters ( $f$  and  $F$ ) scale retrieval latency and may trade-off to determine the response time after other parameters like the base-level constant have settled. They also play a role in determining how many refreshing retrievals may take place during maintenance.

Overall, parameters are assumed to not vary across task conditions. Unique parameter combinations are interpreted as representing potential populations, while repeated simulations of a specific parameter combination are interpreted as sampling individuals from that population. A summary of the values used for fixed parameters and the ranges of values searched over for the free parameters may be found in Table 3.2.

### 4.3 Model Simulation

The entirety of Experiment 3, including training phase, was reconstructed in LISP using the details found in the methods section of Barrouillet et al. (2007). Lists

of target and distractor stimuli were created in parallel such that the conditions with more stimuli (the 6- and 8-distractor conditions) contained the exact same stimuli as the corresponding trials in conditions with fewer between-target distractors while still maintaining the frequencies outlined in Barrouillet et al. (2007). Stimuli lists are randomly generated when the model code is first loaded, but the same lists were used throughout parameter fitting sessions.

Five values for each of the six free parameters composed the grid search, yielding 15625 possible combinations. I ran 50 iterations of each parameter combination, systematically changing the seed of the pseudorandom number generator each time to simulate 50 unique individuals for each condition. In order to evaluate such a large space in a relatively timely manner, I took advantage of the massively distributed, volunteer computing service MindModeling@Home (Harris, Gluck, Mielke, & Moore, 2009). MindModeling@Home distributes individual model runs around the world and uses the idle processor time on volunteers' computers to simulate hundreds of parameter configurations in parallel, completing in days what would take a single machine decades to compute. Due to the complexity of the parameter space, it was used throughout the model building process to explore the behavior of the model and ensure that various components behaved as intended.

## 5 Results

### 5.1 Model Evaluation Metrics

Five group measures were of interest: mean accuracy, mean span, mean RT, mean total processing time per inter-letter interval (TPT), and linear regression coefficient of span upon CL (slope). Five metrics were developed to quantify the agreements between the model and human data for each respective measure. Error for mean span, mean RT, and mean TPT were calculated by summing across the six experimental conditions the standardized (using the standard deviations reported in Barrouillet et al., 2007) absolute differences between model and human group averages (Equations 5.1-5.3). Error due to distractor response accuracy was modeled as a linear function, scaled to yield 4 at the chance rate (50%) and 0 at or above Barrouillet et al.'s criterion of 80% for each condition (Equation 5.4). Because the span error, RT error, and TPT error scores appeared to vary between 3 and 8 during preliminary analysis, the choice of scaling for error due to accuracy reflects the decision that a parameter combination that produces perfect span, RT, and TPT scores but only chance level accuracy is equally as bad as a combination that produces acceptable accuracy levels but poor span, RT, and TPT results. I chose this equivalence because although accuracy is not a DV of remarkable interest to the study, any model that cannot meet such a bare-minimum is categorically wrong.

$$\text{Span Error} = \sum_{i=1}^6 \left| \frac{\hat{s}_i - \bar{s}_i}{\hat{\sigma}_{s_i}} \right| \quad (5.1)$$

$$\text{RT Error} = \sum_{i=1}^6 \left| \frac{\hat{RT}_i - \overline{RT}_i}{\hat{\sigma}_{RT_i}} \right| \quad (5.2)$$

$$\text{TPT Error} = \sum_{i=1}^6 \left| \frac{T\hat{P}T_i - \overline{TPT}_i}{\hat{\sigma}_{TPT_i}} \right| \quad (5.3)$$

$$\text{Accuracy Error} = \frac{40}{3} \sum_{i=1}^6 \max(0, .8 - \overline{\text{accuracy}}_i) \quad (5.4)$$

$$\text{Slope Error} = \sum_i^{\text{parity, location}} |\sin(\arctan(\hat{b}_i) - \arctan(\bar{b}_i))| \quad (5.5)$$

When it comes to calculating the regression slope of span upon CL, one could imagine the case where, due to variability in TPTs, the regression line turns out to be nearly vertical. In this situation, the regression coefficient could be largely negative or largely positive depending on exactly where the points lie. In the limit, each situation should be identical (and equally bad), but a simple difference metric would treat them as very different. In order to maintain continuity in the slope error function as the regression coefficient approaches  $\pm\infty$  (i.e. a completely vertical regression line), the coefficient is transformed into angular space (Equation 5.5). The slope angles for each (parity/location) condition are then passed through a sine function to yield minimum (zero) error when the angles are identical but maximum error when they are complementary (approximately 0.128 and 0.130 for  $\bar{b}_{\text{parity}}$  and  $\bar{b}_{\text{location}}$ , respectively). Figure 5.1 illustrates this error function.

### Slope Error Functions

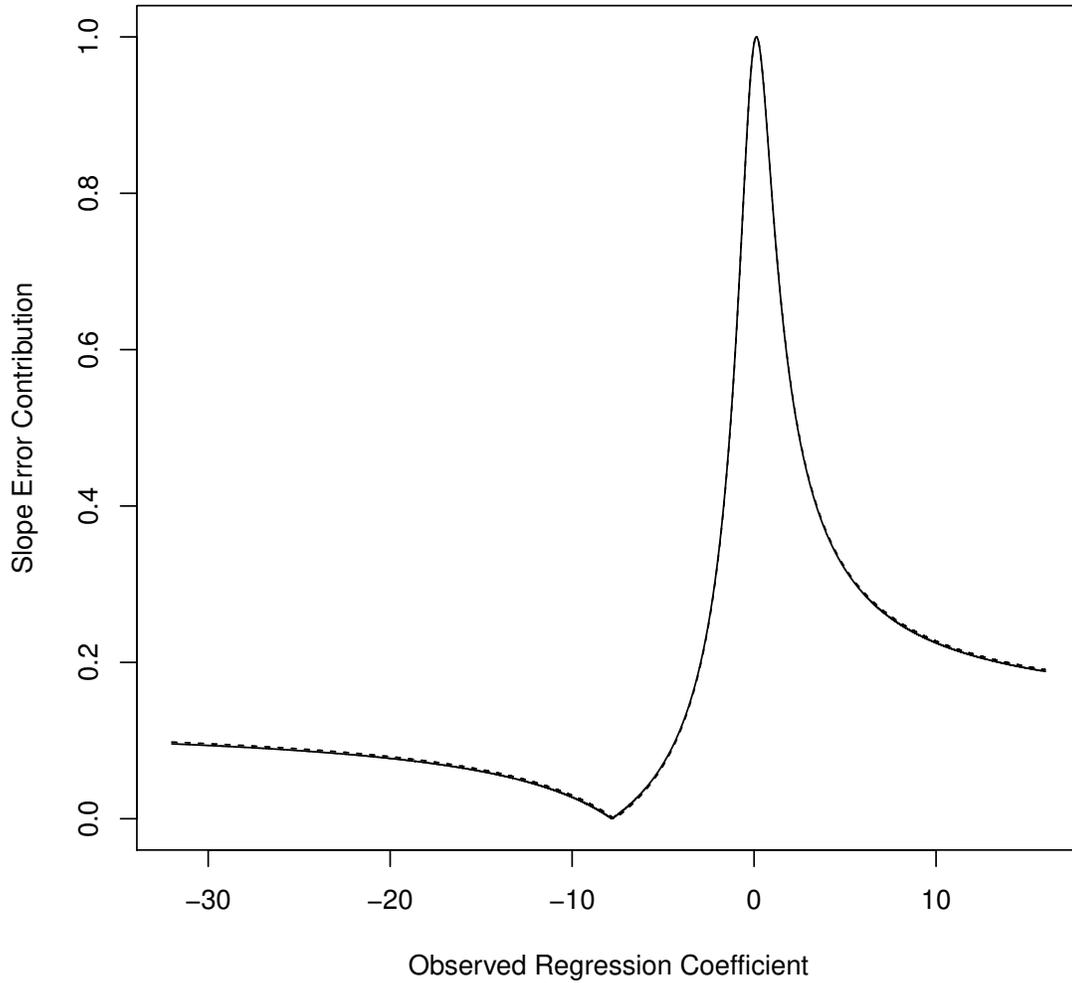


Figure 5.1: Contribution to the slope error from a single condition as a function of the regression coefficient observed in the model data. Note that both parity and location conditions are present yet indistinguishable because the difference between these coefficients in the human data is so small.

## 5.2 Determining Parameterizations of Interest

A cumulative measure is useful for determining the parametric fitness of the model over all areas of the task. The most straightforward method is a simple linear combination of all the error measures previously described. For most of the error measures, there is no a priori reason for favoring one component over another so equal weightings will be used. However, there are justifications for either underweighting or overweighting the slope error. The regression of span on CL reported in Barrouillet et al. (2007) is conducted using group means of span and group means of CL instead of the full set of individual datum pairs. Regressing means onto means may result in overconfidence in the quantification (i.e. regression coefficient or slope) of the effect of CL on span because variability in WMC (i.e. individual differences) introduces uncertainty in the regression coefficient. Furthermore, regression dilution caused by variance in TPTs (and therefore CL) contributes additional uncertainty by biasing the regression coefficients toward zero (Riggs, Guarnieri, & Addelman, 1978). Ultimately, linear regression over three data is never terribly convincing. For these reasons, it may be prudent to place less weight on slope error relative to that accumulated by the other measures. However, this regression is the central result of the original experiment, demonstrating that task type does not affect WM span once CL is controlled. If the uncertainty regarding its presence in the human data is ignored, then it is the most important result to reproduce with the model and should receive more, not less, weight. One approach to resolve this incongruity is to assume the trend between span and CL is actually present in the data, which is not unreasonable given that other studies that have found a similar effect, and use the equivalent weighting chosen for accuracy to weight slope. A more rigorous method would be to treat the weight placed upon slope error as an ad hoc variable and investigate the resulting error functions.

Framed in this fashion, the total error  $\mathcal{T}$  is formulated as a linear function of the

weight  $\varphi$  placed on the contribution from slope error (Equation 5.6). The intercept  $\mathcal{T}_0$  of this function is the sum of the other error metrics (Equation 5.7).

$$\mathcal{T}(\varphi) = \mathcal{T}_0 + \varphi \cdot \text{Slope Error} \quad (5.6)$$

$$\mathcal{T}_0 = \text{Span Error} + \text{RT Error} + \text{TPT Error} + \text{Accuracy Error} \quad (5.7)$$

I evaluated total error functions for  $0 \leq \varphi \leq 25$  and determined which parameter sets ever made it to the top 50 for any value of  $\varphi$  within this range. These functions are plotted in Figure 5.2 with the line color indicating the set’s rank at  $\varphi = 0$  (hotter colors reflect lower  $\mathcal{T}_0$ ). It appears that as phi increases, two clusters emerge. When slope error is plotted against  $\mathcal{T}_0$  (Figure 5.3), the shape of the Pareto frontier (Figure 5.4) suggests that the divide between these clusters seems to take place around *slope error* = 0.65, which I designate with a dashed line. For emphasis, I recolor Figure 5.2 using blue for lines with *slope error* > 0.65 and red for lines with *slope error* ≤ 0.65 (Figure 5.5). In order to determine what parametric differences may exist between these two clusters, I overlaid the parameter density distributions for each group using the same color scheme (Figure 5.6). Upon inspection, the cluster with smaller slope errors tends to allow larger values of the inhibition decay parameter than the cluster with larger slope errors. It also appears to favor lower values for the latency exponent parameter and to converge more strongly on 0.3 for the latency factor parameter. The distributions of reward, base-level constant, and episodic selectivity parameters do not differ between clusters.

To further explore the distributions of parameter values as a function of  $\varphi$ , I plotted the parameter values of the top 50 sets at each point in  $\varphi$ , decreasing the opacity of the plotting symbols as rank increases within a given value of  $\varphi$  (Figure 5.7). Although Figure 5.6 and Figure 5.7 convey similar information, in discrete category

### Error Gradients

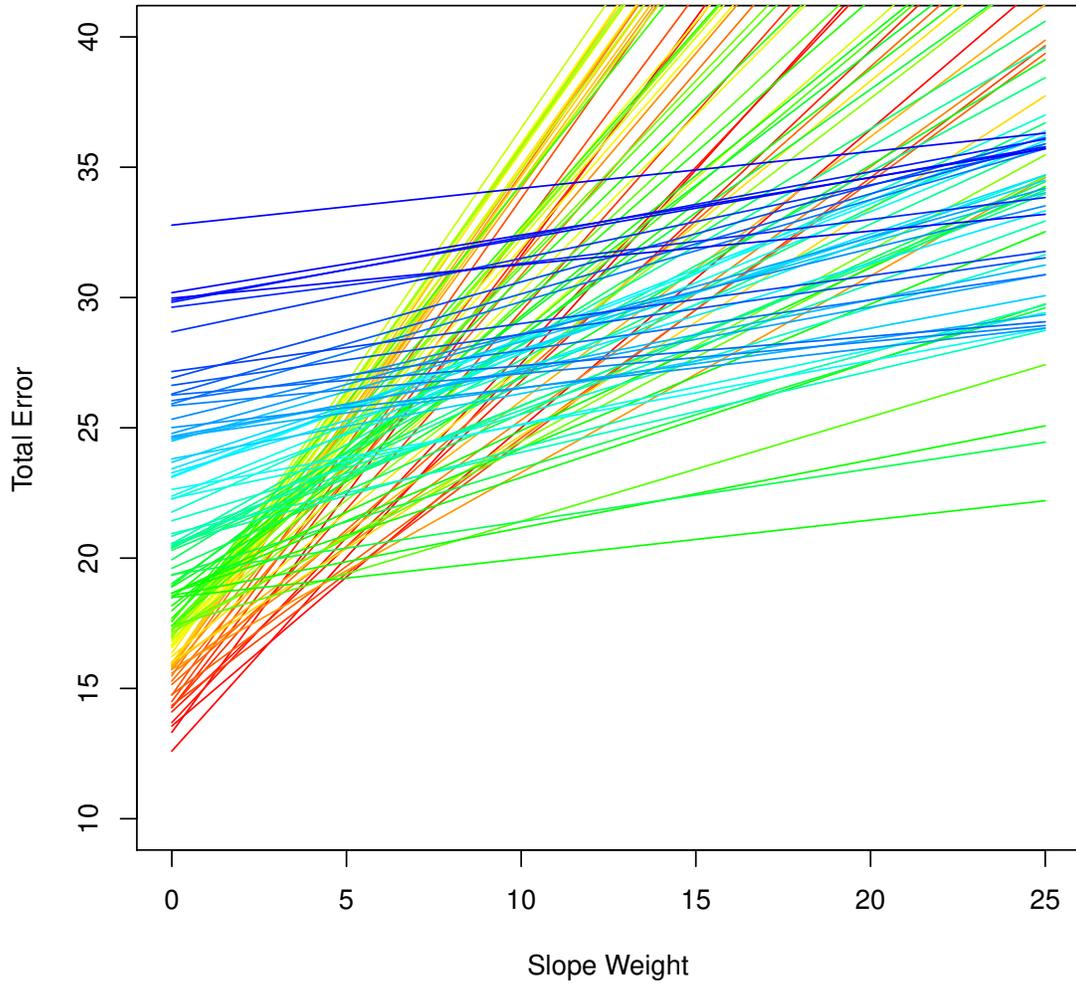


Figure 5.2: Total error (Equation 5.6) as a function of the scalar weighting of slope error ( $\varphi$ ). Any parameter set in the lowest 50 total errors over the range shown are included. Color indicates the relative ordering of these sets when slope error is excluded.

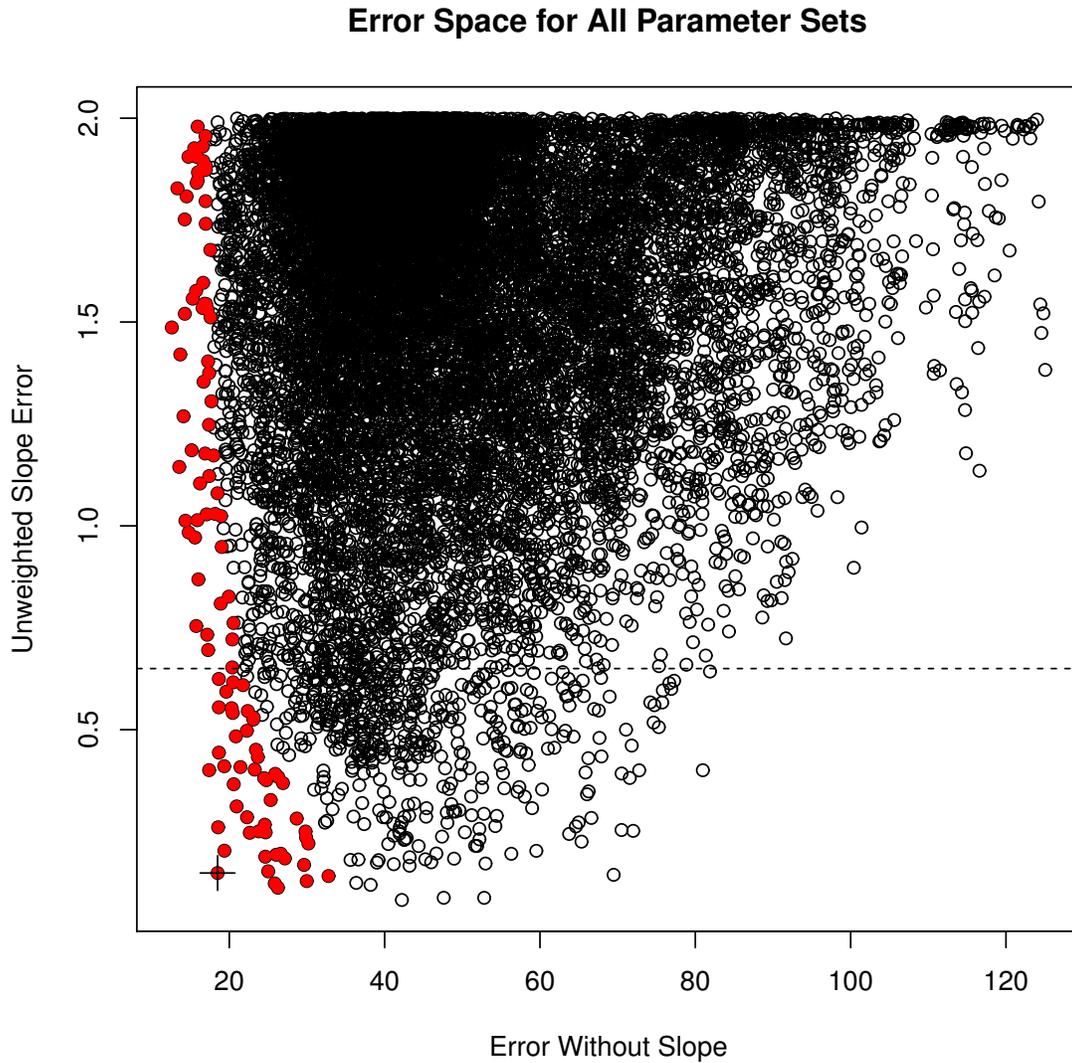


Figure 5.3: Slope error (Equation 5.5) plotted against the total error intercept (Equation 5.7; i.e. the sum of the other error terms). The parameter sets that make it to the top 50 for any weight  $0 \leq \varphi \leq 25$  are plotted in red. The dashed line separates the cluster that performs well in all areas but slope error from the cluster that yields lower slope errors but poorer performance in others. The special parameter set  $\theta_\varphi$  is identified with a cross.

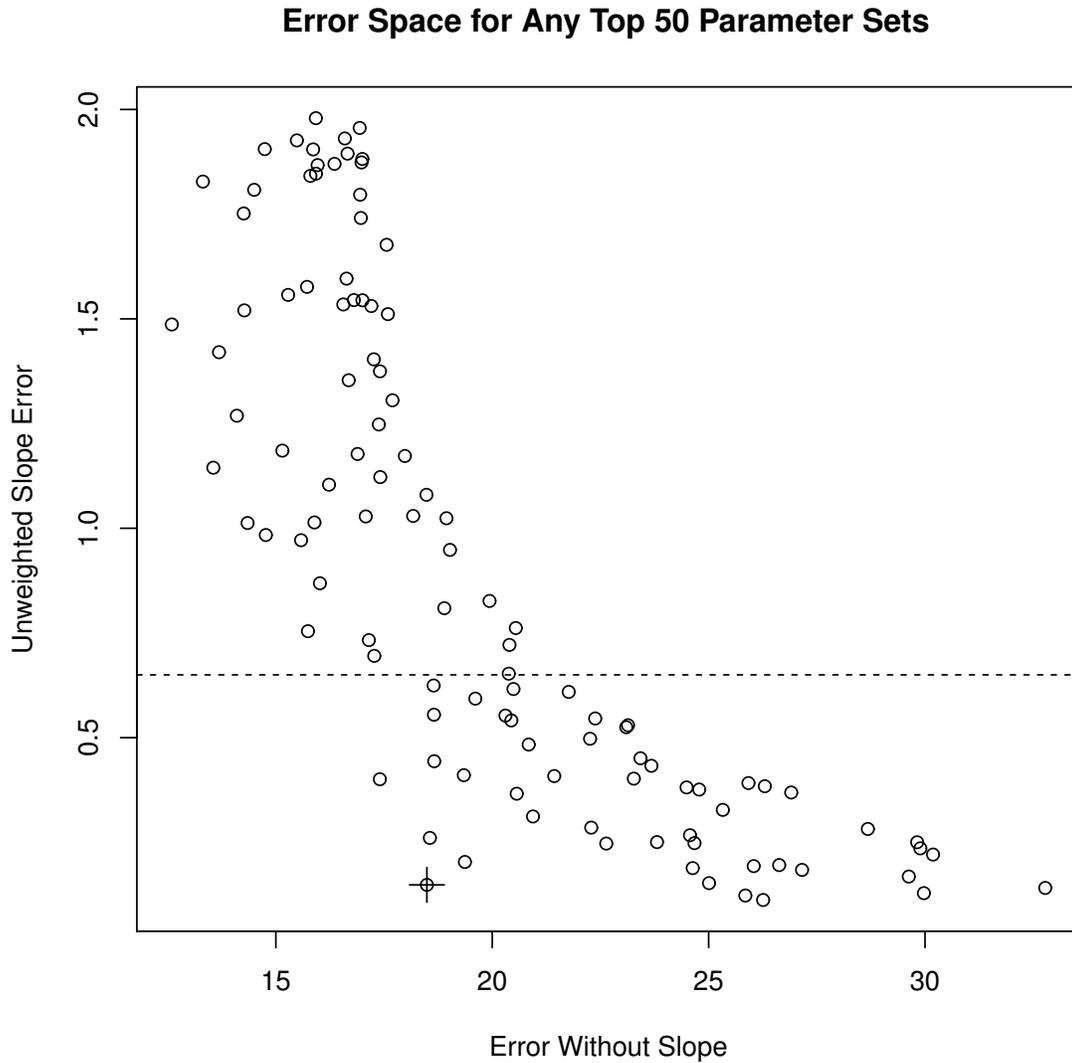


Figure 5.4: Those parameter sets that make it to the top 50 for any weight  $0 \leq \varphi \leq 25$  (the red points in Figure 5.3). Slope error (Equation 5.5) is plotted against the total error intercept (Equation 5.7; i.e. the sum of the other error terms). The dashed line separates the cluster that performs well in all areas but slope error from the cluster that yields lower slope errors but poorer performance in others. The special parameter set  $\theta_\varphi$  is identified with a cross.

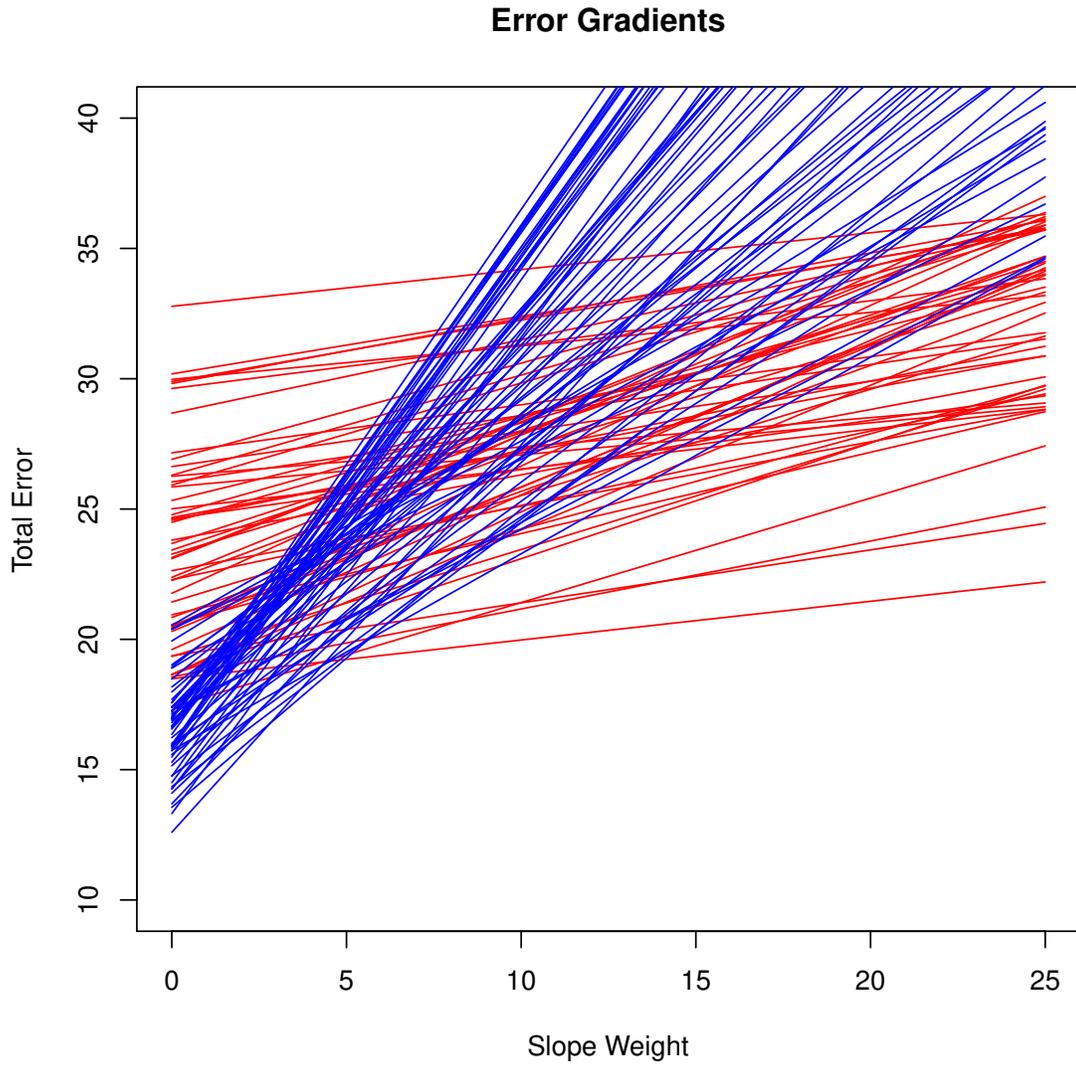


Figure 5.5: Figure 5.2 recolored to demonstrate the two clusters separated by the dashed line in Figures 5.3 & 5.4.

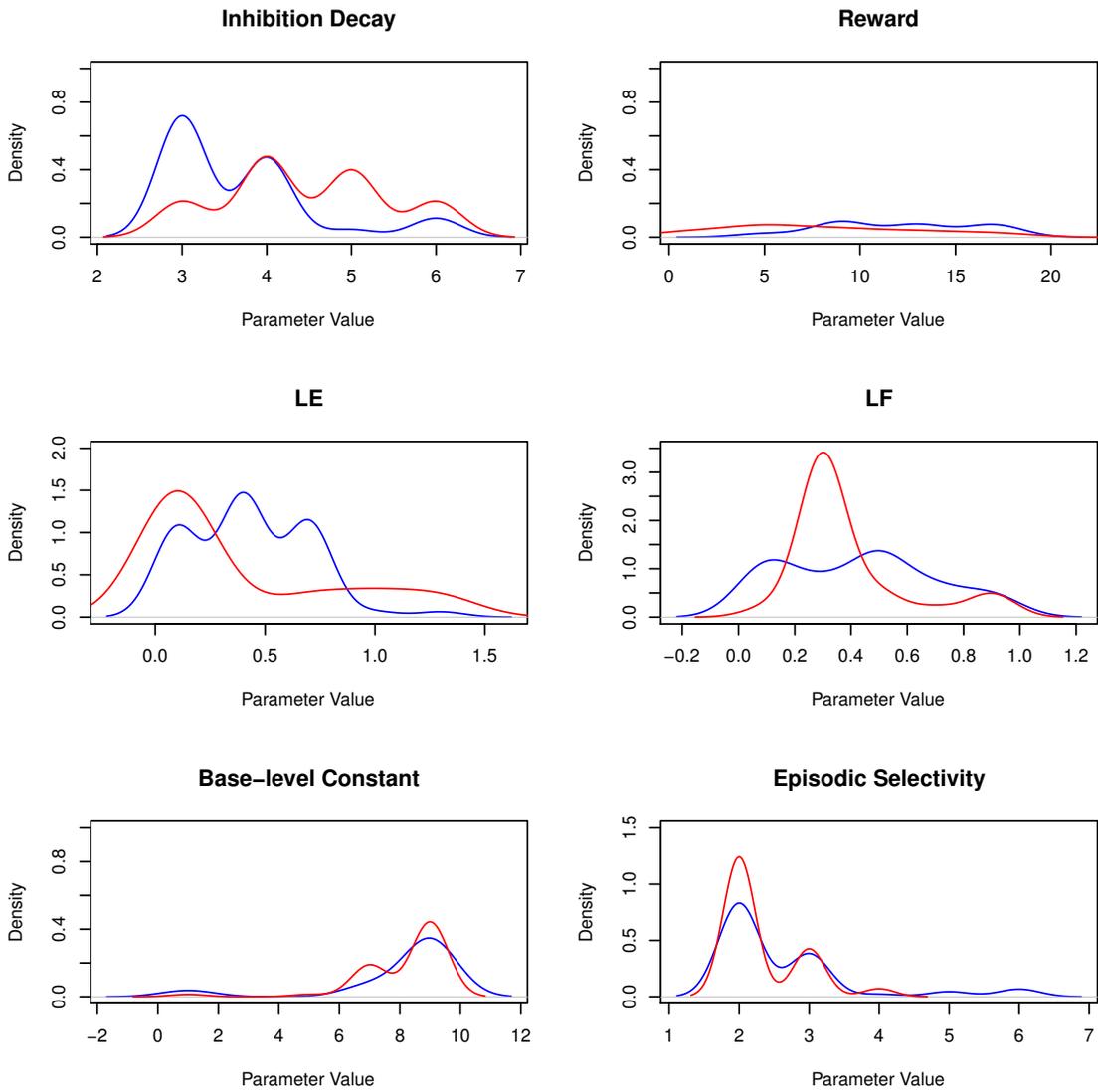


Figure 5.6: Estimated probability density functions for each free parameter. Red indicates the cluster that performs better in terms of slope. Blue indicates the cluster that performs best in the other areas, disregarding slope error.

form and continuous form, respectively, Figure 5.6 essentially portrays the relative frequency of a particular parameter value in the best fitting parameter sets, and Figure 5.7 depicts the values for which a particular parameter are best for a given  $\varphi$ . They mostly tell the same story; in the case of the inhibition decay parameter, not only does the distribution of best fitting values shift toward higher values with increased  $\varphi$  (Figure 5.6a), but Figure 5.7a demonstrates that the best values for this parameter ( $\gamma_d = 5$ ) also shift toward higher values as greater confidence is placed upon the regression. Contrast this trend with the latency exponent parameter, which is distributed more narrowly around lower values for the sets that survive longer in the top as  $\varphi$  increases (Figure 5.6c), but strongly favor a higher value in terms of total fit ( $f = 1.0$ ; Figure 5.7c).

From this analysis, two parameter sets of interest emerge. The combination that dominates a major portion of the error functions (indicated by the lowermost line in Figure 5.2 and by the cross in Figures 5.3 & 5.4) is hereby denoted  $\theta_\varphi$  and represents a moderate weighting of the regression of mean span on mean CL relative to the other DVs. The parameter set with the least  $\mathcal{T}_0$  reflects the hypothesis that the trend between span and CL does not need to be directly selected for (i.e.  $\varphi = 0$ ) and that fitting to the DVs contributing to this relationship alone will be sufficient for recovering it. This parameter set will be denoted  $\theta_0$ . A third parameter set is of interest because it allows for the reverse inference probed by  $\theta_0$ . This set, hereby denoted  $\theta_{RMSE}$ , reflects the conviction that the relationship between span and CL found in the human data is all that matters. This set minimizes the root mean square error between the model's mean span scores and the spans predicted by applying the regression model of the human data to the model's mean CL scores (Equation 5.8). Specific parameter values for each  $\theta$  set can be found in Table 5.1. Correlations among error metrics and fit measures are provided in Table 5.2. It is noteworthy that  $\mathcal{T}_0$  is much more strongly correlated with RT, TPT, and accuracy measures than with span,

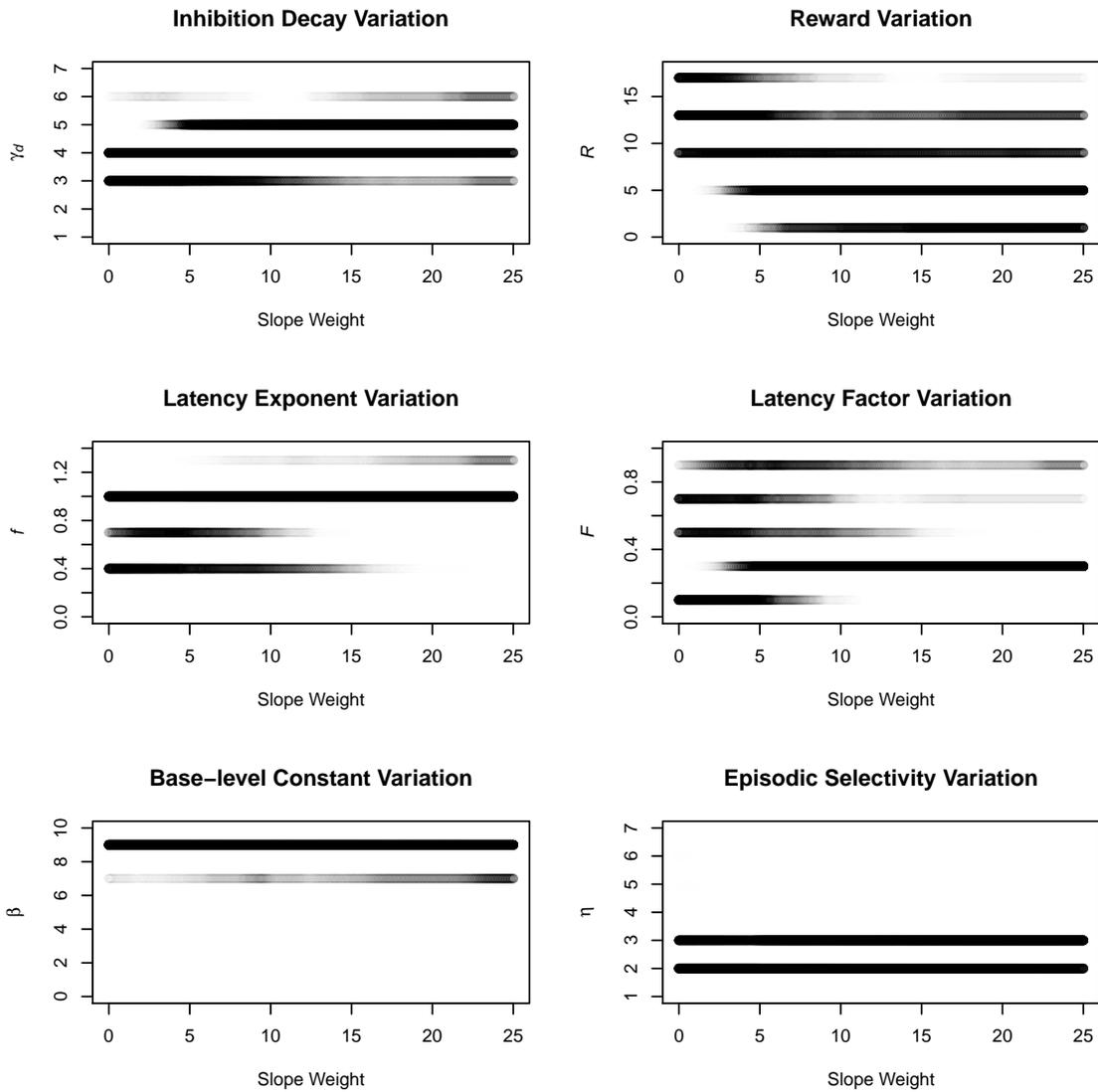


Figure 5.7: Parameter frequencies for the 50 best fitting parameter sets as a function of slope weight. The rank of each set is depicted by the opacity of its plotting character with darker points indicating relatively better fitting parameter sets.

Table 5.1: Parameter values associated with best model fits

Parameter Set	$R$	$\gamma_d$	$\eta$	$\beta$	$f$	$F$
$\theta_0$	13	4	3	9	0.1	0.1
$\theta_\varphi$	5	5	3	9	0.1	0.3
$\theta_{RMSE}$	9	5	2	9	0.1	0.3

Table 5.2: Misfit measure correlations

	1	2	3	4	5	6
1. Span Error						
2. RT Error	0.23					
3. TPT Error	0.15	0.82				
4. Accuracy Error	0.16	0.60	0.90			
5. Slope Error	0.45	-0.04	-0.12	-0.15		
6. $\mathcal{T}_0$	0.54	0.80	0.90	0.86	0.06	
7. RMSE	0.95	0.25	0.28	0.31	0.44	0.62

while the RMSE score is more sensitive to the span and slope errors than the RT, TPT, and accuracy errors. The performance of each set of interest is compared to human performance in Figures 5.8–5.22.

$$RMSE = \sqrt{\frac{1}{2} \sum_{i=1}^6 \left( \bar{s}_i - \left( \frac{1}{6900} \cdot \overline{CL}_i \cdot \hat{b}_i + \hat{a}_i \right) \right)^2} \quad (5.8)$$

While Figures 5.8–5.22 showcase the correspondence between model and human DV means, it remains to be seen whether the variance produced by the model matches that observed in the human data. Table 5.3 provides the results of a series of  $F$ -tests to accompany these figures. It is noteworthy that the model tends to be less variable than the human data in terms of RT and TPT over span or slope. Additionally, the model grows less variable than the human data with increased emphasis on fitting the regression of span on CL; the set  $\theta_{RMSE}$  produces results that are significantly less variable than the human data across the board.

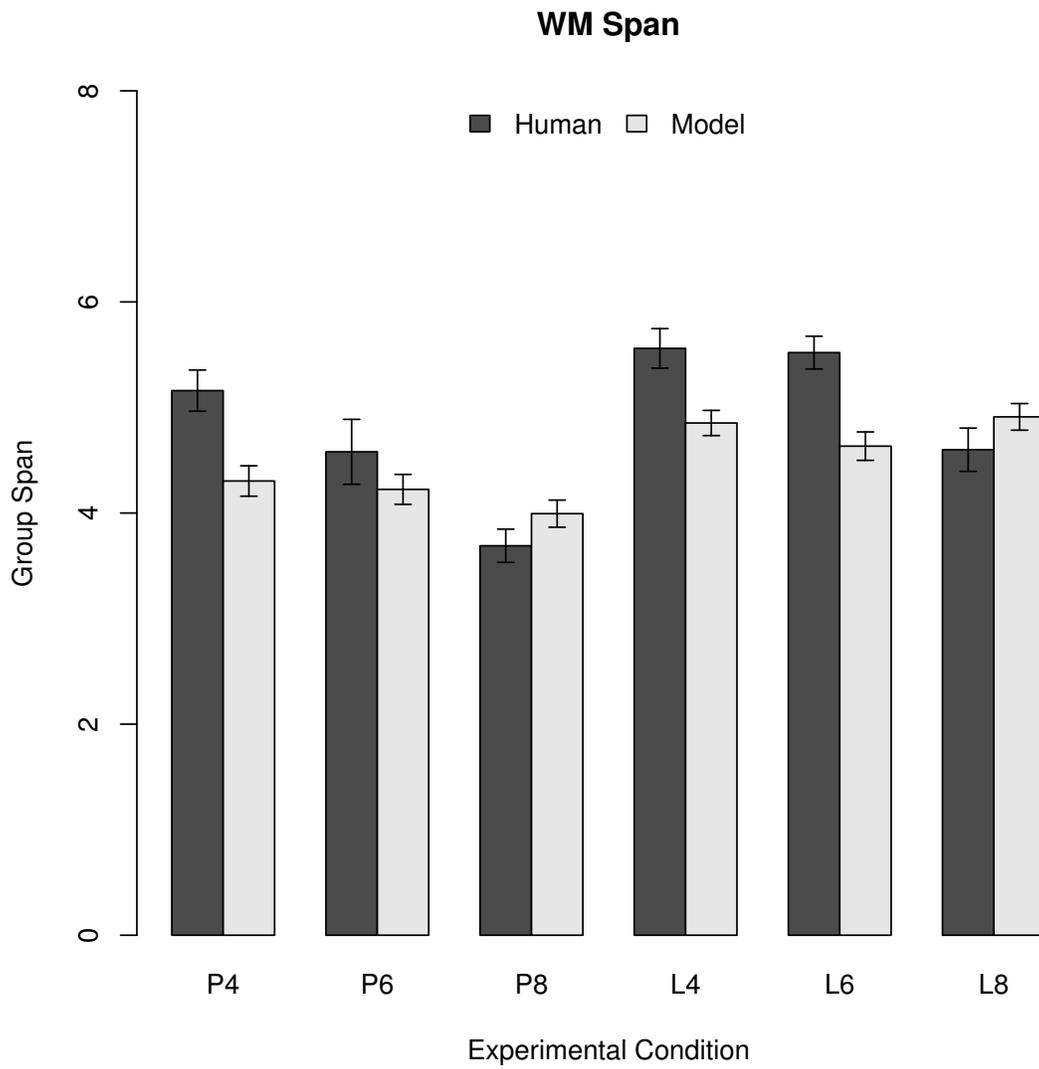


Figure 5.8: Comparison of model performance to human performance with respect to mean span using the  $\theta_0$  parameterization.

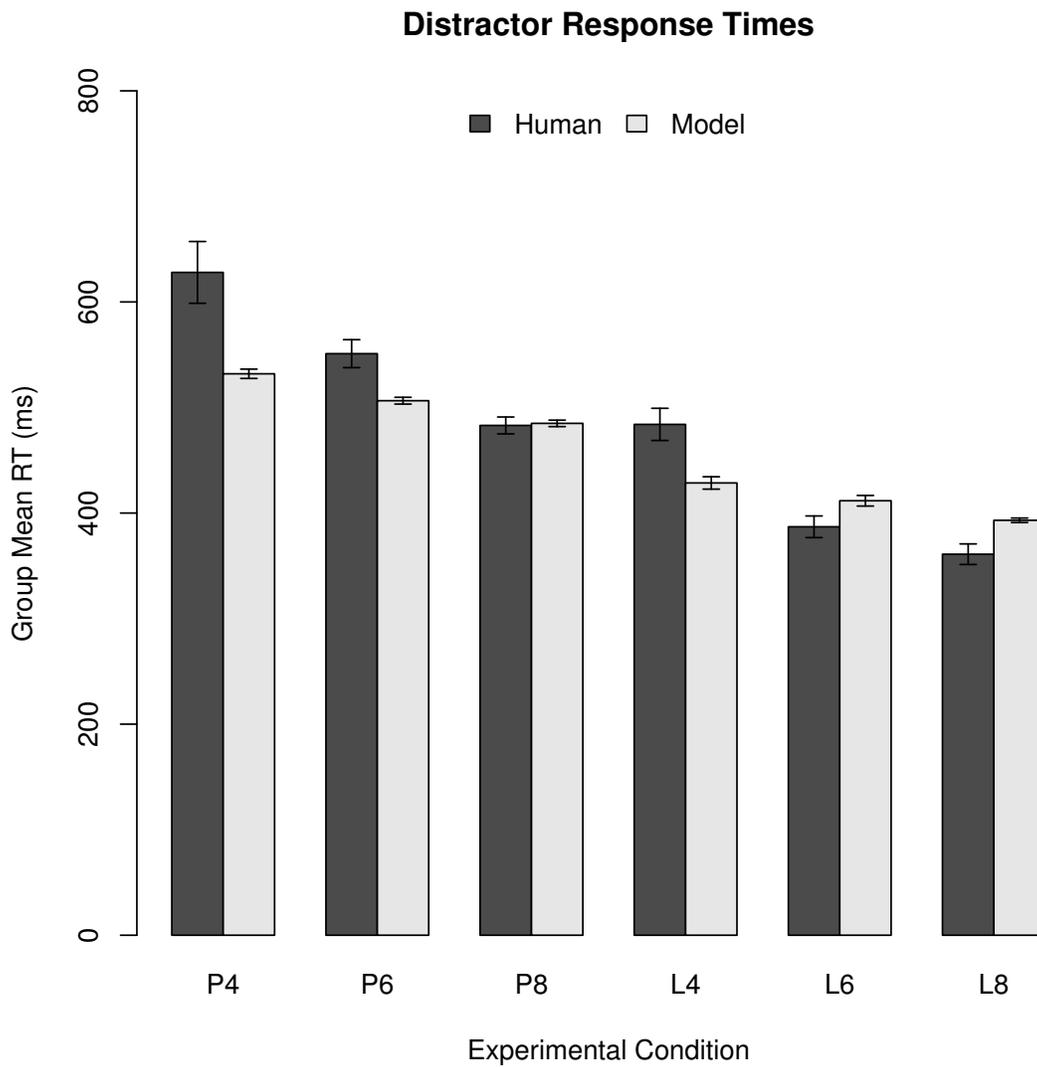


Figure 5.9: Comparison of model performance to human performance with respect to mean RT using the  $\theta_0$  parameterization.

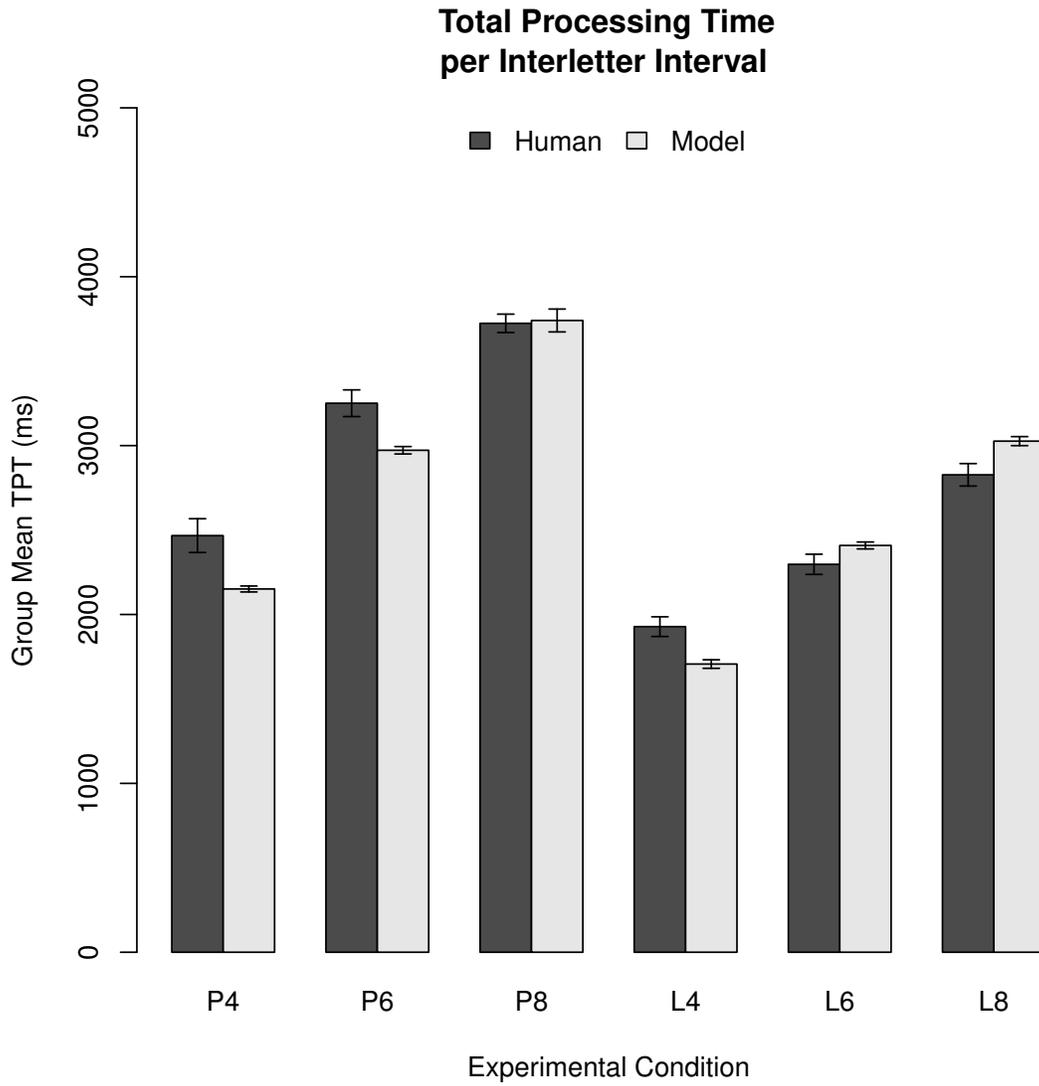


Figure 5.10: Comparison of model performance to human performance with respect to mean TPT using the  $\theta_0$  parameterization.

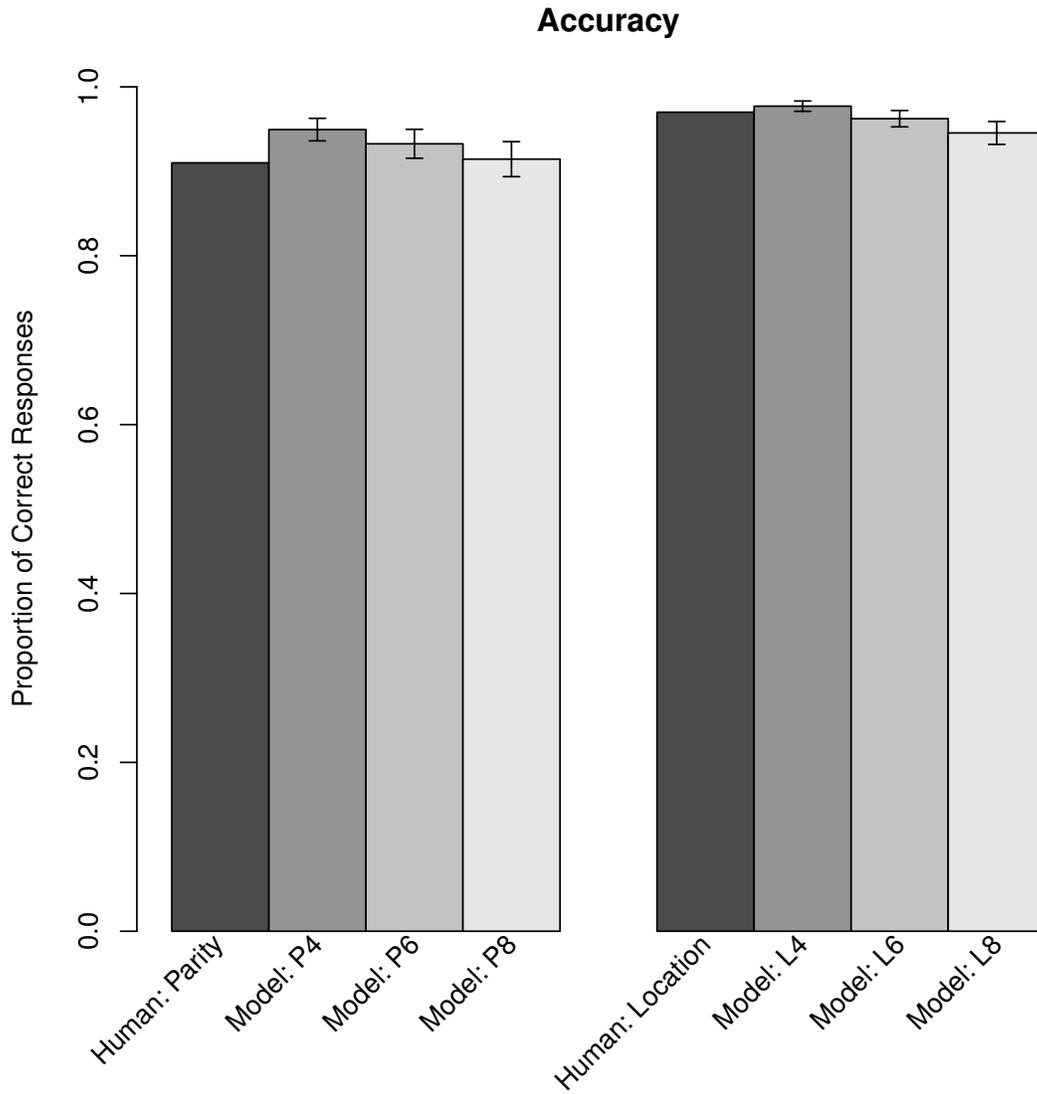


Figure 5.11: Comparison of model performance to human performance with respect to mean accuracy using the  $\theta_0$  parameterization.

### Effect of CL on Span

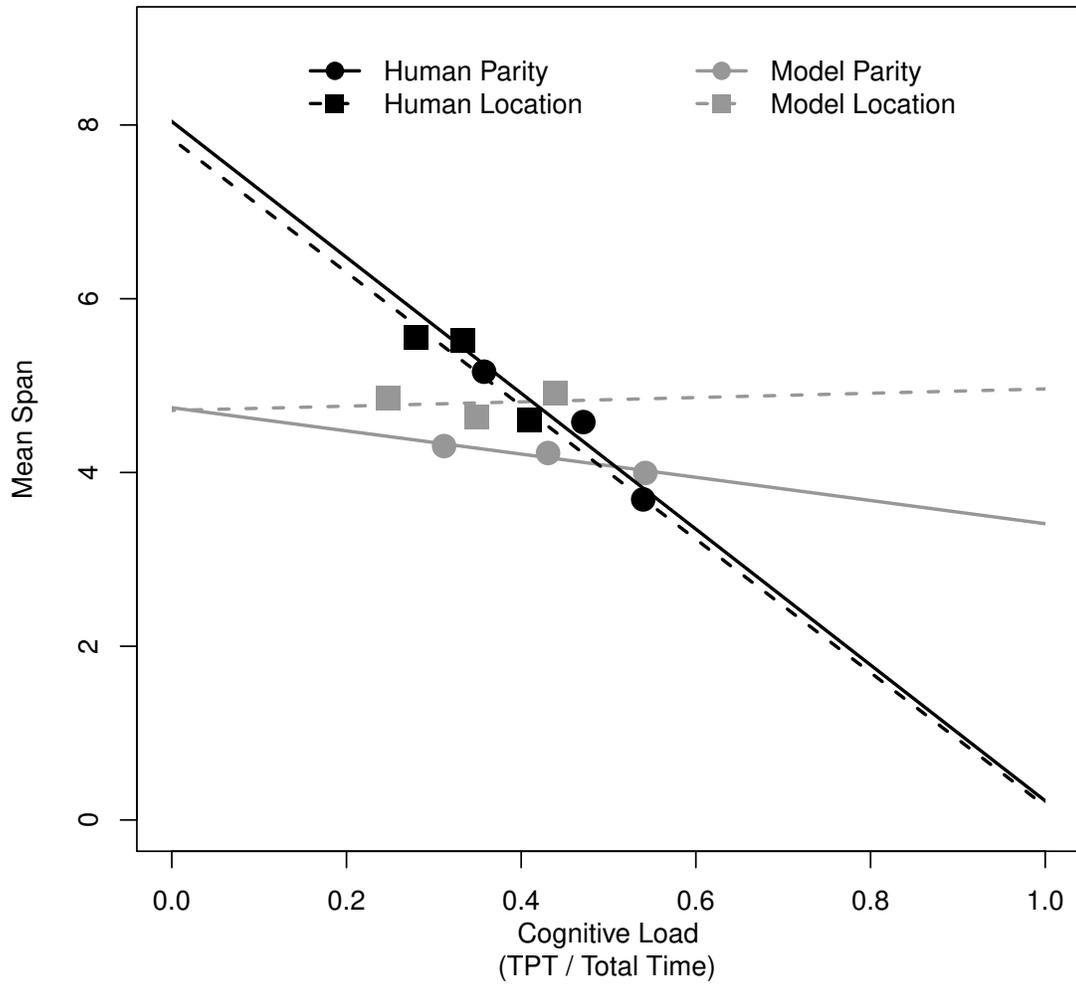


Figure 5.12: Comparison of model performance to human performance with respect to mean span regression on cognitive load using the  $\theta_0$  parameterization.

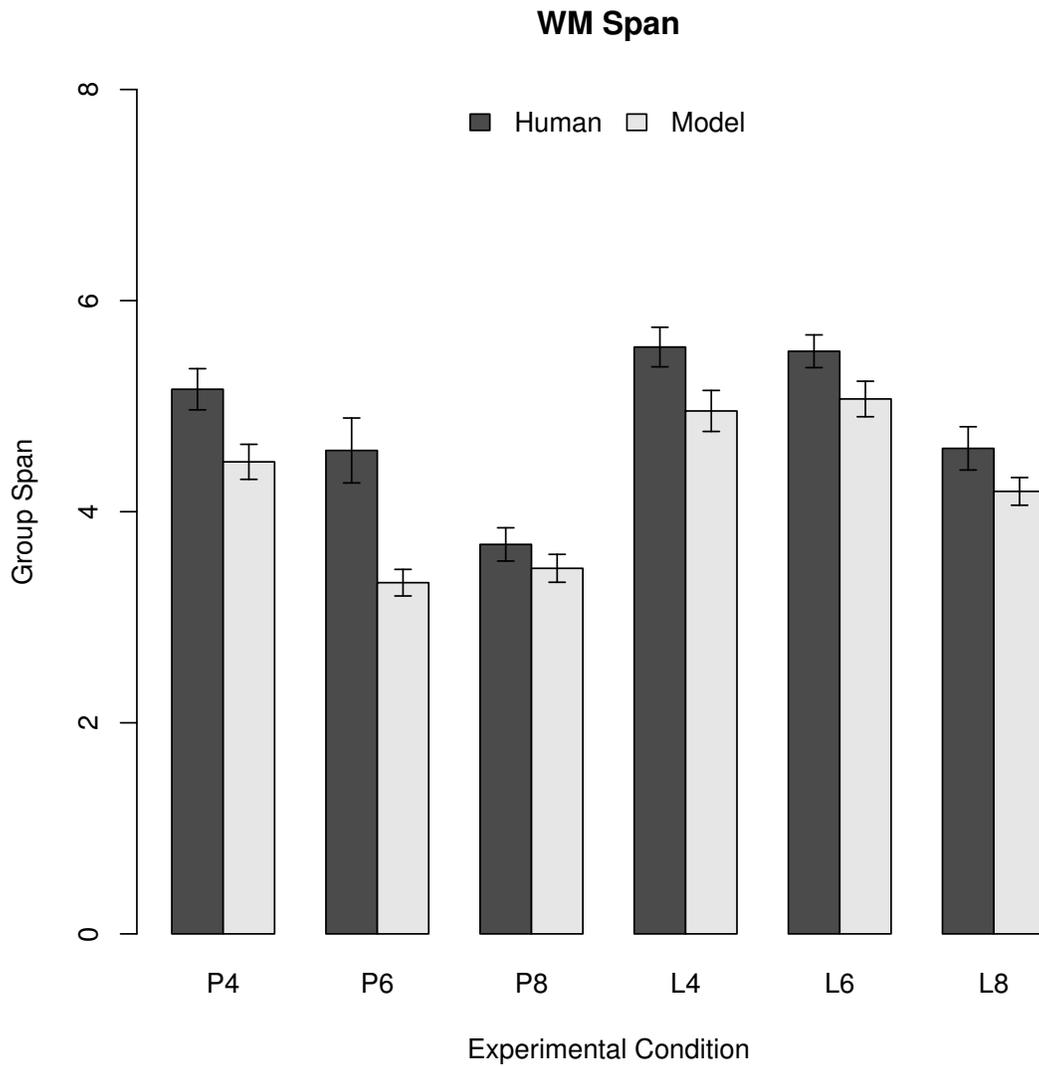


Figure 5.13: Comparison of model performance to human performance with respect to mean span using the  $\theta_\varphi$  parameterization.

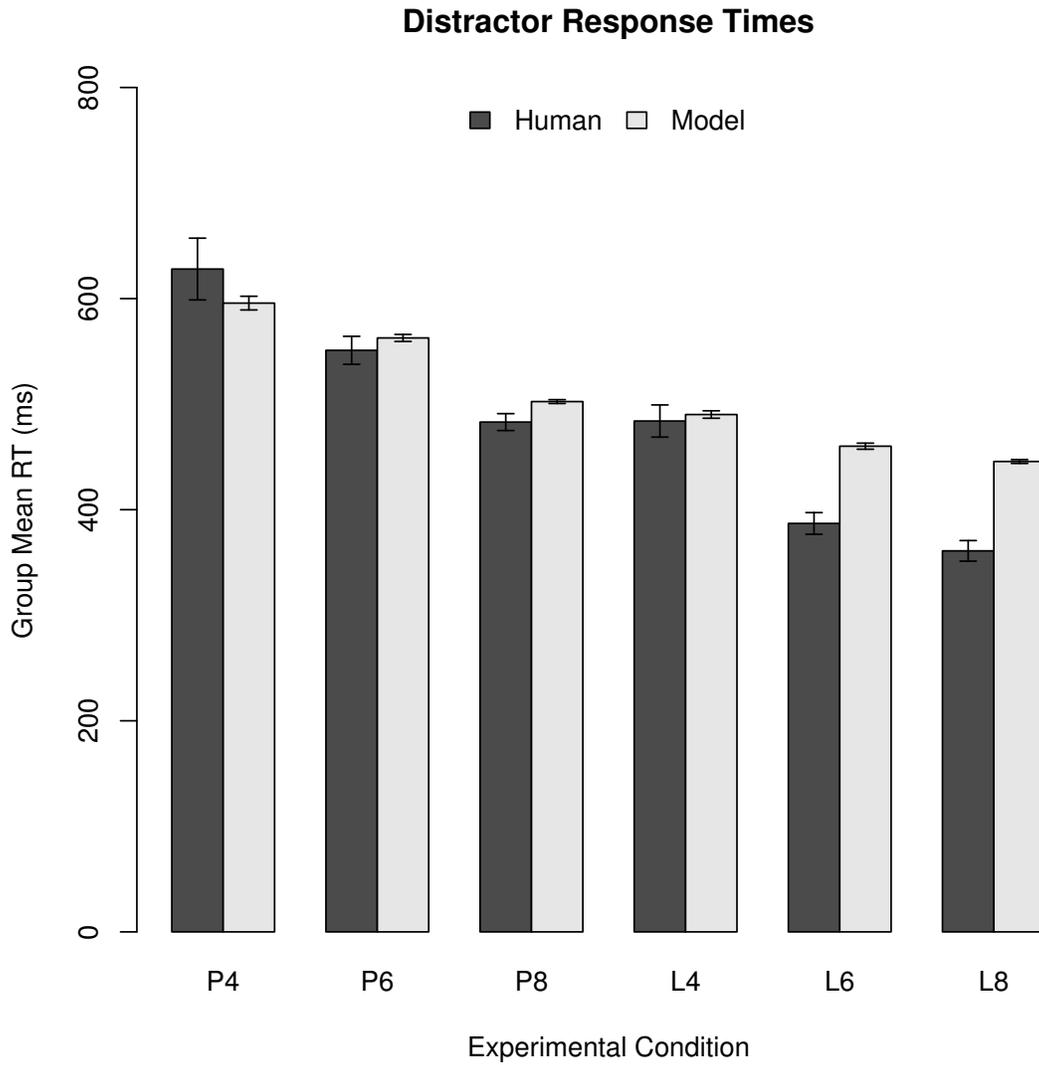


Figure 5.14: Comparison of model performance to human performance with respect to mean RT using the  $\theta_\varphi$  parameterization.

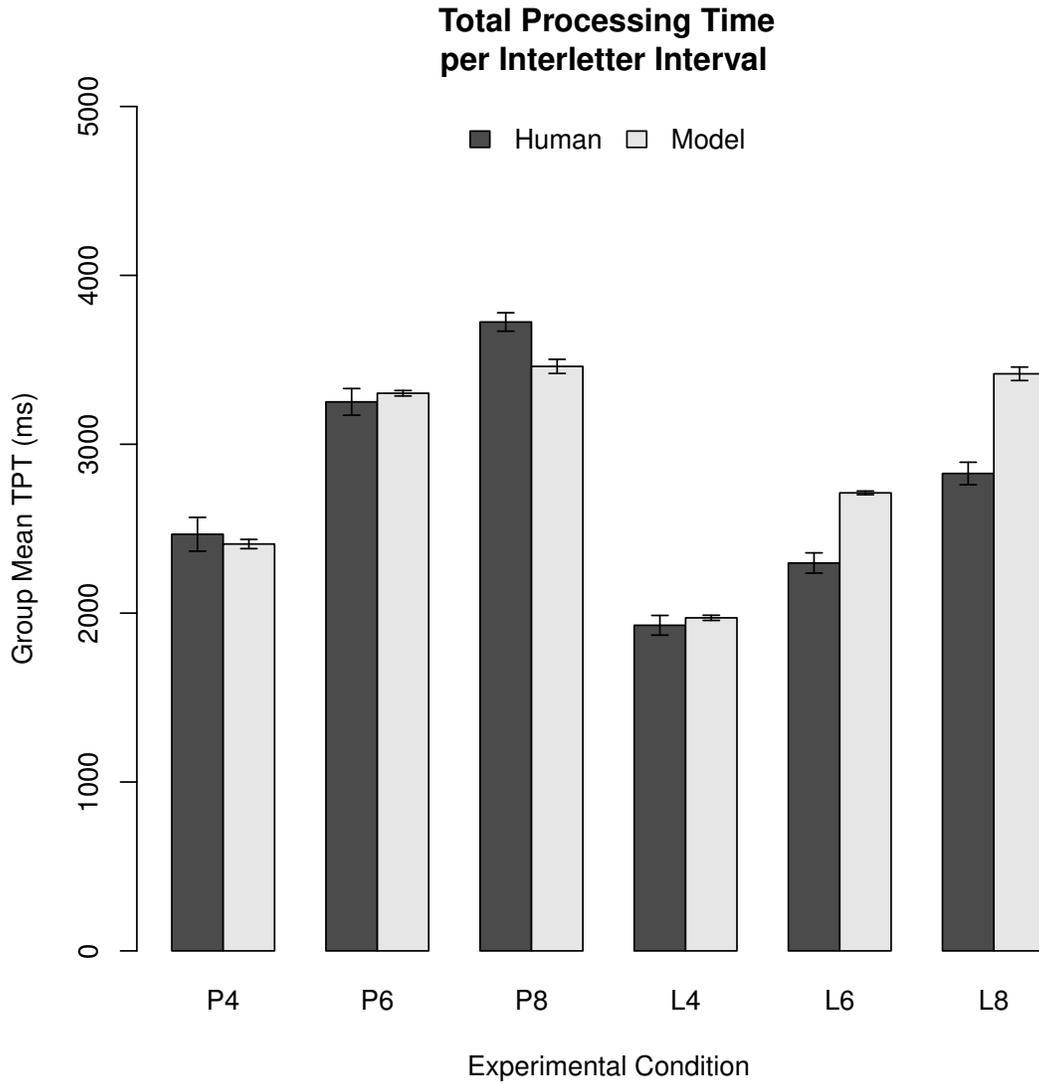


Figure 5.15: Comparison of model performance to human performance with respect to mean TPT using the  $\theta_\varphi$  parameterization.

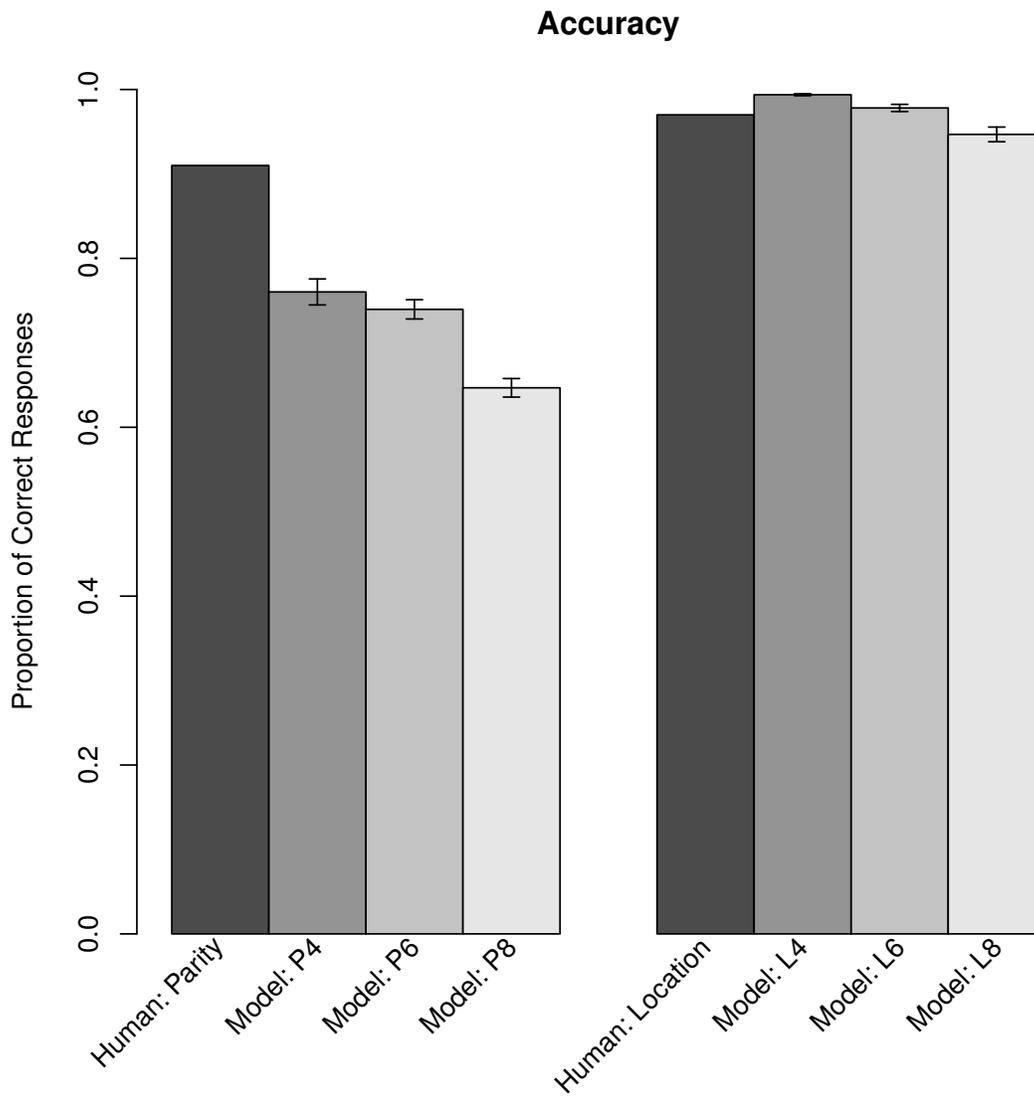


Figure 5.16: Comparison of model performance to human performance with respect to mean accuracy using the  $\theta_\varphi$  parameterization.

### Effect of CL on Span

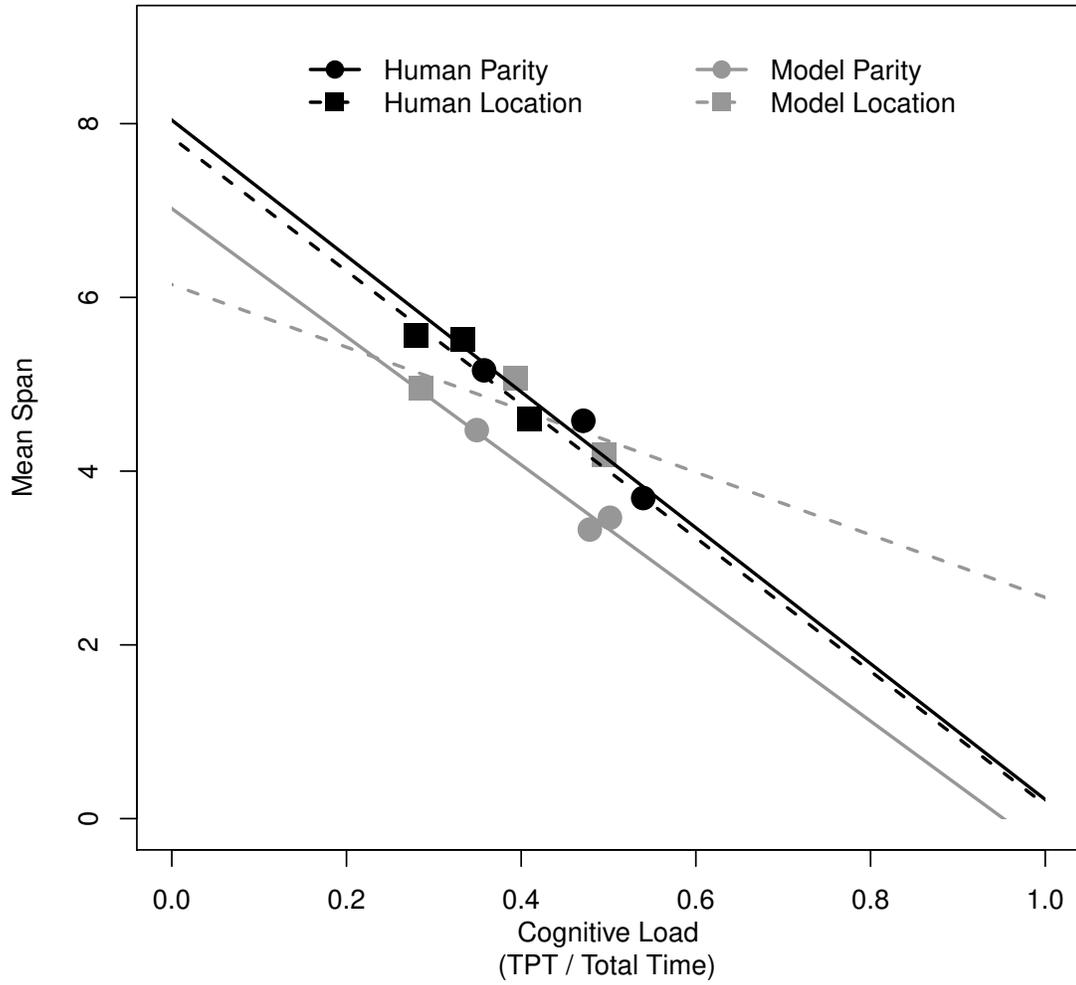


Figure 5.17: Comparison of model performance to human performance with respect to mean span regression on cognitive load using the  $\theta_\varphi$  parameterization.

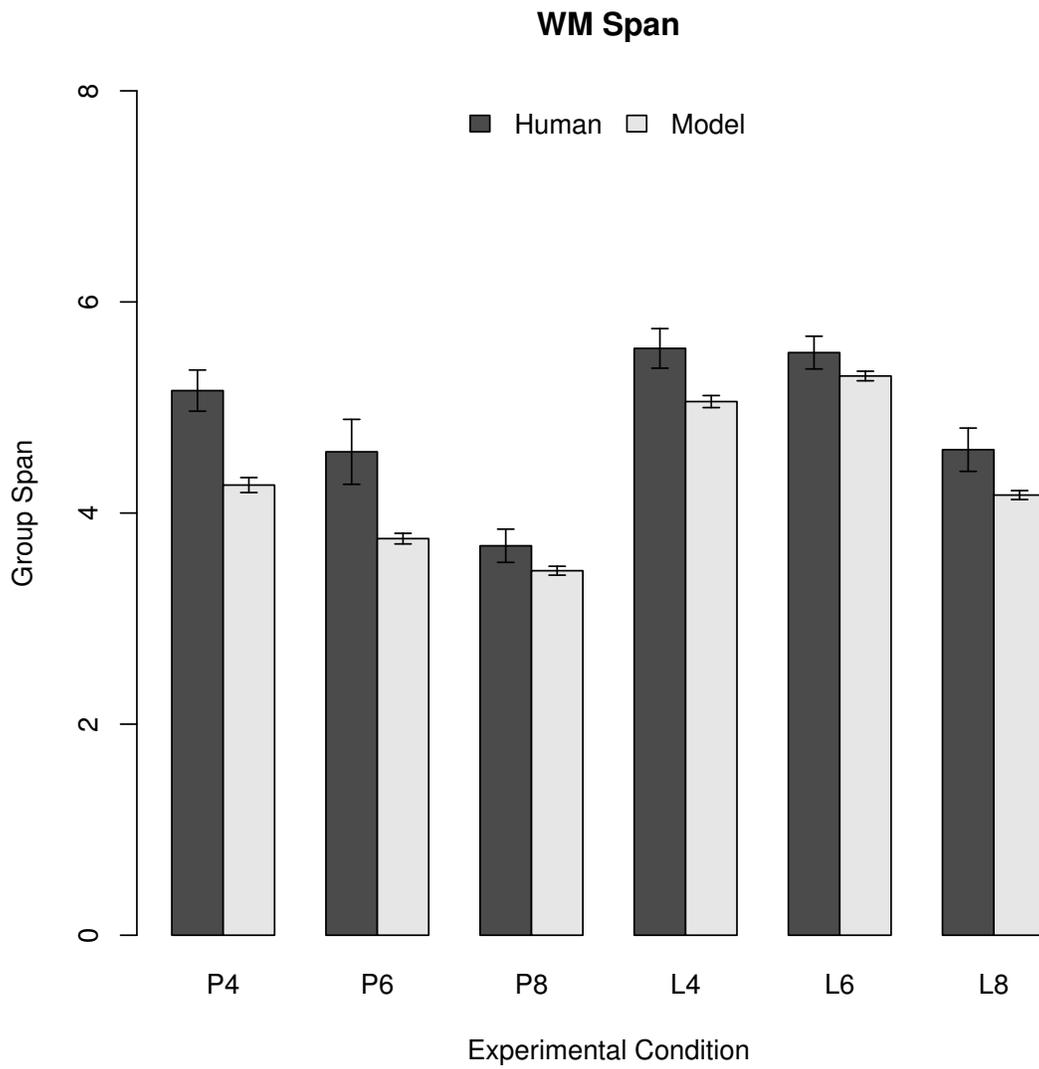


Figure 5.18: Comparison of model performance to human performance with respect to mean span using the  $\theta_{RMSE}$  parameterization.

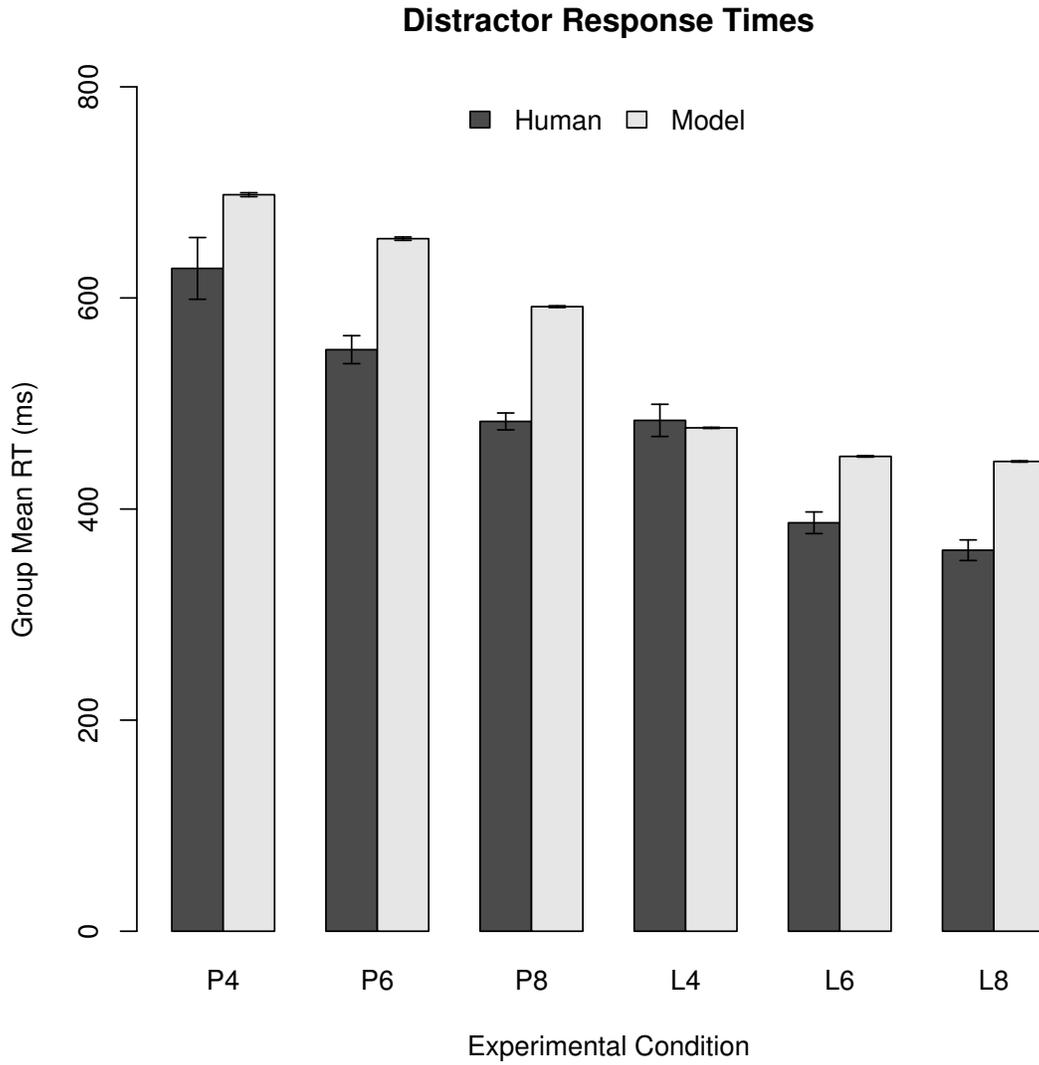


Figure 5.19: Comparison of model performance to human performance with respect to mean RT using the  $\theta_{RMSE}$  parameterization.

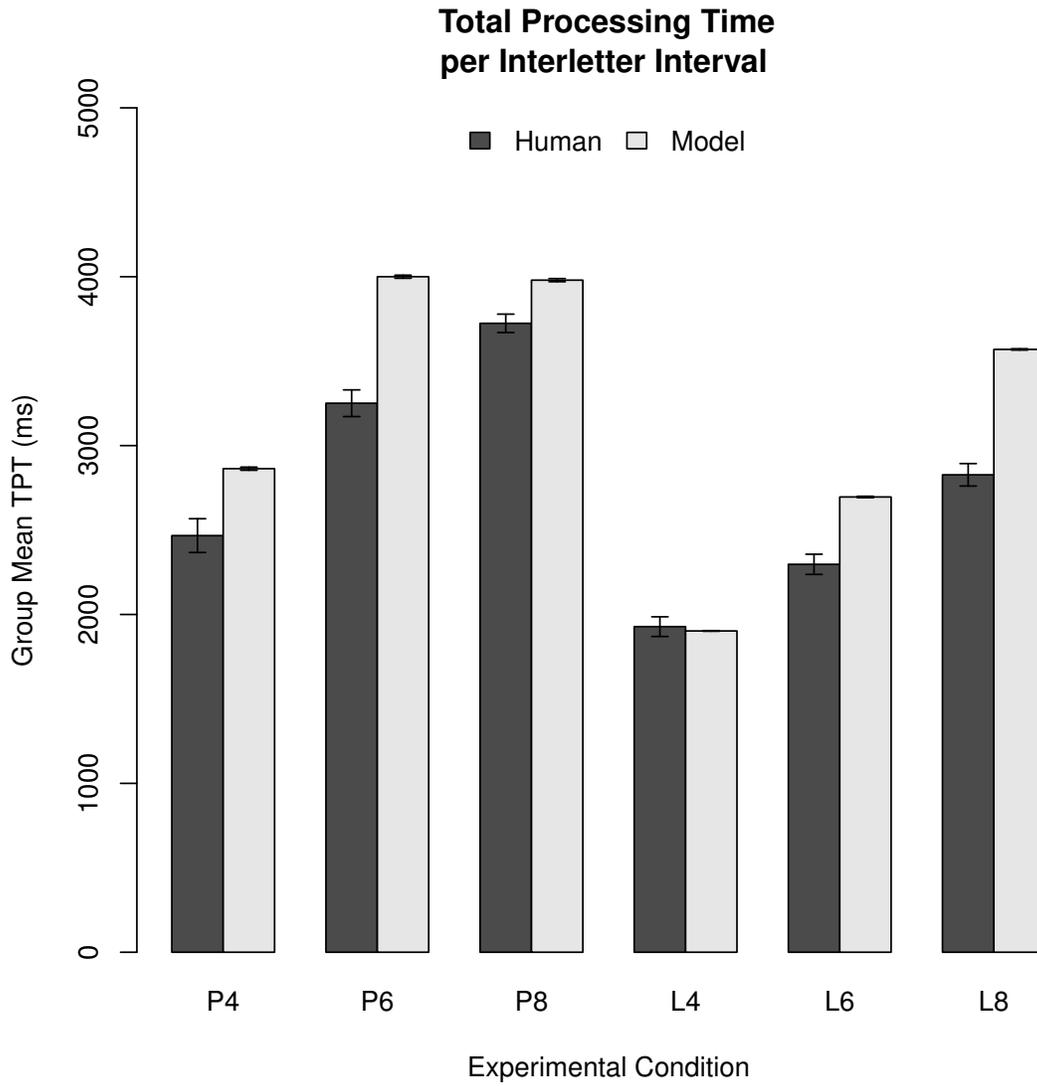


Figure 5.20: Comparison of model performance to human performance with respect to mean TPT using the  $\theta_{RMSE}$  parameterization.

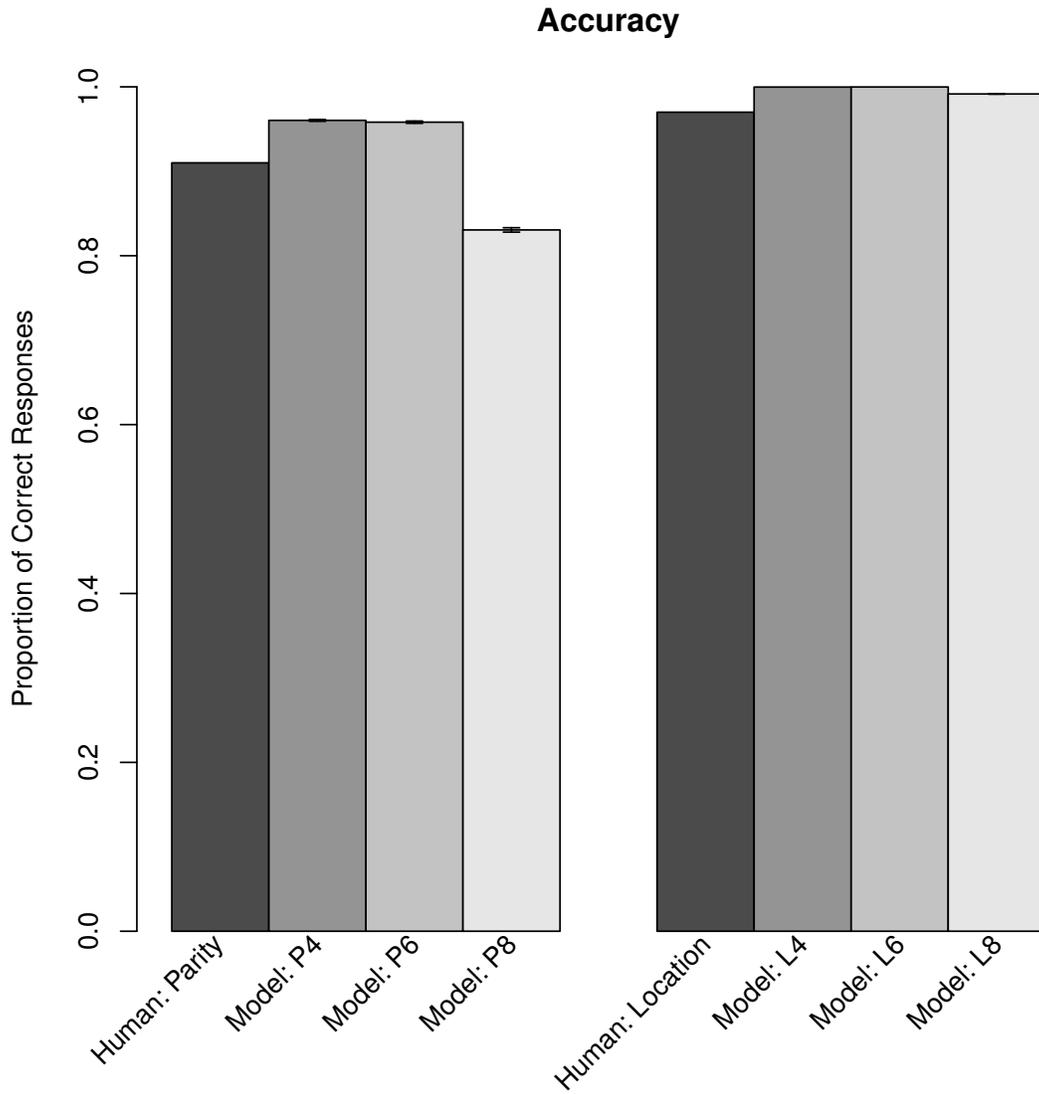


Figure 5.21: Comparison of model performance to human performance with respect to mean accuracy using the  $\theta_{RMSE}$  parameterization.

### Effect of CL on Span

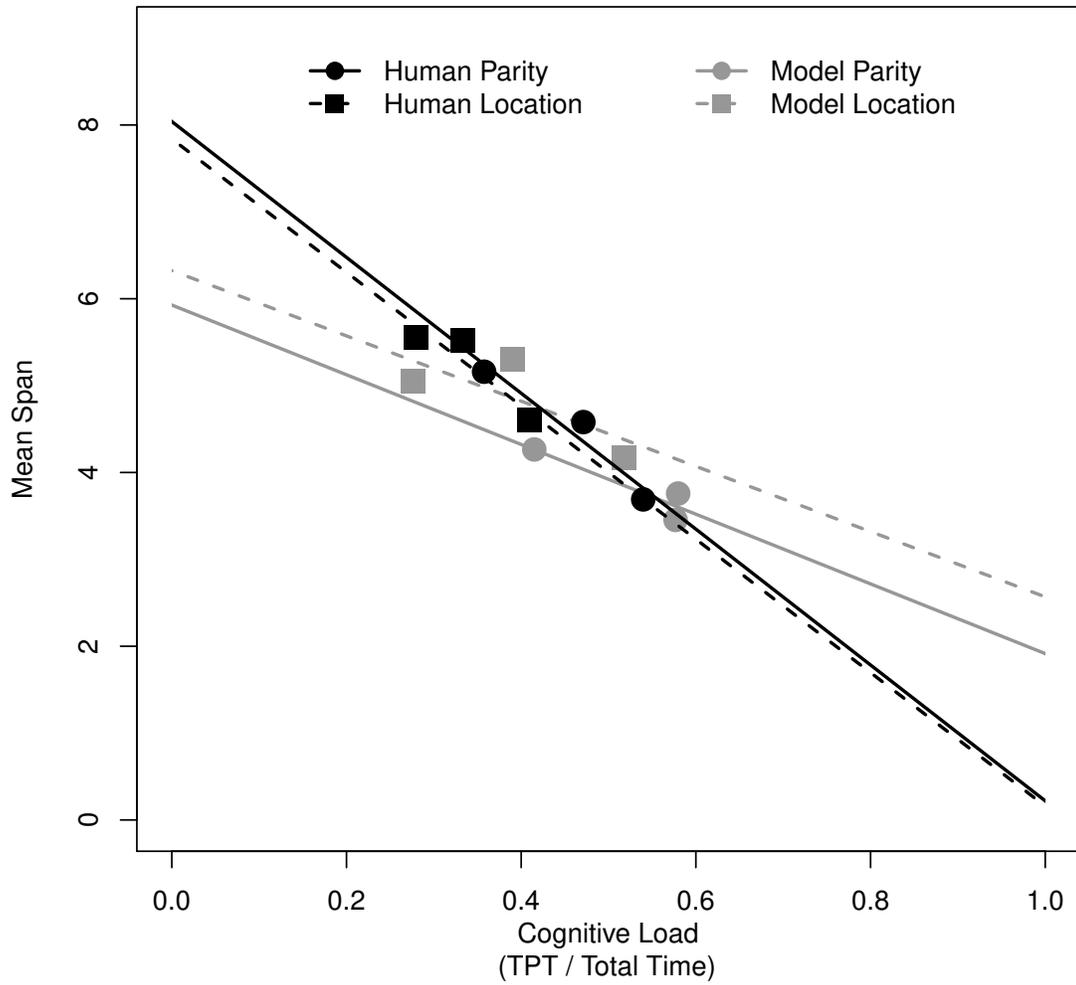


Figure 5.22: Comparison of model performance to human performance with respect to mean span regression on cognitive load using the  $\theta_{RMSE}$  parameterization.

Table 5.3:  $F$ -tests

Measure	Condition		$\theta_0$		$\theta_\varphi$		$\theta_{RMSE}$	
	Task	Pace	$F$	$p$	$F$	$p$	$F$	$p$
Span	Parity	4	0.59	.871	0.44	.957	2.42	.010
		6	1.49	.145	1.89	.048	11.82	< .001
		8	0.47	.943	0.45	.955	4.39	< .001
	Location	4	0.79	.686	0.30	.994	3.44	< .001
		6	0.43	.964	0.27	.996	3.74	< .001
		8	0.84	.628	0.78	.692	7.50	< .001
RT	Parity	4	14.04	< .001	6.48	< .001	79.88	< .001
		6	5.43	< .001	5.07	< .001	22.11	< .001
		8	2.19	.020	6.29	< .001	31.12	< .001
	Location	4	2.17	.021	6.02	< .001	271.11	< .001
		6	1.34	.213	4.02	< .001	84.96	< .001
		8	6.58	< .001	9.11	< .001	82.87	< .001
TPT	Parity	4	10.37	< .001	4.26	< .001	37.23	< .001
		6	4.12	< .001	7.65	< .001	24.05	< .001
		8	0.21	.999	0.54	.901	11.05	< .001
	Location	4	1.70	.081	4.62	< .001	323.72	< .001
		6	2.73	.004	9.90	< .001	66.57	< .001
		8	1.99	.036	0.89	.580	90.29	< .001
Slope	Parity		0.89	.655	0.77	.307	4.84	< .001
	Location		0.79	.347	0.43	.002	2.90	< .001

**Note.**  $F < 1$  indicates that the model is more variable than the human data.

## 5.3 Parametric Effects on Components of Model Misfit

To further explore the contribution of each parameter to model misfit, I conducted a series of multiple linear regressions for each error measure using the six free model parameters as predictors. I provide visualizations of the significant effects in Figures 5.23–5.30, showing the average difference between the model and human data for each of the six experimental conditions. Error bars indicate the standard error of this difference:  $SE_{M-H} = \sqrt{SE_M^2 + SE_H^2}$ . For simplicity I use the value from the  $\theta_\varphi$  set for marginalized parameters rather than aggregating over all parameter combinations.

### 5.3.1 Span

The negative interaction of the inhibition decay parameter  $\gamma_d$  and the base-level constant  $\beta$  suggests that the model fits significantly better in terms of span with greater temporal inhibition and larger base-level activation,  $B = -0.65$ ,  $t(15618) = -2.56$ ,  $p = .011$ . The three-way interaction between these two parameters and the episodic selectivity parameter  $\eta$  approached significance,  $B = 0.12$ ,  $t(15618) = 1.95$ ,  $p = .051$ ; however, speculative examination of this three-way interaction may help to better characterize the relationship between the inhibition decay parameter and the base-level constant. At the lowest level of the episodic selectivity parameter examined ( $\eta = 2$ ; Figure 5.23), the inhibition decay parameter and base-level constant clearly interact such that mean span increases with increased inhibition decay only for higher levels of the base-level constant. At lower to middle values ( $2 < \eta \leq 4$ ; Figures 5.24–5.25), the interaction between the inhibition decay parameter and base-level constant appears to become nonlinear with mean span increasing for extreme values of the base-level constant (i.e.  $\beta = 1, 9$ ) while middle values yield relatively lower mean spans. However, this trend only holds for lower to middle values of the inhibition decay parameter

( $\gamma_d \leq 4$ ), after which the dominance of greater  $\beta$  reemerges before dropping off itself. At larger values of the episodic selectivity parameter ( $\eta > 4$ ; Figures 5.26–5.27), the influence of both the inhibition decay parameter and the base-level constant become muddled so that any effect of either on mean span is impossible to determine.

### 5.3.2 RT

The latency factor parameter  $F$  significantly contributed to RT error,  $B = 15.04$ ,  $t(15618) = 3.11$ ,  $p = .002$ . The nearly significant interaction with the base-level constant ( $B = -1.65$ ,  $t(15618) = -1.95$ ,  $p = .051$ ) reflects a tradeoff between these parameters. While the effect of  $F$  dominantly increases mean RTs beyond that observed in the human data by increasing retrieval latencies, the effect of  $\beta$  counters this through an inverse relationship on retrieval latencies, possibly explaining the crossover at lower values of  $F$  (Figure 5.28).

### 5.3.3 TPT

The latency factor parameter alone significantly contributed to TPT error,  $B = 31.22$ ,  $t(15618) = 3.93$ ,  $p < .001$ . Inspection of Figure 5.29 reveals that the increased error is caused by a massive decrease in mean TPT as  $F$  increases.

### 5.3.4 Accuracy

The regression results concerning accuracy were very similar to those for TPT error. The latency factor parameter significantly contributed to accuracy error,  $B = 35.95$ ,  $t(15618) = 5.11$ ,  $p < .001$ . Similar to the previous finding, accuracy decreases considerably as  $F$  increases (Figure 5.30).

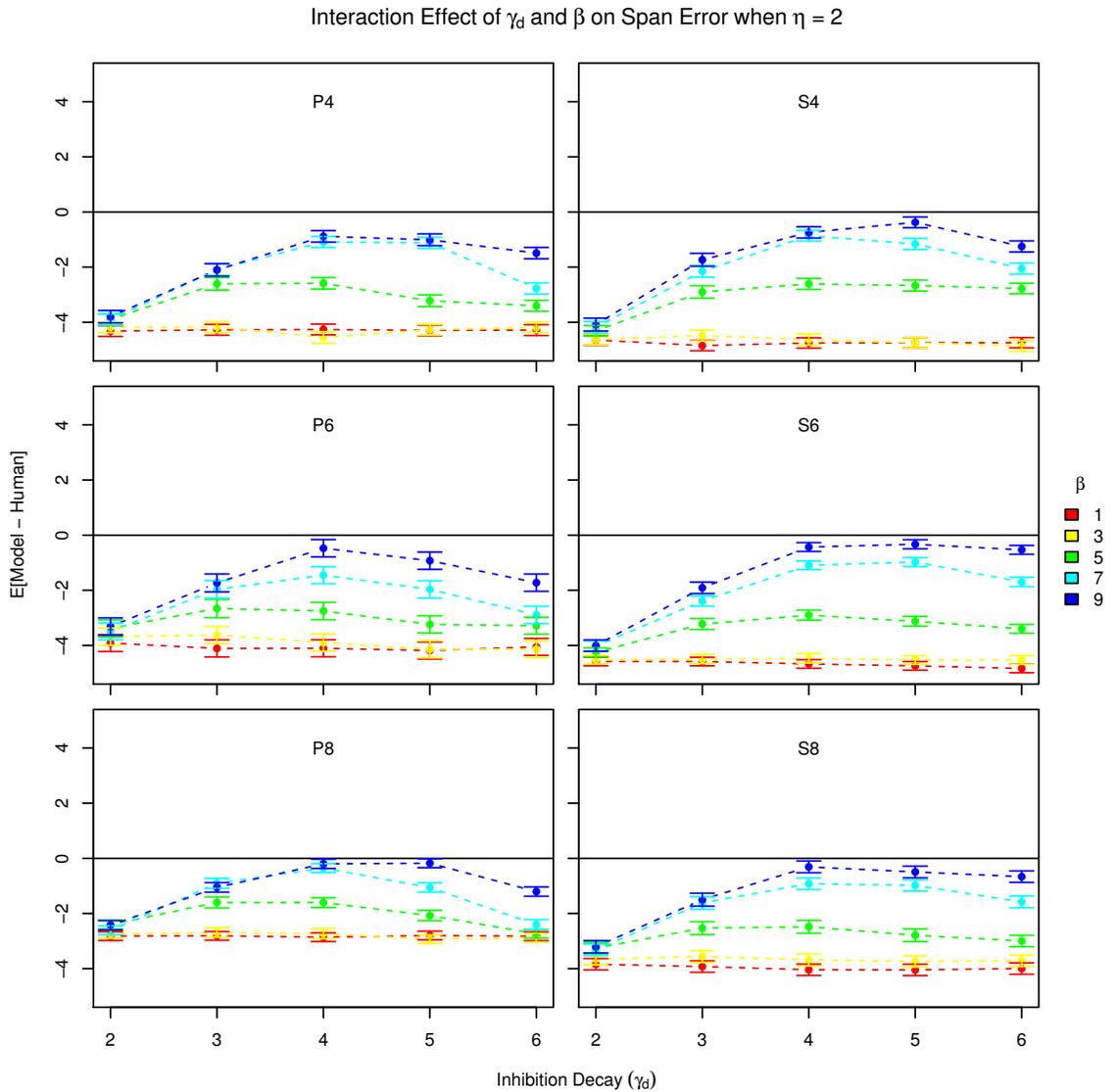


Figure 5.23: Difference between the model's and the humans' mean span scores as a function of the temporal inhibition decay parameter and the base-level constant across experimental conditions when  $\eta = 2$ . Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

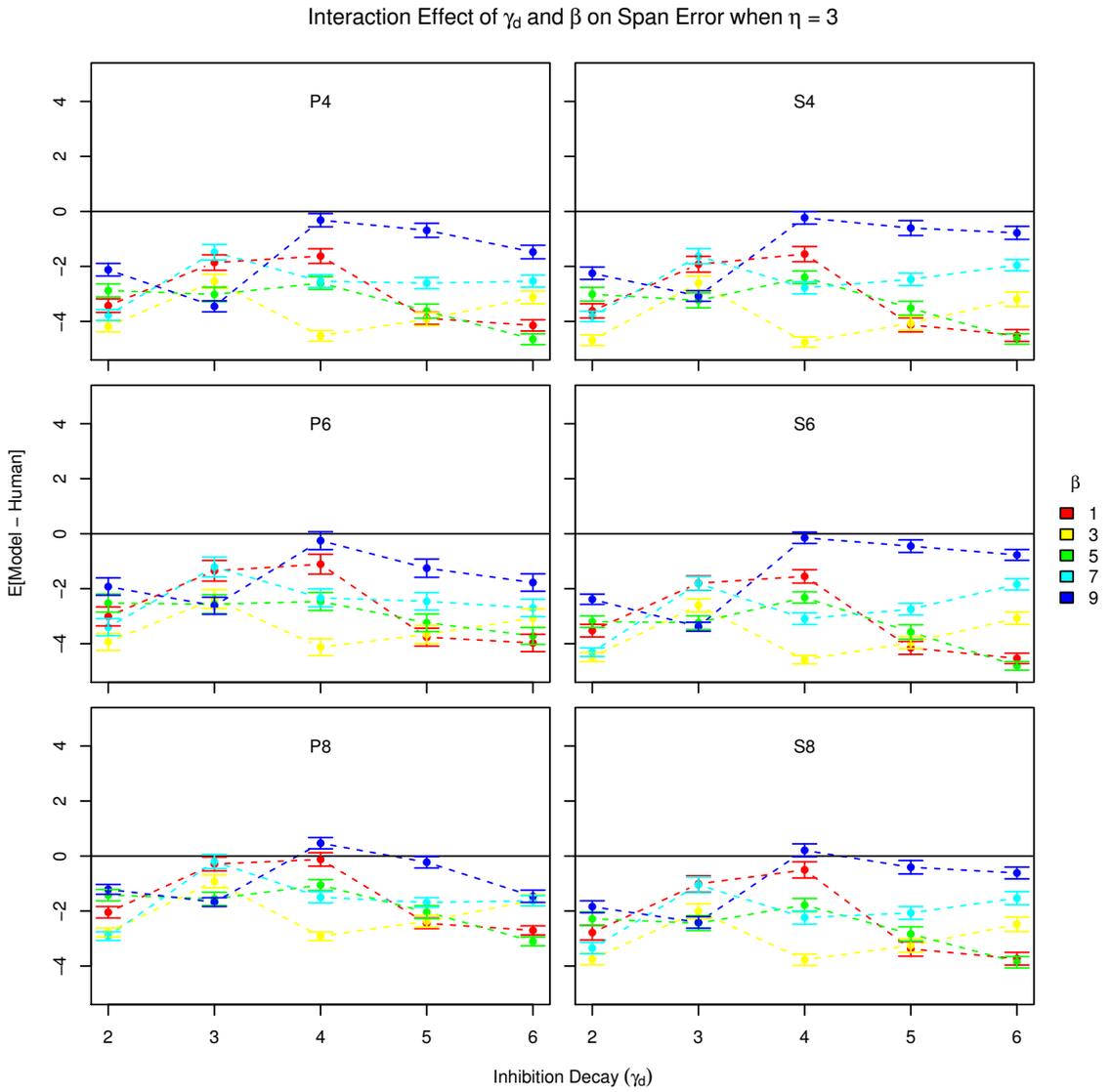


Figure 5.24: Difference between the model's and the humans' mean span scores as a function of the temporal inhibition decay parameter and the base-level constant across experimental conditions when  $\eta = 3$ . Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

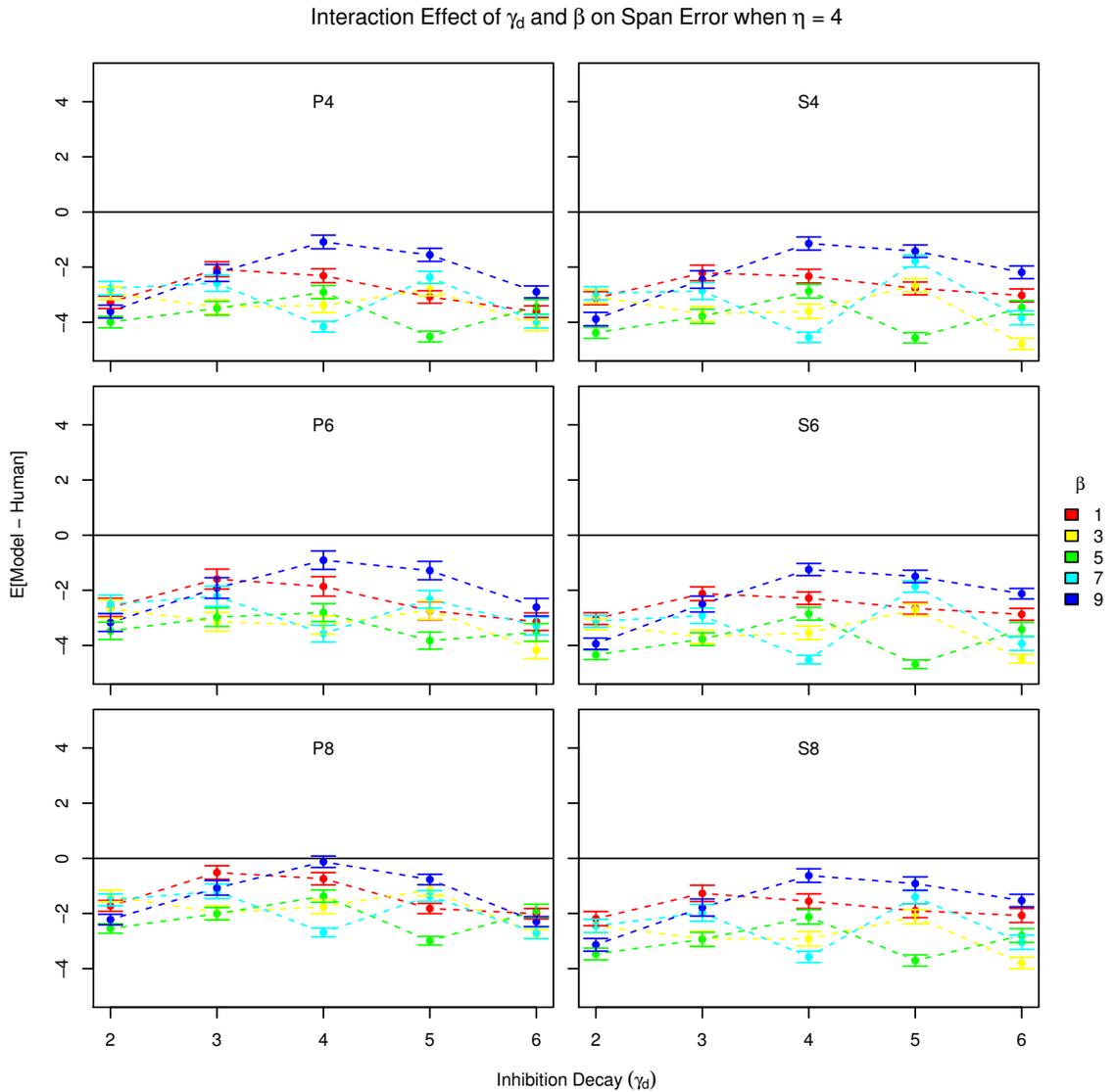


Figure 5.25: Difference between the model's and the humans' mean span scores as a function of the temporal inhibition decay parameter and the base-level constant across experimental conditions when  $\eta = 4$ . Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

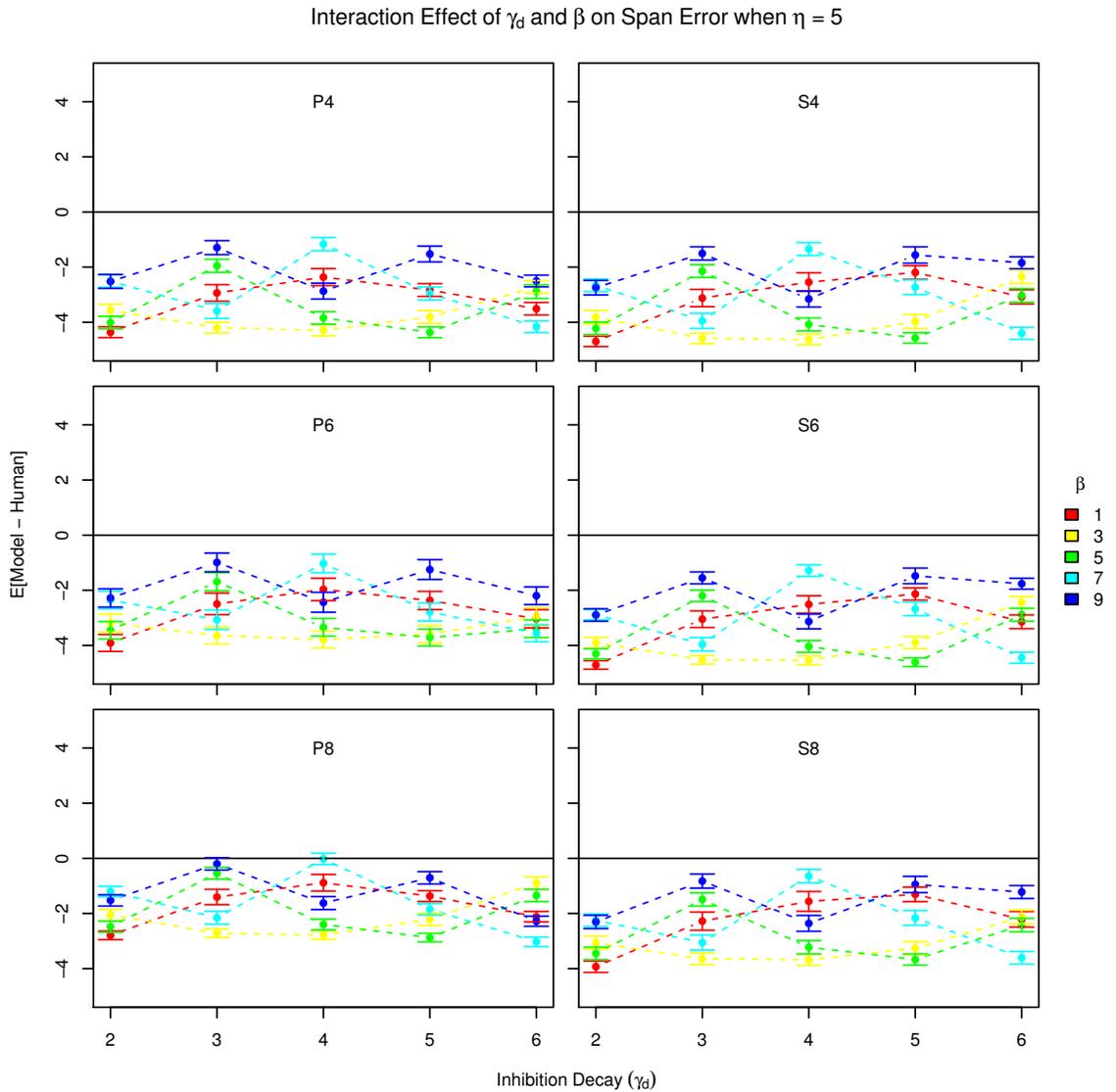


Figure 5.26: Difference between the model's and the humans' mean span scores as a function of the temporal inhibition decay parameter and the base-level constant across experimental conditions when  $\eta = 5$ . Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

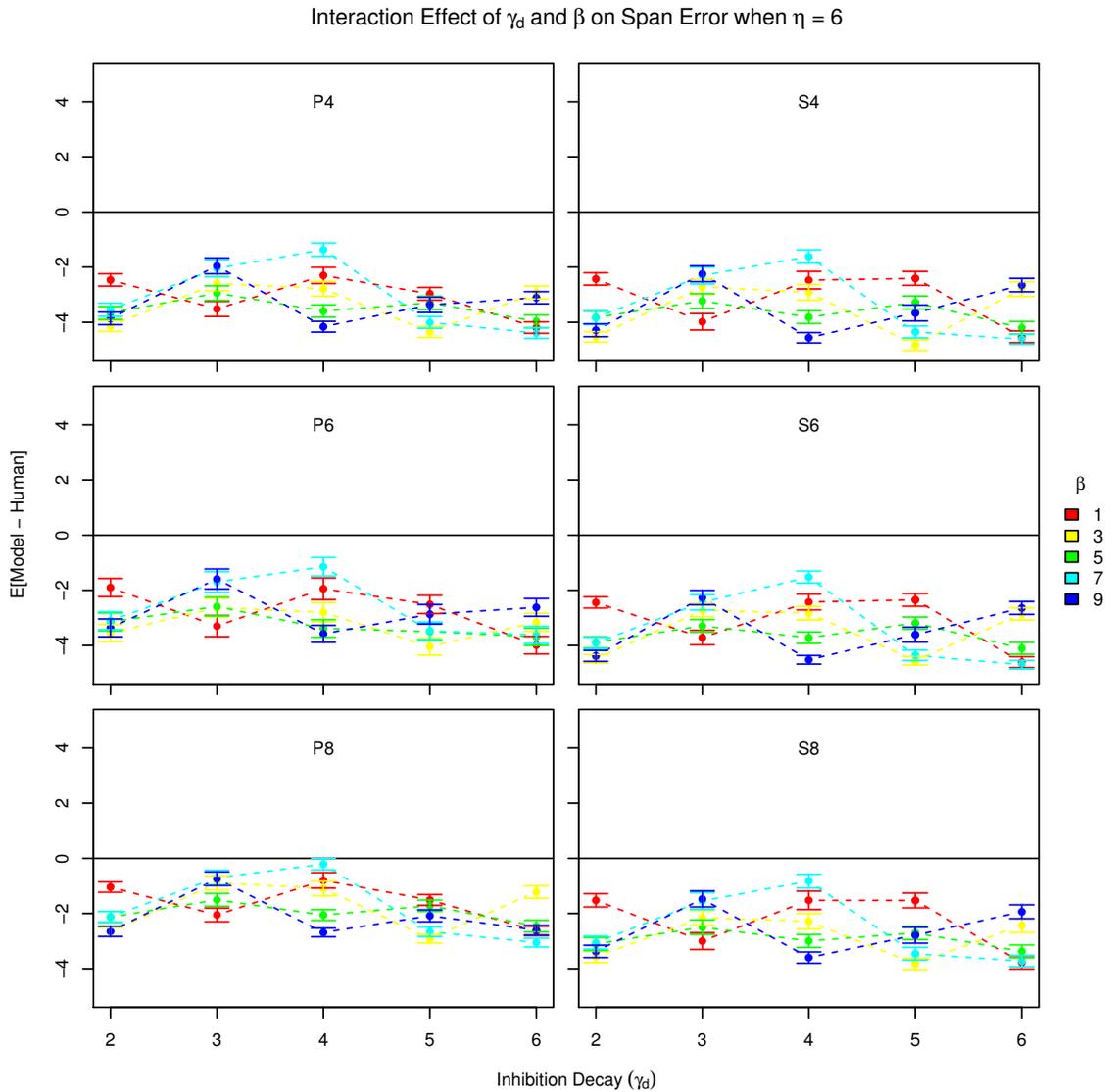


Figure 5.27: Difference between the model's and the humans' mean span scores as a function of the temporal inhibition decay parameter and the base-level constant across experimental conditions when  $\eta = 6$ . Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

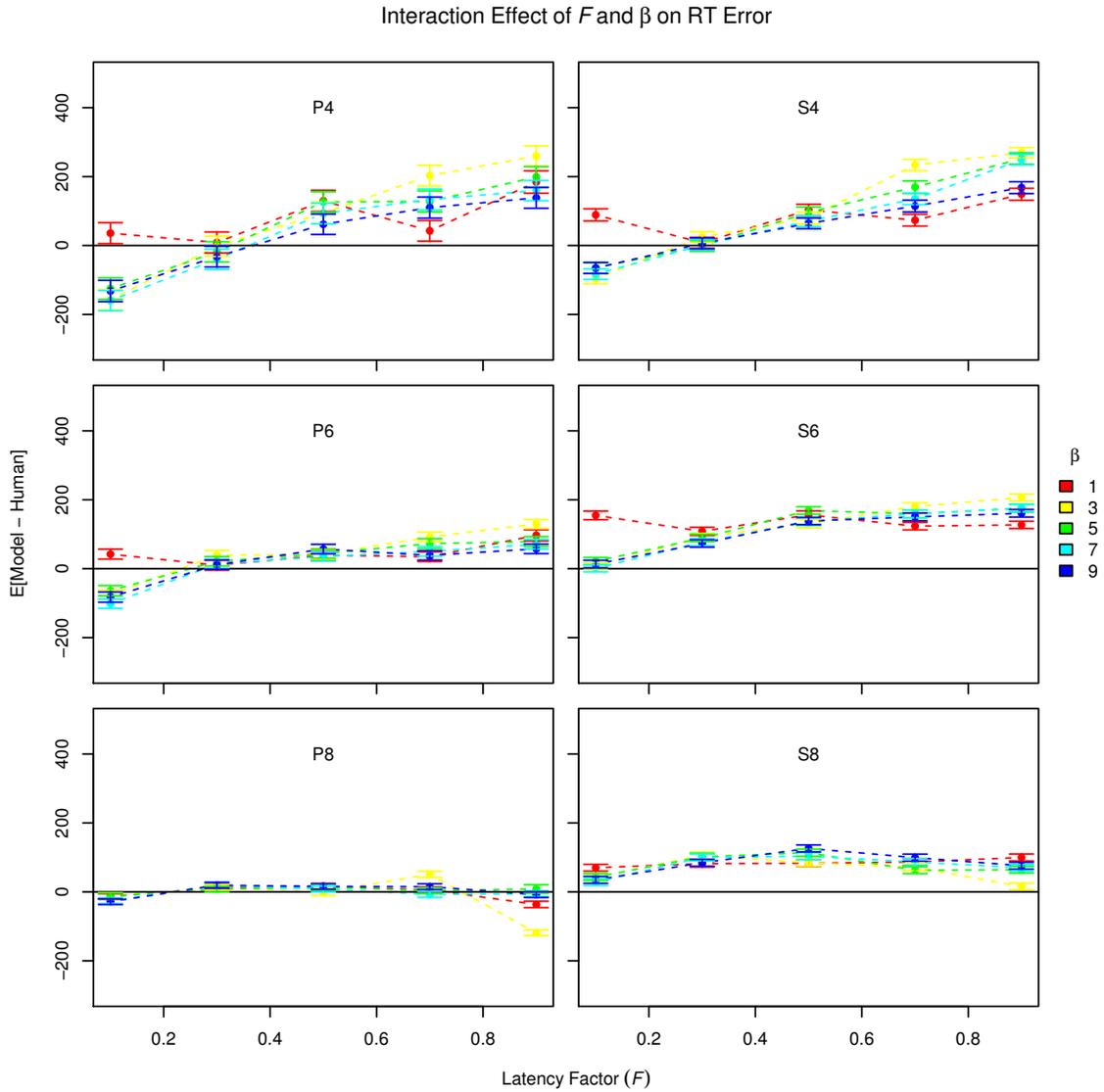


Figure 5.28: Difference between the model's and the humans' mean RT as a function of the latency factor parameter and the base-level constant across experimental conditions. Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

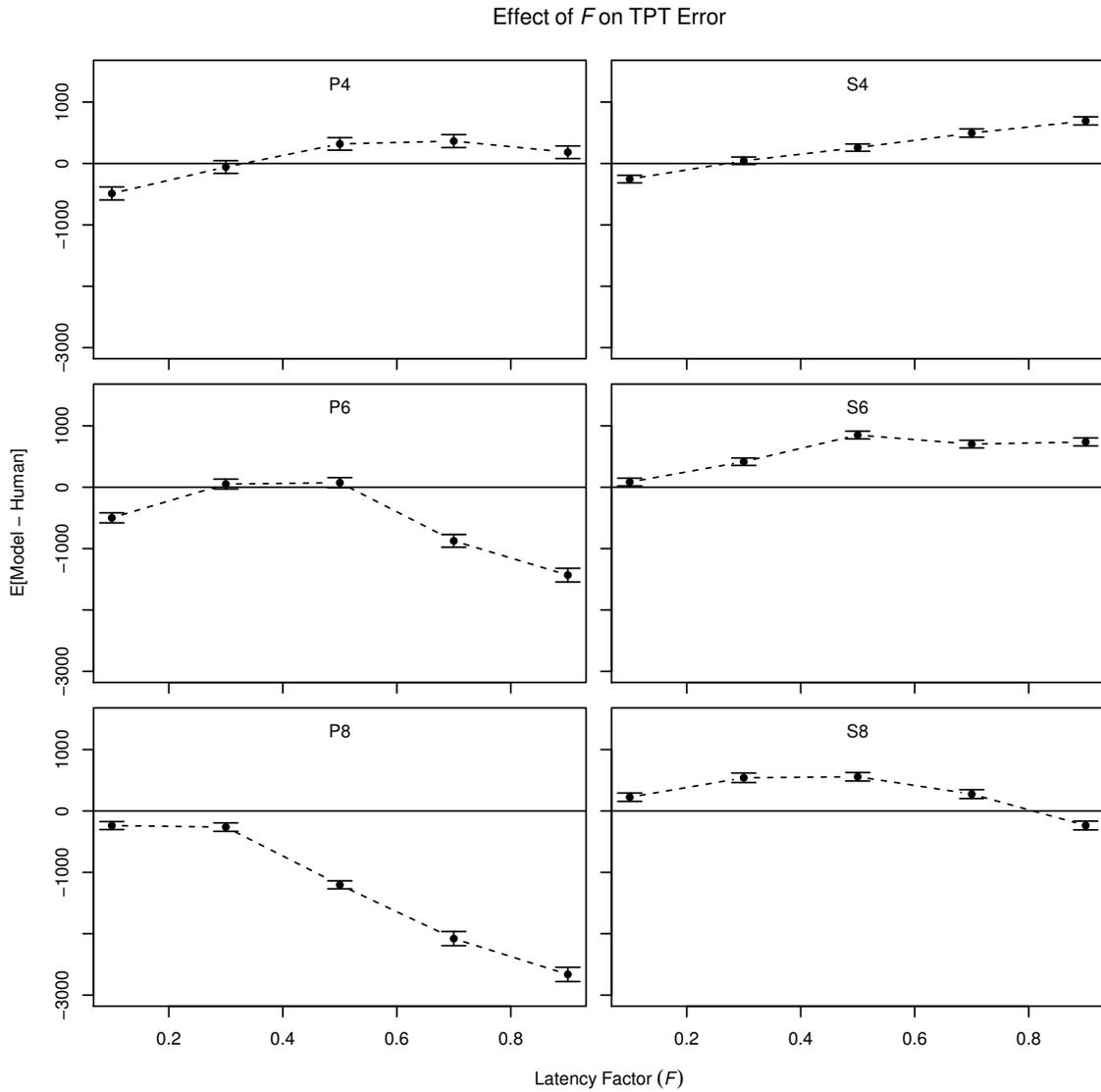


Figure 5.29: Difference between the model's and the humans' mean TPT as a function of the latency factor parameter across experimental conditions. Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

Effect of  $F$  on Accuracy Error

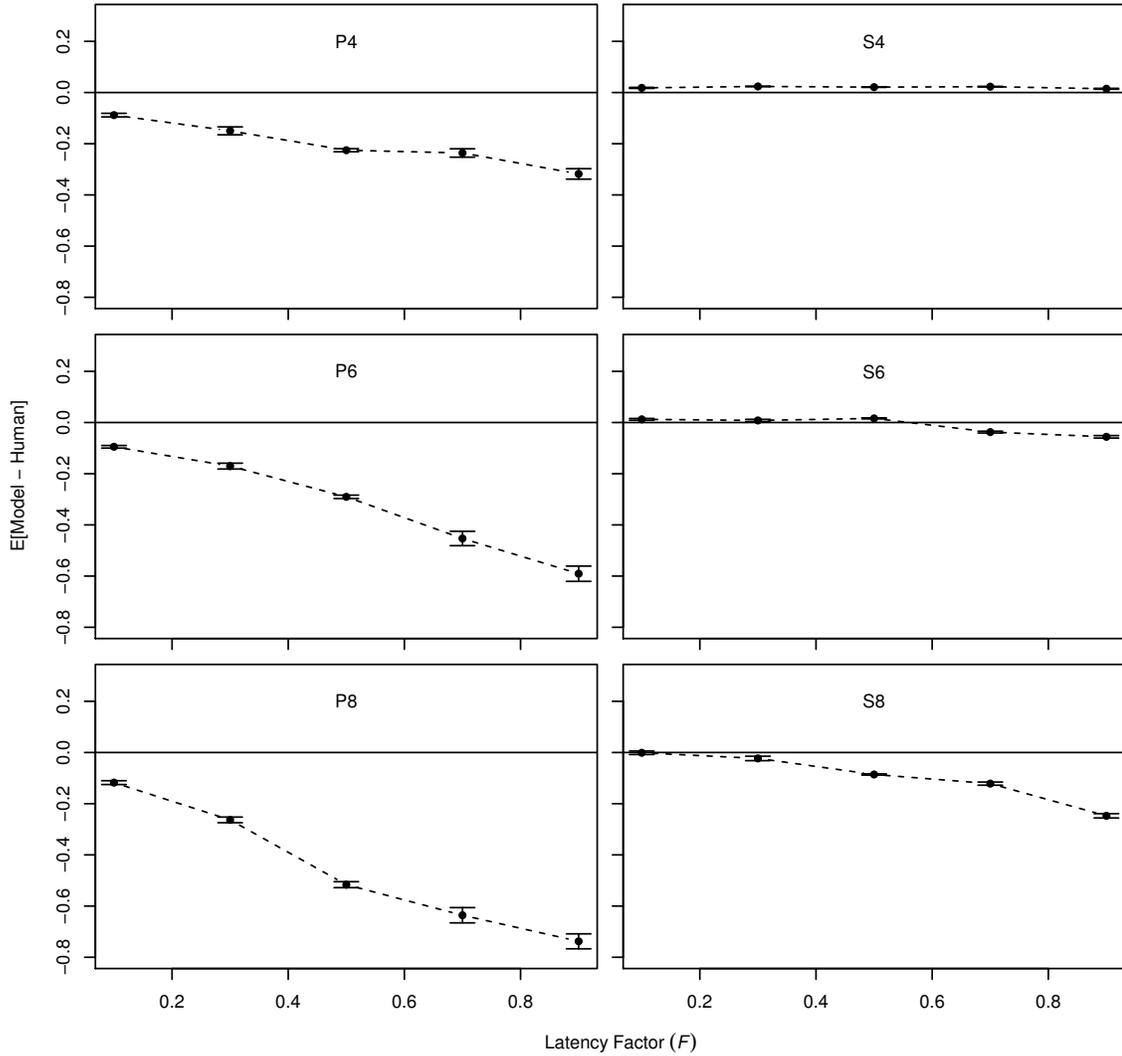


Figure 5.30: Difference between the model's and the humans' mean accuracy as a function of the latency factor parameter across experimental conditions. Error bars indicate the standard error of the difference in means. The horizontal line at 0 represents a perfect fit to the human data.

### 5.3.5 Slope

Overall, the error behind the mean differences in regression slopes was much greater than for the other measures. No parameters were found to be significant predictors of slope error.

### 5.3.6 RMSE

In addition to the error measures directly associated with each dependent variable, the root mean squared error used to determine  $\theta_{RMSE}$  (Equation 5.8) can also be regressed onto the free parameters. Rather than treating span, TPT, and slope errors separately, this error indicates how well the model fits the human span function proposed by Barrouillet et al. (2007, see also Barrouillet & Camos, 2015; Barrouillet, Portrat, & Camos, 2011). Similar to what was found with span error, there was a slight negative interaction of the inhibition decay parameter  $\gamma_d$  and the base-level constant  $\beta$ , suggesting that this measure primarily corresponds to the model's ability to fit the span data despite incorporating TPT and slope information,  $B = -0.08$ ,  $t(15618) = -2.72$ ,  $p = .023$  (Figure 5.31).

Interaction Effect of  $\gamma_d$  and  $\beta$  on RMSE

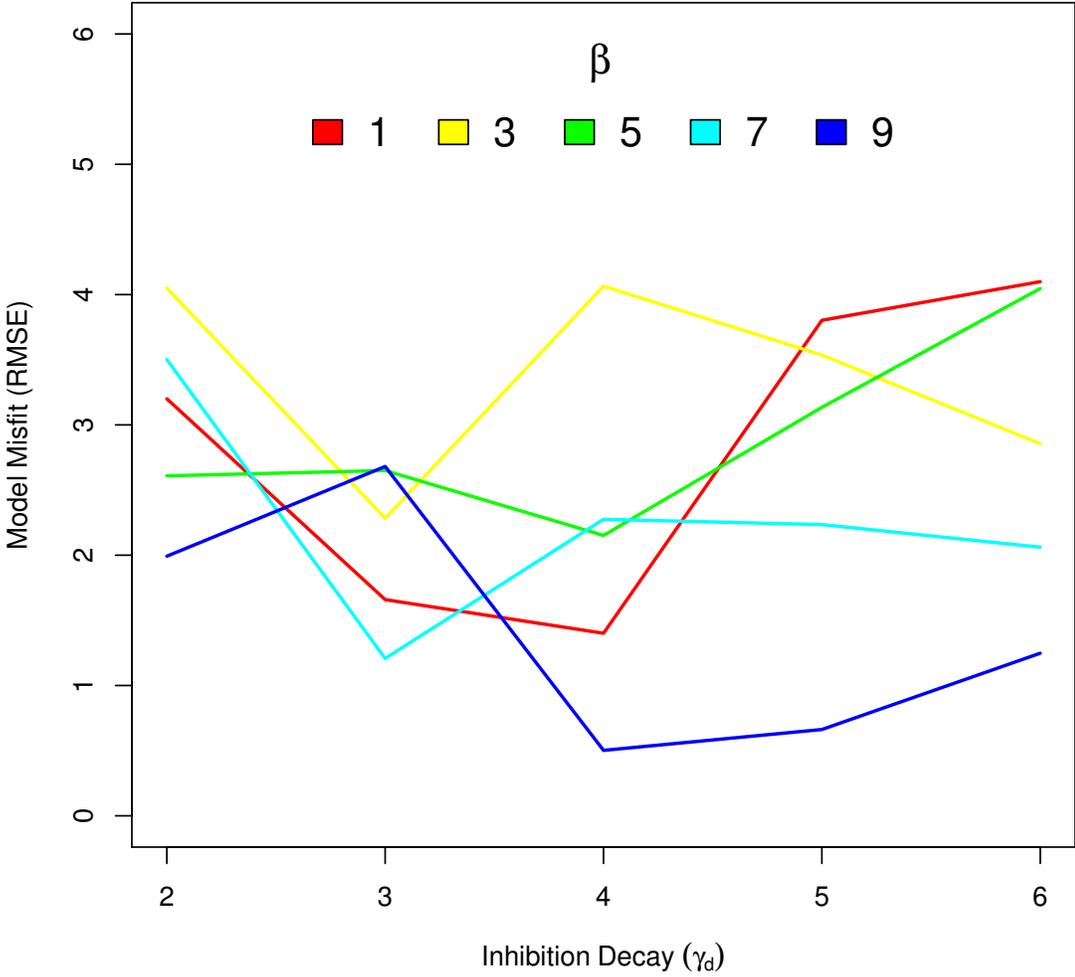


Figure 5.31: Interaction between the inhibition decay parameter and the base-level constant on the root mean squared error of the ability of the human regression function to predict the model's span from its cognitive load.

## 6 Discussion

I have successfully modeled the results of Barrouillet et al. (2007). The model emulates human performance in both recall and distractor tasks while expanding ACT-R to implement the first computational TBRS architecture. To accentuate this achievement, I next review the best fitting parameter sets designated by each approach to cumulative error assessment. These results are the product of a coarse grid search meant to aid in the understanding of general patterns of behavior over a generous range of the parameters hypothesized to affect the model’s performance. It is a proviso of discrete parameter optimization that the “best” parameter combination found is likely not the best possible parameterization of the model because the parameters are actually continuous, not discrete, variables. Likely the optimal parameter values lie somewhere between those evaluated. Nevertheless, examination of these sets as a whole reveals the peak limitations of the model, while comparisons between the sets themselves characterizes performance along the Pareto frontier (Figure 5.4), as fitting emphasis moves from matching individual areas of performance independently (i.e.  $\theta_0$ ) to capturing their holistic interaction (i.e.  $\theta_{RMSE}$ ).

### 6.1 Best Fitting Parameter Sets Based on Each Cumulative Metric

The parameter set  $\theta_0$ , which is the best fitting parameterization of the model when regression slope is not explicitly selected for, captures the effect of task-type well; however, with the exception of TPT, fails to produce much of an effect of distractor

pace. This is particularly clear in Figure 5.8 where the invariance of mean span with respect to CL results in relatively flat gradients in Figure 5.12.

The parameter set  $\theta_\varphi$ , which is the best fitting parameterization of the model when an acceptable regression slope is expressly mandated, replicates both experimental effects slightly better than  $\theta_0$ , albeit somewhat weakly in the case of RT. It is much better at fitting the regression of mean span on mean CL, almost perfectly in the parity condition despite consistently underperforming in terms of span. This parameterization has trouble fitting distractor response accuracy, noticeably underperforming in the parity condition. This is likely caused by  $\theta_\varphi$ 's relatively low reward parameter ( $R_\varphi = 5$ ) compared to the other “best” sets ( $R_0 = 13$ ;  $R_{RMSE} = 9$ ).

The parameter set  $\theta_{RMSE}$ , which is the parameterization of the model that minimizes the RMSE between model performance and Barrouillet et al.'s (2007) regression of mean span on mean CL, similarly reproduces the effects of task and pace. It overestimates mean RT and mean TPT in every condition, and, like  $\theta_0$  and  $\theta_\varphi$ , it tends to underestimate mean span. The direction of misfit with regard to span and TPT explains the diminished effect of CL (i.e. flatter regression lines with respect to the human data) in Figure 5.22. Although the magnitude of this effect for  $\theta_{RMSE}$  is less than the magnitude of the effect in the parity condition for  $\theta_\varphi$ ,  $\theta_{RMSE}$  does what the other “best” sets fail to do: it replicates the central result of the original experiment. The parallel regression lines in Figure 5.22 would be interpreted according to Barrouillet et al. (2007) as demonstrating little effect of task type once CL is controlled.

It is not clear why the model grows increasingly less variable than the human data with increased emphasis on fitting the regression of span on CL (i.e. from  $\theta_0$  to  $\theta_\varphi$  to  $\theta_{RMSE}$ ; Table 5.3). One reason may be that if there is little variance in terms of RT, and thereby TPT and CL, then according to Equation 1.3 there should

be less variance in terms of span. Variance in RT may be reduced in the model if it has settled into a single response strategy or pattern; therefore, inordinate procedural learning may be responsible. Alternatively, the simplified approach to modeling the response process, where the model cannot retrieve erroneous information and can only respond incorrectly by guessing, likely reduces variability in an unrealistic way. More work is required to determine the source of this discrepancy.

## 6.2 Parametric Effects on Model Behavior

While Figures 5.8–5.22 provide face validity, the more worthwhile contributions of the model come from its quantitative mapping of psychological constructs to behavior, and these parametric effects and specific recommendations for their instantiation in future work are discussed next.

I expected a three-way interaction of inhibition, temporal association, and baseline activation on mean span because of the formulation of the activation equation (Equation 3.1). Although this interaction just failed to reach statistical significance, the lower order interaction of inhibition and baseline activation did significantly contribute to misfit. Naively, the parameterization of the model that will produce the largest mean spans should have higher values for each of these three parameters because increased inhibition biases retrieval toward less recently retrieved items, increased temporal association biases retrieval toward items near the position of the most recently retrieved item, and baseline activation offsets the penalties of both. However, the optimal parameterization should balance the temporal inhibition and episodic similarity gradients in a way that promotes a steady refreshing cycle. Looking at Figure 5.23, in which the effect of temporal association is small, the interaction of inhibition and baseline activation resembles the performance of a primacy gradient model because, without the chaining-like influence of item similarity, list order is

preserved through the relative activation of each item, and greater inhibition produces a steeper primacy-like gradient that is able to resolve more items. As temporal association increases, the influence of episodic similarity becomes more pronounced, and the previously primacy gradient-like behavior of the inhibition and baseline activation interaction is disrupted because now the convoluted temporal inhibition and episodic similarity penalties must be coordinated. Optimal balance of these penalties should produce peak performance. Currently, such an equilibrium cannot be determined analytically because there are too many degrees of freedom; however, serial position errors (discussed later) could be used to determine episodic similarity a priori, which would in turn make the solution to this interaction more tractable.

The reduced effect of retrieval latency scaling on RT error in conditions with more inter-letter distractors (Figure 5.28) indicates some type of learning process. The model may be learning to respond quicker through production compilation, whereby fewer retrievals would occur and mean RT would become less dependent on retrieval latency in general, or it may be learning to avoid retrievals altogether and to guess more frequently. Evidence from TPT and accuracy errors (Figures 5.29 & 5.30, respectively) suggests that the latter may be closer to the truth. Not only does the number of learning trials increase with the number of distractors, but the length of time during which a response may be made also decreases. This is why mean TPT increases with mean RT as retrieval latency scaling increases in the four-distractor conditions, but it fails to keep up in the additional distractor conditions. Because TPT is the sum of the RTs to distractors following a given target, only a failure to respond can produce the disassociation between RT and TPT. If the model fails to respond on a given trial, no RT is available to count toward the TPT, and the model receives zero reward (as if it had made an incorrect response). This biases the model away from the long retrievals that led to the timeout and toward guessing. Learning to guess when retrieval latencies are long has multiple consequences. First, the effect

of retrieval latency scaling is diminished as the flat, minimum time associated with guessing becomes more prevalent. Second, if the model guesses incorrectly, it will then be forced to refrain from guessing on the next trial to ensure it learns the correct production sequence, as part of the strategy for modeling the speed-accuracy tradeoff (Peebles & Bothell, 2004). However, if retrievals always take too long because of the choice of scaling ( $F$ ), then the model is doomed to timeout again in a vicious cycle that tanks performance with respect to both TPT and accuracy. Furthermore, the decrease in accuracy below chance performance (Figure 5.30) is unlikely to have been caused purely by guessing; a substantial number of nonresponses are required.

No parameters were found to significantly affect slope error, but it is unlikely that this is because none of the free parameters contribute to the linear effect of CL. The interaction of inhibition and baseline activation on RMSE demonstrates this. The null finding can more likely be explained by the aggregation of individual results into three means for each regression of span on CL. As previously discussed, such a reduction in data introduces a large amount of uncertainty to the estimate of each regression coefficient (e.g. more potential lines can be drawn through three points than fifty). One solution to this problem would be to use individual, rather than group average, scores in the regression, but then repeated measures would be required in order to estimate the variance of the individual scores. Another strategy would be to design an experiment with more induced levels of CL so as to sample the full range of CL. Not only would this method yield more points to which to aggregate, but it could also reveal previously unknown characteristics of the span function. For example, TBRS predicts that span is a completely linear function of CL; however, there is no evidence to believe this prediction would hold true at extreme levels of CL. Intuitively, the mean span function seems far more likely to become exponential at the tails (i.e. as CL approaches 0 or 1) than to remain linear. Further, one could combine both techniques and estimate the effect of CL using hierarchical regression.

### 6.3 Recommendations for Future Parameterization

I include this section mainly for those proficient in ACT-R who might wish to implement the model presented here in their own work. Casual readers may ignore specific values as they do not provide additional comprehension beyond that presented in the previous section; however, their discussion does prime certain topics for follow-up investigation.

The reward parameter  $R$  did not substantially affect model fitness. Upon review, it appears that 9 may be the minimum value needed to yield sufficient accuracy (without regard to other parameters) while higher values do not produce increased performance.

Multiple methods (e.g., analysis of  $\mathcal{T}(\varphi)$ , multiple regression of span error and RMSE, and inspection of  $\theta$  sets) agree that larger values for the temporal inhibition decay parameter, approximately  $\gamma_d = 5$ , produce better mean spans and regression upon CL slopes.

An optimal range for the episodic selectivity parameter could not be established from this analysis, although the  $\theta$  sets suggest it may be nearer to  $\eta = 3$ . In future studies, this parameter may be determined more directly through the analysis of transposition errors (see comments regarding serial memory later in Discussion).

Overall, larger values for the base-level constant are favored because they allow greater temporal inhibition and episodic selectivity penalties to be applied. Values greater than 9 may need to be explored in the future.

Multiple regression results suggest that the latency factor parameter is positively related to RT error, TPT error, and accuracy error. Inspection of Figures 5.28–5.30 suggests the best fitting value of this parameter, which represents the maximum

retrieval time, to be approximately  $F = 0.3$  seconds. This value is in agreement with the  $\theta$  sets and the conclusions of the analysis of  $\mathcal{T}(\varphi)$ , namely that  $F = 0.3$  is associated with improved slope fits.

Because the latency exponent parameter, which scales the latency of a chunk's retrieval to its instantaneous activation, was not found to significantly affect any measure of model misfit, future applications of this model may reasonably fix this parameter at its default value ( $f = 1$ ). However, because the  $\theta$  sets and mixed evidence from the analysis of  $\mathcal{T}(\varphi)$  seem to favor lower values, particularly  $f = 0.1$ , it may be worth exploring an interesting alternative specification of the model. Through various memory-loading paradigms, Vergauwe, Camos, and Barrouillet (2014) and Vergauwe and Cowan (2014) have demonstrated that processing time increases an average of 50 ms or 35-40 ms, respectively, per item in short-term memory. Vergauwe and Cowan (2014) go so far as to suggest that this rate corresponds with 40 Hz gamma oscillations observed in cortical and hippocampal tissue. Such a constant refreshing rate could be tested in the current model by modifying it so that refreshing retrievals are forced to take a fixed amount of time, analogous to  $f = 0$ . Given Vergauwe et al.'s (2014) results, 50 ms, the default action time in ACT-R (the amount of time it takes for a production to fire), is probably a good place to begin such an investigation. Furthermore, fixing the latency of refreshing retrievals at a constant interval greatly simplifies the activation equation (Equation 3.1) because the retrieval times of each chunk  $t_j$  become determinable instead of chaotic. The activation of each item in a list could potentially be computed analytically, and small-scale simulations could be used to set the  $\gamma_d$ ,  $\eta$ , and  $\beta$  parameters of the more complex ACT-R model.

## 6.4 Shortcomings and Assumptions of the Model

### 6.4.1 Reward and Utility Learning

Two misconceptions about reward in ACT-R impeded my progress in model development. Utility, like activation, in ACT-R is measured on an interval scale; the difference between two utilities matters, but zero utility does not imply a lack of utility. I originally tried to apply a ratio scale to utility, treating productions with positive utility as likely to lead to positive outcomes, productions with negative utility as likely to lead to negative outcomes, and zero utility to describe neutral productions. I had one nonzero reward parameter that was awarded for correct responses, and its opposite was awarded for incorrect responses. In the current experiment where there are only two alternatives, a correct response and an incorrect response, this mistake by itself would effectively only cause the reward parameter to scale at twice its current rate (i.e. when incorrect responses receive zero reward), although the interpretation of the parameter would be somewhat different.

The second early mistake I made stemmed from the first. Because the reward a production receives is discounted by the amount of time between when it fires and when the reward is triggered, I realized that a correct response needed to be awarded with at least its latency in order for every production that led to it to receive nonnegative reward. However, in the present experiment, some conditions allow more time than others in which to respond, which means that in order to preserve a neutral zero point (i.e. ratio scaling), more reward would be available to some conditions than in others. In order to equate the reward received across conditions, I set the reward parameter to the RT on each trial. This initially made sense because longer RTs indicated extended effort, which should result in greater reward/penalty for correct/incorrect responses, but it was disastrous because it caused the model to learn

productions that led to slower, not faster responses. Additionally, although compiled productions would still learn to overtake their parents, utility learning (Equation 3.4) would then adjust their utilities until they were equal, causing the slower response path and the faster response path to be chosen with equal probability.

## 6.4.2 Metacognition and Feedback

Barrouillet et al. (2007) reported a speed-accuracy tradeoff in the human responses to distractor items. In order to model this, I sought a method used in other ACT-R models to produce such an effect. Peebles and Bothell's (2004) technique, which I implemented in my model, uses the ACT-R procedural learning system to bias the model toward either a slow but accurate strategy or a quick but less accurate strategy. This method requires some form of feedback to guide utility learning, but Barrouillet et al. only provided their participants with feedback during the training phase of the study. Without trial-by-trial data, which would have revealed any learning effects over trials, I was forced to assume whether procedural learning, necessary for Peebles and Bothell's technique, took place only during training or over the course of the entire experiment. Choosing the latter, I needed to give the model a way to generate its own feedback, which I achieved through "metacognition".

Although a metacognitive strategy solved my missing feedback problem, it is clear from post-analysis that metacognition introduced its own issues. First, it may remove any incentive for responding prematurely because the model must always make every retrieval necessary for determining the correct response so that it can reward itself, which undermines the faster strategy in Peebles and Bothell's (2004) technique. Secondly, the retrievals in service of metacognition, potentially taking place after the model initiates a response, would not be represented in the RT-based proxy of cognitive load used in my and Barrouillet et al.'s (2007) analyses, despite certainly

contributing to the cognitive load experienced by the model. This could pose a serious problem when interpreting my results. Underestimating cognitive load may partly explain the model’s overall underperformance in terms of mean span. It may also contribute to response lapses.

Not only would the removal of metacognition produce better agreement between proxies of cognitive load and actual cognitive load, it would restore a real incentive for guessing because faster responses would be rewarded sooner. Crucially, this approach would entirely depend on whether the training phase was sufficiently long enough for procedural learning to produce the speed-accuracy tradeoff across conditions observed in the human data. The trial-by-trial responses in a new experiment should be sufficient for resolving this decision, but it could be explored using the current model and data. A new experiment could be designed to test for metacognition by manipulating whether feedback is provided. If the human participants employ such a metacognitive strategy when experimenter-provided feedback is unavailable, then lower mean spans should be observed in the condition without feedback because the additional retrievals will induce relatively more cognitive load than the condition with experimenter-provided feedback.

### **6.4.3 Declarative Long-Term Memory**

In early versions of the model, before I modified the number of references to LTM chunks to stabilize their base-level activation, the model treated the creation time of these chunks as the moment the model was loaded (i.e. the very beginning of the experiment), causing these chunks to have a base-level activation that was smoothly decaying as opposed to stable. Failing to stabilize the activation of these chunks prior to the onset of the experiment caused unwanted model behavior. Because activation was no longer a linear function of the base-level constant  $\beta$ , the tradeoff between  $\beta$

and the retrieval latency scaling parameters made fitting the model tricky. Lower values of  $\beta$  performed better because longer retrievals allowed for greater disparity in RTs once the retrievals were obviated by production compilation, thus producing a more pronounced speed-accuracy tradeoff across conditions; however, if  $\beta$  decreased below some threshold, then the model would fail catastrophically by failing to retrieve simple LTM chunks (e.g., the asterisk, numbers, or letters), akin to forgetting how to read.

This anecdote highlights a prominent feature inherited by the model: ACT-R has a unitary system of declarative memory. In contrast to other models of human memory (notoriously Atkinson & Shiffrin, 1968, but more recently Baddeley, 2012), which have separate systems for STM and LTM, my model blurs such a distinction. The degree to which a memory is more long-term than short-term is related to the frequency of its use in the past. In my model, what would be considered short-term memories in the divided system view are simply memories that have not been retrieved often enough to survive the decay experienced over an interval of interest. Because the declarative memory module in ACT-R unmistakably instantiates LTM, some (such as Anderson et al., 1996), working under the assumption of separate LTM and STM systems, have interpreted the module buffers as representative of STM. My model rejects using module buffers for storage and instead advocates the stronger view of TBRS that buffers represent bottlenecks on the flow of information.

#### **6.4.4 Serial Memory**

Complex span tasks are designed to measure WMC, and the evidence in the literature suggests that they do this by tapping into the ability to selectively control attention (Chow & Conway, 2015; Conway et al., 2005). However, many assumptions about list and order representation must be made in order to construct a model of

WM, which theoretically is a separate construct. Doubtless, any model will have to make assumptions about representation, but the dependence of complex span performance on serial memory should lead one to question the degree to which a WM model's success can be attributed to the validity with which it captures the workings of any real WM mechanism or to the quality of its assumptions about list and order representation. One method commonly used in the literature for avoiding this problem when measuring humans' WM abilities in order to predict other constructs (e.g. general intelligence) is to administer multiple WM tasks, such as binding and updating tasks in addition to complex span tasks, and then use latent variable analysis to gauge the common variance. Similarly, the robustness of WM models should be assessed by determining their convergent validity across multiple tasks. My model of TBRS is exceptionally suited for undertaking such a challenge because it was developed using the ACT-R framework, and many studies have demonstrated that ACT-R is capable of modeling a variety of different tasks. This is one future direction in which to take my model that would be particularly interesting because TBRS was developed around complex span tasks and to my knowledge has seen only limited application to alternative classes of tasks.

Another way to address the validity of my model's serial memory assumptions is to evaluate serial position errors. Unfortunately, these data are not reported in Barrouillet et al. (2007). Therefore, I did not make capturing such error patterns a priority of my model. That being said, serial position errors are heavily utilized throughout the serial memory literature for inferring list structure. A new dataset that includes serial position errors from a study employing similar manipulations of cognitive load would go a long way in further constraining my model. For example, the episodic selectivity parameter  $\eta$  effectively governs the rate at which an item at position  $x + 1$  is retrieved over an item at position  $x + 2$ , etc. The rate at which these transposition errors are found in human data could be used to a priori fix  $\eta$ .

Furthermore, such a dataset would allow me to make a principled investigation into Portrat and Lemaire’s (2014) claim that TBRS models with a focus of attention of one item cannot produce recency effects.

Even without serial position data, there are some types of errors that my model definitively cannot produce in its current form. Unless its activation somehow drops below threshold, which is unlikely because it has the most opportunities to be refreshed out of all the list items, the first item recalled will always be the first target in the list because the first retrieval in the recall subroutine (Figure 3.7) specifically requests a chunk explicitly marked as first. The model is also unable to make intrusion errors, where an item from a previous list is erroneously recalled in the current list, because I did not define a similarity function for list contexts. The partial-matching function (Equation 3.2) could be expanded in future work to also include list contexts, but I avoided this in the current model because it would require additional free parameters to fit data I do not even have. Yet simply expanding partial-matching in this manner would not be able to account for the finding that intrusions from prior lists tend to maintain their relative position (Burgess & Hitch, 2006). This effect is commonly taken as evidence for positional coding because encoding items with their absolute time of encoding, which the model currently does, would not produce the same cross-position similarities. However, one could imagine using a hierarchical memory structure where the highest item is encoded with an absolute time, and items at each successively lower level are recursively encoded with the relative time difference between their encoding and the sum of their parents’ encodings (e.g., the list is encoded with context  $x$ ; a group within the list, created at time  $t$ , is encoded with context  $y = t - x$ ; and an item within that group, created at time  $u$ , is encoded with context  $z = u - (y + x)$ ; etc.). Such a coding scheme would preserve within-list relative similarity while continuing to use temporal, rather than ordinal, contextual information. Interestingly, this algorithm is compatible with a theory that items are encoded in the brain with high-frequency

neural oscillators and that the structure in which they are nested (e.g. the list context) is encoded with neurons oscillating at lower, harmonic frequencies (Lisman & Idiart, 1995; Vergauwe & Cowan, 2014). It is also congruent with Farrell’s (2012) theory that temporally similar items are spontaneously grouped into clusters.

Lastly, with regard to serial memory, there seems to be an implicit assumption in the literature that requesting participants to “recall the items in the correct order” is equivalent to asking them to “recall the items in the correct positions”, but the latter request is actually a stronger case of the former. We know that different emphases in instructions can influence how participants approach a task (e.g. speed/accuracy bias). It may be the case that emphasizing item order may bias participants toward using a chaining representation, and emphasizing item position may bias participants toward using a positional coding representation. Failing to acknowledge this subtle difference in task instructions may cause undue heterogeneity of variance from averaging over different strategies, which would in turn make further theory development more difficult. Computational modeling provides a means for testing this hypothesis in future studies that manipulate subject instructions in such a way. Otherwise identical models could be created that use chaining or positional coding, and the hypothesis would be supported if they differ in their ability to fit each form of instructions.

#### **6.4.5 Other Methodological Restrictions**

The availability of only those group-level data reported in Barrouillet et al. (2007) limited this project. Individual trial responses, necessary for determining serial position errors, response lapse rates, and other alternative indices of performance that could further constrain parameters, are no longer available (P. Barrouillet, personal communication, February 4, 2015). That being said, the published data available was sufficient for developing the model and exploring its behavior. It may even be

adequate for investigating some proposed extensions of the model. Undoubtedly, a new study must be conducted in order to robustly scrutinize the model, but in its current state the model is able to suggest how such a study should be designed and to predict how human participants will perform in it. Such a study should follow the advice of Conway et al. (2005) and use partial-credit scoring because it is slightly more reliable than all-or-nothing scoring, which Barrouillet et al. (2007) used. Because I did not have access to their raw data, I scored my model as Barrouillet et al. did in order to be consistent with their results, but a more reliable metric would reduce variability in model performance. Partial-credit scoring would particularly benefit a study of individual differences by yielding more stable individual scores.

## **6.5 Future Versions of the Model**

### **6.5.1 Refreshing Strategies**

In addition to the alternative versions of the model discussed above, a variety of other implementations may be interesting to explore. The current refreshing strategy of the model is appealing because it is simple and epiphenomenal. The order in which items are refreshed is not explicitly specified, rather the next item to be refreshed is selected based on its instantaneous activation, which is dependent on inhibition, contextual association, and refreshing history. That being said, a number of different explicit refreshing strategies could be experimented with. For example, Oberauer and Lewandowsky (2011) describe trying out multiple refreshing strategies (i.e. with which item to restart the refreshing loop after interruption). They found that restarting the refreshing loop with the first item in the list worked best, followed closely by continuing the loop with the last item refreshed. They comment that using the first item in the list resulted in slightly more monotonic span/CL gradients than using the last item refreshed. Anderson et al. (1998) used a strategy that randomly chose

between rehearsing the current item displayed on screen and rehearsing the entire list in order. My own simulations in R (Figure 6.1; R Development Core Team, 2008) suggest that this strategy, when used in conjunction with ACT-R's base-level learning mechanism, may be sufficient for producing the U-shaped serial position curves Anderson et al. (1998) obtained. This is significant because Portrat and Lemaire (2014) have challenged key assumptions of TBRS based on the difficulty of TBRS\* to match human serial position curves. The results of Anderson et al. (1998) suggest that refreshing strategy, rather than an expanded focus of attention, may be sufficient for producing recency and other serial position effects, leaving the original TBRS specification intact.

### Simulation of Anderson et al. (1998)

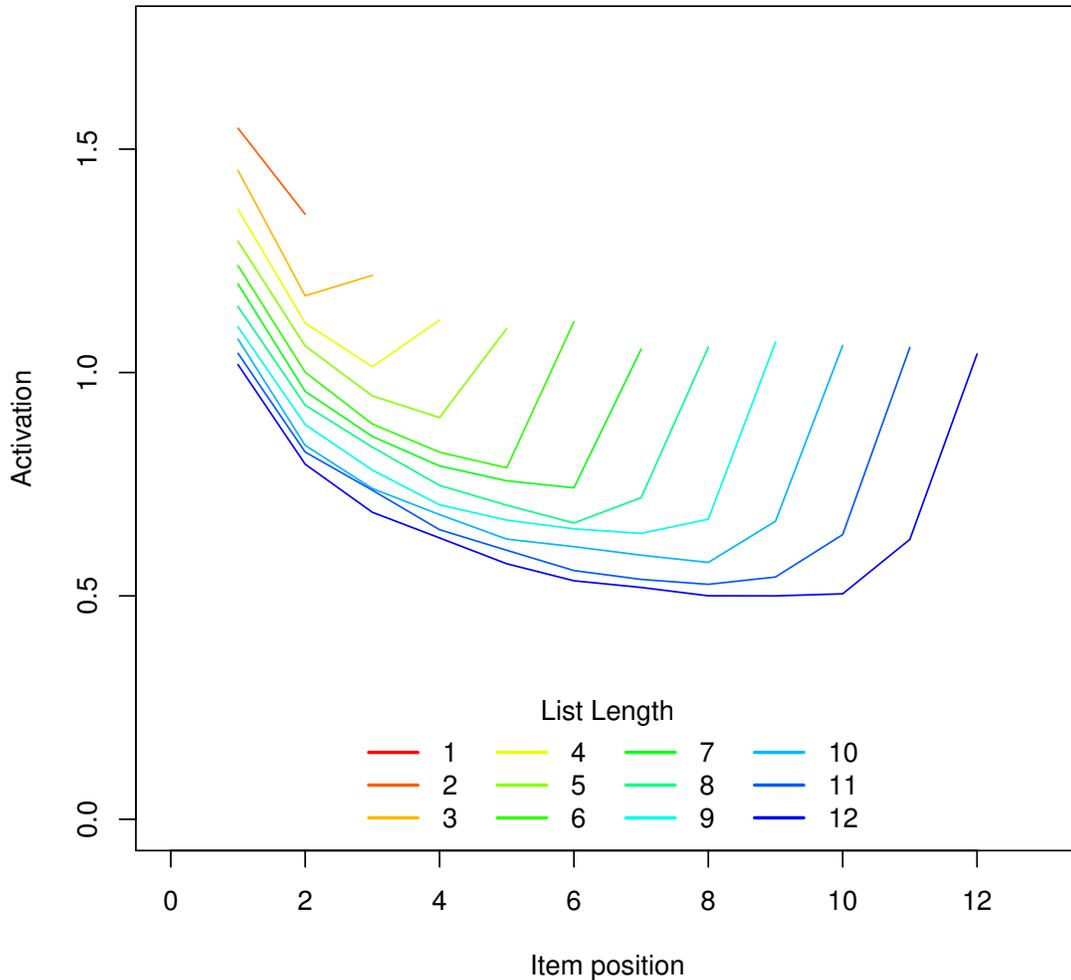


Figure 6.1: Simplified simulation of Anderson et al.'s (1998) model of immediate serial recall constructed in R. Average activation of each item in lists of size 1-12 items after 1000 runs. For each item in the list, the model has one second to make four rehearsals. For each rehearsal, the model is equally likely to choose one of two strategies. The first strategy is to rehearse the item currently displayed on screen. The second strategy is to rehearse the item in the position after the last item rehearsed. At all times, the model retains a pointer to the last item rehearsed, excluding rehearsals initiated by the first strategy. Characteristic U-shaped serial position gradients are caused by earlier items benefiting most from the latter strategy, while later items benefit from the relative recency with which they were rehearsed by the former strategy. Items in the middle of the list suffer most because they are less likely than the early items to be chosen by the latter strategy and because the former strategy will have rehearsed them less recently than the later items.

### 6.5.2 Individual Differences and the Fan Mechanism

Another potential direction for further exploration of the model is to reconsider the role of the fan mechanism. I intentionally avoided this mechanism because it divides spreading activation equally amongst the items in the list, and TBRS strongly opposes such capacity sharing. That being said, the fan effect is an early premise of ACT-R (Anderson, 1976, 2007; Pirolli & Anderson, 1985), and it would be worthwhile to add this component back into the model, particularly because one parameter of spreading activation,  $W_j$ , has been proposed as the source of individual differences in WMC (Lovett, Reder, & Lebiere, 1999). Considering Unsworth and Engle's (2006) suggestion that the ability to efficiently use temporal-contextual information is responsible for individual differences in WMC and the similarity between spreading activation (Equation 6.1) and a rearranged form (Equation 6.2) of partial-matching from my model (Equation 3.2), one might suspect the episodic selectivity parameter  $\eta$  to account for individual differences in WMC.

$$S_i = \sum_j W_j (\varsigma - \ln m) \quad (6.1)$$

$$P_i = \eta \left( \frac{\xi}{\eta} - \ln \left[ 1 + \frac{|\epsilon_i - \epsilon_{\text{requested}}|}{\omega} \right] \right) \quad (6.2)$$

Let  $\varsigma, \xi = 0$  and substitute the quantity  $x$  for  $\left[ 1 + \frac{|\epsilon_i - \epsilon_{\text{requested}}|}{\omega} \right]$ , and one can see that, when assuming one source of activation (e.g. list context),

$$S_i = W (0 - \ln m) \quad (6.3)$$

is analogous to

$$P_i = \eta (0 - \ln x) \quad (6.4)$$

and that  $\eta$  is functionally equivalent to  $W_j$ . Further studies are required, but a comprehensive interpretation of my model suggests that cognitive load explains task-driven, between-subjects constraints on WMC, while  $\eta$  explains within-subjects constraints on individual WMC.

In models of list memory that utilize continuous refreshing or rehearsal techniques, the boost in activation from the fan mechanism at the time of recall is negligible compared to that from base-level learning. It has the greatest influence upon retrieval for very short lists and during the maintenance of the first few items in longer lists, but only when very few refreshing retrievals are allowed. These effects, in addition to TBRS's emphasis on temporal constraints, suggest that the fan mechanism is not necessary to produce the list length effect. However, experiments designed to elicit the fan effect without invoking list length (e.g. sentence recognition paradigms; Anderson & Reder, 1999) indicate that the fan effect is likely a component of associative memory. The present model suggests a role for the fan mechanism in serial memory. The model can run into a problem where early items, particularly the first two items, experience a lot of temporal inhibition because they are always likely to have been recently retrieved. This excessive penalty causes refreshing retrievals to fail; thereby, forcing the model to wait for the full duration of the latency factor ( $F$ ) before another retrieval can be attempted. Such failures drastically reduce the number of refreshing retrievals that can be attempted during a period of maintenance. Conversely, late items in the list, which are refreshed less frequently because there are more items to refresh before their next turn, experience very little temporal inhibition by comparison. Attempts to minimize the excessive inhibition of early items by simply lowering the inhibition decay parameter  $\gamma_d$  will further reduce the effect of inhibition at longer list lengths, introducing extra transposition errors and reducing observed span because the early items have now built up too much activation. The solution to these problems is a bonus to activation that decreases with list length (or a penalty that increases with

list length, depending on implementation), which is exactly what the fan mechanism provides. The parameterization that approximately equates the rates of change in fan bonus and inhibition penalty over list length will ensure that the influence of temporal inhibition upon maintenance is constant across list length.

Interestingly, a model that implements such a system would predict that individual differences in WMC arise from the ability to appropriately balance two factors that have independently been proposed in the literature to explain such differences: the previously mentioned strength of association weight  $W_j$ , which is meant to represent the amount of activation that can be divided amongst associated chunks (Lovett et al., 1999), and efficient use of temporal-contextual information, represented in the current model as temporal inhibition and episodic association. These parameters could be empirically disassociated by independently manipulating target list length and inter-target pacing. Individual differences might be caused by inappropriate weighting of episodic association ( $\eta$ ) or strength of temporal inhibition ( $\gamma_d$ ), insufficient sensitivity to inhibition or reduced plasticity in modulation of compensating attentional strength ( $W_j$ ; or simply ceiling effects in the attentional capacity available). Future work in modeling serial recall paradigms will not only be able to validate this compensation hypothesis but will also be able to identify which mechanism(s) (i.e. parameter) covaries with observed individual differences by systematically manipulating each parameter.

### 6.5.3 Temporal Decay Versus Representation-Based Interference

The assumption of temporal decay is fundamental to TBRS, and careful experimental manipulations have been used to provide evidence for it over representation-based interference (Barrouillet et al., 2012; Barrouillet, Portrat, Vergauwe, et al., 2011; Lewandowsky & Oberauer, 2009). While ACT-R also assumes temporal decay in

base-level learning (Equation 1.6), this choice was not originally a strong assumption of the theory. Indeed, early works suggest that the critical variable for decay could be the number of intervening events (Anderson et al., 1998: footnote 3). It would require modifying the declarative module, but my ACT-R model of TBRS could be used to directly compare these two competing hypotheses by implementing each into otherwise identical models and then comparing their respective fits to human data.

#### **6.5.4 Non-Attentional Mechanisms of Maintenance**

Recent research has investigated the possibility that the TBRS conception of WM is supported by the coordination of separate attentional refreshing and articulatory rehearsal loops (Barrouillet & Camos, 2015; Camos & Barrouillet, 2014; Vergauwe et al., 2010). The version of the model presented here does not use articulatory rehearsal; all maintenance is performed using attentional refreshing. However, because participants in Barrouillet et al. (2007) were not under articulatory suppression, it is reasonable to believe that participants did engage in rehearsal. I developed the code necessary to implement three WM strategies: attentional refreshing alone, articulatory rehearsal alone, and coordinated refreshing and rehearsal; despite only evaluating the model with attentional refreshing alone. I look forward to assessing the other two maintenance strategies in future work.

#### **6.5.5 Adaptive Maintenance**

The model has identified one place where the TBRS theory can be strengthened. Whereas TBRS expanded task-switching models to allow refreshing to take place during breaks in processing within tasks as well as between tasks, my efforts suggest that this is not common, at least for simple tasks like Barrouillet et al. (2007). Modeling each step of the distractor task reveals that retrieval takes longer than any other operation in the processing chain. Thus, as soon as one retrieval is completed, the

system needs to make another processing-related retrieval, and there is no free time with which to perform maintenance. I speculate that in more complicated processing episodes, where the prolonged obstruction of the central bottleneck makes it likely that all to-be-remembered items are lost to decay, humans insert endogenous breaks in processing so as to perform maintenance. Because the intermediate processing-related information needs be retained until after this burst of maintenance, it too will need to be refreshed. The dynamics of this interplay, and people's ability to adaptively strategize their refreshing process, are worthy of further study.

## 6.6 Summary of Predictions

This work provides additional support for the predictions of the TBRS model, that variations in WM span can be explained in terms of variations in cognitive load. Going beyond Oberauer and Lewandowsky (2011), I have demonstrated that when situated within a larger cognitive architecture, ACT-R, a computational model of TBRS is also capable of predicting accuracy and RT in the distracting task.

The current model indicates that greater WMC is caused by increased self-inhibition and better temporal acuity. Inhibition is a key component of cognitive control (Juvina & Taatgen, 2009), which has itself been previously linked to WM (Conway et al., 2005). Likewise, the ability to more effectively use temporal context has also been suggested to be associated with WM span (Unsworth & Engle, 2006). Future work may reveal individual differences in WMC that reflect variation in the parameters controlling these mechanisms.

The model suggests that additional reward pressure in the distractor task would not improve performance in terms of RT or span. For this experiment at least, it is likely that motivation is already at ceiling for such simple tasks (i.e. parity and location judgments).

By fixing the model’s retrieval threshold at zero, I was able to coerce ACT-R’s retrieval latency factor parameter to represent maximum retrieval time. Parameter fitting suggested this limit to be approximately 300 ms. Electrophysiological studies using event-related potentials may be able to corroborate this prediction.

The model makes additional predictions that were not directly tested here. While the model does demonstrate a linear relationship between cognitive load and WM span, I only evaluated the six levels of cognitive load in Barrouillet et al. (2007). Further simulation of other distractor paces will make stronger predictions about the form of this function, including at extreme values of cognitive load.

The interplay of decay and inhibition in the model hints at a possible optimal refreshing pace. On one hand, if refreshing is too frequent, items may have not recovered from inhibition by the time of their next attempted retrieval. On the other hand, if refreshing is too slow, then gratuitous decay may occur resulting in inefficient maintenance. The optimal pace balances these forces such that the most information possible to retain is preserved. Assuming such an ideal rate exists, then it would be advantageous for the cognitive system to have some means of regulating it. Future research is needed to identify this proposed mechanism.

The model’s partial matching function (Equation 3.2) compares the temporal context of each target item when selecting the next item to refresh. Just as two items that were encoded closely in time are likely to be recalled in succession, the overall proximity of each item determines the main transposition error rate. Because items are essentially encoded with the time of their presentation, this predicts that the order of items is more likely to be confused the more quickly they are presented. Irregular target presentation rates could be used to test for selective induction of serial position errors. This prediction only holds in the current model if parameters are static – adaptive modulation of inhibition or temporal association scaling ( $\omega$ ) precludes this

assessment. Pure positional coding models (i.e. where each item is encoded with its ordinal position) would also predict no increase in transposition errors.

I identified various amendments to be made to the model and considered their impact. One such area of improvement is the memory structure used to represent the target list. Currently the model uses a simple feature to mark items as belonging to the same list but does not have the means for relating different lists. While this makes it impossible for the current model to make intrusion errors, the groundwork for developing this capability in the future lies in the way it implements same-list transposition errors now. I have outlined how the existing episodic similarity function could be generalized to higher-order structures (e.g., separate lists, clusters within those lists, or even the experiment itself). Just as the episodic selectivity parameter ( $\eta$ ) controls the migration of items within a list, additional similarity strength parameters would constrain errors between elements of the episodic network.

I outlined issues with the model's current method of procedural learning with respect to the speed-accuracy tradeoff observed in human data. It is not clear from the data available whether learning the tradeoff continues after training, but the model predicts that for such learning to occur then some form of feedback, whether self-generated or experimenter-provided, is required. If subjects employ the same metacognitive strategy as the model, then they incur extraneous cognitive load. Spans should increase if this cognitive load is reduced by external feedback. However, if people do not use this strategy, then when feedback is not provided by the experimenter (as in Barrouillet et al., 2007) the model predicts no improvement in performance in the distractor task following training.

Although I did not include the fan mechanism in the model because it has been proposed to effectuate capacity-sharing (Anderson et al., 1996), which is antithetical to TBRS, I have discussed how it is a prominent element of associative memory. It

should be reintroduced to the model in the future to reflect the diminishing predictive ability of an increasing number of associations (Anderson, 2007) rather than as a way to explain limited WMC. That being said, the model suggests that fan may benefit WM as a way to balance the differential effect of inhibition, as the reward and penalty of each, respectively, decreases with set size.

A speculation that follows from this model is that a hard division between working memory (or short-term memory) and long-term memory may not be necessary. The model fits human data using the same mechanism (retrieval) to enact attentional refreshing and LTM access. The model is not evidence against such a separation, but it does suggest that a unitary model of declarative memory is possible. This avenue merits further exploration.

## **6.7 General Impact**

In TBRS-related research, cognitive load is carefully controlled in order to measure its effect on WMC. Now that this link has been established, one may wish to study the effect of cognitive load on more macro level tasks since WM has been shown to predict performance in a variety of domain-general, high-level behavior (Conway et al., 2005). Outside of the lab, real-world work activities require extended, complex processing. Computational modeling allows for the quantification of the cognitive load of these activities in a way that traditional cognitive work analysis cannot. By tracking buffer usage over time, extreme peaks and troughs in cognitive load can be determined and used to identify choke points in work processes. This entirely new approach to work analysis may be used to more efficiently organize job procedures.

The broader contribution of my model to the ACT-R theory is to recommend a best practice for formulating WM within the architecture. While the model itself is fairly complex, the actual maintenance component is relatively simple. The refreshing

loop in my model is a special case of a general mechanism. All ACT-R models could incorporate this implementation of WM by including base-level inhibition and one production that makes a generic retrieval request any time the central bottleneck is free.

Moreover, this general mechanism could have an exceptional impact on models of goal-directed behavior. The maintenance and setting of goals within WM can be modeled by using spreading activation to allow buffer contexts, such as the goal buffer, to influence what is retrieved by the general WM loop (perhaps in cooperation with dynamic pattern matching, which allows productions to be variabilized based on buffer contents). One example where this would be helpful is in Salvucci and Taatgen's (2008) model of threaded cognition, which accounts for concurrent multitasking behavior by expanding ACT-R's goal buffer to hold multiple goal chunks. Representing different task threads, these goal chunks are acted upon serially, similar to how a list of to-be-remembered items is rehearsed one at a time in a serial memory task. It is a reasonable hypothesis that such goals are maintained by WM, and a generalized version of the mechanism in my model could be used to implement this without modifying a core assumption of the ACT-R architecture (that buffers may only hold one chunk).

## 7 Concluding Remarks

I have provided model-based support for the TBRS theory beyond that of existing computational implementations such as TBRS\* by formalizing both maintenance and processing roles of WM. My model generates the qualitative patterns of behavior observed in human data, establishing evidence for the model's core assumptions. Imperfect quantitative fits suggest further exploration of certain ancillary assumptions is needed. Prominent among these is the structure and representation of declarative memory. While I proposed a novel combination of existing theories of serial memory that I believe best integrates ACT-R mechanisms with contemporary findings, additional data is required to definitively assess this approach. I determined that distractor-response strategy, typically thought to be irrelevant to WM, has a serious impact on cognitive load and therefore WMC. Although the TBRS theory has thus far left response processing underspecified, this area must be investigated before computational modeling of WM can advance. My model provides the means for predicting results in novel empirical paradigms and for evaluating the cognitive demand of real-world tasks.

## References

- Ahn, W.-Y., Busemeyer, J. R., Wagenmakers, E.-J., & Stout, J. C. (2008). Comparison of decision learning models using the generalization criterion method. *Cognitive Science*, *32*(8), 1376–1402.
- Altmann, E. M., & Trafton, J. G. (2002). Memories for goals: An activation-based model. *Cognitive Science*, *26*(2), 39–83.
- Anderson, J. R. (1976). *Language, memory, and thought*. Hillsdale, NJ: Erlbaum.
- Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, *29*(3), 313–341.
- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.
- Anderson, J. R., & Betz, J. (2001). A hybrid model of categorization. *Psychonomic Bulletin & Review*, *8*(4), 629–647.
- 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., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*(4), 341–380.
- Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, *128*, 186–197.
- Anderson, J. R., Reder, L. M., & Lebiere, C. (1996). Working memory: Activation limitations on retrieval. *Cognitive Psychology*, *30*(3), 221–256.
- Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. *Psychology of Learning and Motivation*, *2*, 89–195.

- Baddeley, A. D. (1986). *Working memory*. Oxford: Oxford University Press.
- Baddeley, A. D. (2000). The episodic buffer: A new component of working memory? *Trends in Cognitive Sciences*, 4(11), 417–423.
- Baddeley, A. D. (2012). Working memory: Theories, models, and controversies. *Annual Review of Psychology*, 63, 1–29.
- Baddeley, A. D., & Hitch, G. J. (1974). Working memory. *Psychology of Learning and Motivation*, 8, 47–89.
- Baddeley, A. D., & Logie, R. H. (1999). Working memory: The multiple component model. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 28–61). Cambridge, England: Cambridge University Press.
- Ball, J. T. (2004). A cognitively plausible model of language comprehension. In *Proceedings of the 13th conference on behavior representation in modeling and simulation* (pp. 305–316).
- Ball, J. T. (2013). Modeling the binding of implicit arguments in complement clauses in ACT-R/Double-R. In *Proceedings of the 12th international conference on cognitive modeling* (pp. 84–88).
- Barrouillet, P., Bernardin, S., & Camos, V. (2004). Time constraints and resource sharing in adults' working memory spans. *Journal of Experimental Psychology: General*, 133(1), 83.
- Barrouillet, P., Bernardin, S., Portrat, S., Vergauwe, E., & Camos, V. (2007). Time and cognitive load in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 33(3), 570.
- Barrouillet, P., & Camos, V. (2001). Developmental increase in working memory span: Resource sharing or temporal decay? *Journal of Memory and Language*, 45(1), 1–20.
- Barrouillet, P., & Camos, V. (2010). Working memory and executive control: A

- time-based resource-sharing account. *Psychologica Belgica*, 50(3/4), 353–382.
- Barrouillet, P., & Camos, V. (2015). *Working memory: Loss and reconstruction*. Psychology Press.
- Barrouillet, P., De Paepe, A., & Langerock, N. (2012). Time causes forgetting from working memory. *Psychonomic Bulletin & Review*, 19(1), 87–92.
- Barrouillet, P., Portrat, S., & Camos, V. (2011). On the law relating processing to storage in working memory. *Psychological Review*, 118(2), 175.
- Barrouillet, P., Portrat, S., Vergauwe, E., Diependaele, K., & Camos, V. (2011). Further evidence for temporal decay in working memory: Reply to Lewandowsky and Oberauer (2009). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37, 1302–1317.
- Blessing, S. B., & Anderson, J. R. (1996). How people learn to skip steps. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(3), 576.
- Brown, G. D., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review*, 107(1), 127.
- Budiu, R., & Anderson, J. R. (2004). Interpretation-based processing: A unified theory of semantic sentence comprehension. *Cognitive Science*, 28(1), 1–44.
- Burgess, N., & Hitch, G. J. (2006). A revised model of short-term memory and long-term learning of verbal sequences. *Journal of Memory and Language*, 55(4), 627–652.
- Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review*, 108(4), 847.
- Camos, V., & Barrouillet, P. (2014). Attentional and non-attentional systems in the maintenance of verbal information in working memory: The executive and phonological loops. *Frontiers in Human Neuroscience*, 8, 900.
- Camos, V., Mora, G., & Barrouillet, P. (2013). Phonological similarity effect in complex span task. *The Quarterly Journal of Experimental Psychology*, 66(10),

1927–1950.

- Camos, V., Mora, G., & Oberauer, K. (2011). Adaptive choice between articulatory rehearsal and attentional refreshing in verbal working memory. *Memory & Cognition*, *39*(2), 231–244.
- Cao, S., Qin, Y., Zhao, L., & Shen, M. (2015). Modeling the development of vehicle lateral control skills in a cognitive architecture. *Transportation Research Part F: Traffic Psychology and Behaviour*, *32*, 1–10.
- Case, R., Kurland, D. M., & Goldberg, J. (1982). Operational efficiency and the growth of short-term memory span. *Journal of Experimental Child Psychology*, *33*(3), 386–404.
- Chow, M., & Conway, A. R. (2015). The scope and control of attention: Sources of variance in working memory capacity. *Memory & Cognition*, *43*(3), 325–339.
- Conway, A. R., Cowan, N., Bunting, M. F., Theriault, D. J., & Minkoff, S. R. (2002). A latent variable analysis of working memory capacity, short-term memory capacity, processing speed, and general fluid intelligence. *Intelligence*, *30*(2), 163–183.
- Conway, A. R., Kane, M. J., Bunting, M. F., Hambrick, D. Z., Wilhelm, O., & Engle, R. W. (2005). Working memory span tasks: A methodological review and user's guide. *Psychonomic Bulletin & Review*, *12*(5), 769–786.
- Cowan, N. (1992). Verbal memory span and the timing of spoken recall. *Journal of Memory and Language*, *31*(5), 668–684.
- Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*, 97–185.
- Cowan, N., Keller, T. A., Hulme, C., Roodenrys, S., McDougall, S., & Rack, J. (1994). Verbal memory span in children: Speech timing clues to the mechanisms underlying age and word length effects. *Journal of Memory and Language*, *33*(2), 234–250.

- Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, *19*(4), 450–466.
- Demir, M., McNeese, N. J., Cooke, N. J., Ball, J. T., Myers, C., & Friedman, M. (2015). Synthetic teammate communication and coordination with humans. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 59, pp. 951–955).
- Diamond, A. (2013). Executive functions. *Annual Review of Psychology*, *64*, 135–168.
- Dickison, D. H., & Taatgen, N. A. (2007). ACT-R models of cognitive control in the abstract decision making task. In *Proceedings of the eighth international conference on cognitive modeling* (pp. 79–84).
- Estes, W. K. (1972). An associative basis for coding and organization in memory. *Coding Processes in Human Memory*, 161–190.
- Farrell, S. (2012). Temporal clustering and sequencing in short-term memory and episodic memory. *Psychological Review*, *119*(2), 223.
- Farrell, S., Hurlstone, M. J., & Lewandowsky, S. (2013). Sequential dependencies in recall of sequences: Filling in the blanks. *Memory & Cognition*, *41*(6), 938–952.
- Fukuda, K., Vogel, E., Mayr, U., & Awh, E. (2010). Quantity, not quality: The relationship between fluid intelligence and working memory capacity. *Psychonomic Bulletin & Review*, *17*(5), 673–679.
- Grange, J. A., & Juvina, I. (2015). The effect of practice on n–2 repetition costs in set switching. *Acta Psychologica*, *154*, 14–25.
- Grange, J. A., Juvina, I., & Houghton, G. (2013). On costs and benefits of n–2 repetitions in task switching: Towards a behavioural marker of cognitive inhibition. *Psychological Research*, *77*(2), 211–222.
- Guhe, M., Pease, A., & Smaill, A. (2009). A cognitive model of discovering commutativity. In *Proceedings of the 31st annual conference of the cognitive science society* (pp. 727–732).

- Halverson, T., & Gunzelmann, G. (2011). Visual search versus memory in a paired associate task. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 55, pp. 875–879).
- Harris, J., Gluck, K. A., Mielke, T., & Moore, L. R. (2009). Mindmodelinghome... and anywhere else you have idle processors. In A. Howes, D. Peebles, & R. Cooper (Eds.), *Proceedings of the ninth international conference on cognitive modeling*. Manchester, United Kingdom: University of Manchester.
- Harrison, A. M., & Trafton, J. G. (2010). Cognition for action: an architectural account for “grounded interaction”. In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of the 32nd annual conference of the cognitive science society* (pp. 246–251). Austin, TX.
- Henson, R. N., Norris, D. G., Page, M. P., & Baddeley, A. D. (1996). Unchained memory: Error patterns rule out chaining models of immediate serial recall. *The Quarterly Journal of Experimental Psychology: Section A*, 49(1), 80–115.
- Huss, D., & Byrne, M. (2003). An ACT-R/PM model of the articulatory loop. In *Proceedings of the fifth international conference on cognitive modeling* (pp. 135–140).
- Janssen, C. P., & Gray, W. D. (2012). When, what, and how much to reward in reinforcement learning-based models of cognition. *Cognitive Science*, 36(2), 333–358.
- Juvina, I., & Taatgen, N. A. (2009). A repetition-suppression account of between-trial effects in a modified stroop paradigm. *Acta Psychologica*, 131(1), 72–84.
- Kane, M. J., Hambrick, D. Z., Tuholski, S. W., Wilhelm, O., Payne, T. W., & Engle, R. W. (2004). The generality of working memory capacity: A latent-variable approach to verbal and visuospatial memory span and reasoning. *Journal of Experimental Psychology: General*, 133(2), 189.
- Kieras, D. E., Meyer, D. E., Mueller, S. T., & Seymour, T. L. (1999). Insights into

- working memory from the perspective of the EPIC architecture for modeling skilled perceptual, motor and cognitive human performance. In A. Miyaki & P. Shah (Eds.), *Models of working memory*. New York: Cambridge University Press.
- Lebiere, C., & Best, B. J. (2009). Balancing long-term reinforcement and short-term inhibition. In *Proceedings of the 31st annual conference of the cognitive science society* (pp. 2378–2383).
- Lebiere, C., & Lee, F. J. (2002). Intention superiority effect: A context-switching account. *Cognitive Systems Research*, 3(1), 57–65.
- Lewandowsky, S., & Murdock, B. B. (1989). Memory for serial order. *Psychological Review*, 96(1), 25–57.
- Lewandowsky, S., & Oberauer, K. (2009). No evidence for temporal decay in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(6), 1545.
- Lisman, J. E., & Idiart, M. A. (1995). Storage of 7 plus/minus 2 short-term memories in oscillatory subcycles. *Science*, 267(5203), 1512.
- Lovett, M. C., Reder, L. M., & Lebiere, C. (1999). Modeling working memory in a unified architecture: An ACT-R perspective. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 135–182). Oxford University Press.
- Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis*. New York: John Wiley and Sons.
- Manly, T., Davison, B., Heutink, J., Galloway, M., & Robertson, I. H. (2000). Not enough time or not enough attention? Speed, error and self-maintained control in the sustained attention to response test (SART). *Clinical Neuropsychological Assessment*, 3(10).
- Marewski, J. N., & Mehlhorn, K. (2011). Using the ACT-R architecture to specify 39

- quantitative process models of decision making. *Judgment and Decision Making*, 6(6), 439–519.
- Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, 104, 2–65.
- Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2), 81.
- Mora, G., & Camos, V. (2013). Two systems of maintenance in verbal working memory: Evidence from the word length effect. *PloS One*, 8(7).
- Oberauer, K., & Lewandowsky, S. (2011). Modeling working memory: A computational implementation of the time-based resource-sharing theory. *Psychonomic Bulletin & Review*, 18(1), 10–45.
- Oberauer, K., Süß, H. M., Wilhelm, O., Sander, N., Conway, A. R., Jarrold, C., . . . Towse, J. N. (2007). Individual differences in working memory capacity and reasoning ability. *Variation in Working Memory*, 49–75.
- An online database for ACT-R estimated parameters.* (n.d.). (Retrieved February, 2015 from <http://www-abc.mpib-berlin.mpg.de/actrdb/>)
- Page, M., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review*, 105(4), 761.
- Pashler, H. (1984). Processing stages in overlapping tasks: Evidence for a central bottleneck. *Journal of Experimental Psychology: Human Perception and Performance*, 10(3), 358.
- Peebles, D., & Bothell, D. (2004). Modeling performance in the sustained attention to response task. In *Iccm* (pp. 231–236).
- Pirolli, P. L., & Anderson, J. R. (1985). The role of practice in fact retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(1), 136.
- Portrat, S., & Lemaire, B. (2014). Is attentional refreshing in working memory

- sequential? A computational modeling approach. *Cognitive Computation*, 7(3), 333–345.
- Pylyshyn, Z. W. (1989). The role of location indexes in spatial perception: A sketch of the first spatial-index model. *Cognition*, 32, 65–97.
- R Development Core Team. (2008). R: A language and environment for statistical computing.
- Reitter, D., Juvina, I., Stocco, A., & Lebiere, C. (2010). Resistance is futile: Winning lemonade market share through metacognitive reasoning in a three-agent cooperative game. In *Proceedings of the 19th behavior representation in modeling & simulation (BRIMS)*. Charleston, SC.
- Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive Science*, 35(4), 587–637.
- Riggs, D. S., Guarnieri, J. A., & Addelman, S. (1978). Fitting straight lines when both variables are subject to error. *Life Sciences*, 22(13), 1305–1360.
- Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(2), 362–380.
- Salvucci, D. D. (2010). Cognitive supermodels. In *Invited talk, European ACT-R Workshop*. Groningen, The Netherlands.
- Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science*, 37(5), 829–860.
- Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, 115(1), 101.
- Solway, A., Murdock, B. B., & Kahana, M. J. (2012). Positional and temporal clustering in serial order memory. *Memory & Cognition*, 40(2), 177–190.
- Spurgeon, J., Ward, G., Matthews, W. J., & Farrell, S. (2015). Can the effects of temporal grouping explain the similarities and differences between free recall

- and serial recall? *Memory & Cognition*, *43*(3), 469–488.
- Surprenant, A. M., Kelley, M. R., Farley, L. A., & Neath, I. (2005). Fill-in and infill errors in order memory. *Memory*, *13*(3-4), 267–273.
- Taatgen, N. A., Huss, D., Dickison, D., & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General*, *137*(3), 548.
- Taatgen, N. A., Van Rijn, H., & Anderson, J. R. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review*, *114*(3), 577.
- Tamborello, F. P., & Byrne, M. D. (2006). Adaptive but non-optimal visual search behavior with highlighted displays. In *Proceedings of the seventh international conference on cognitive modeling* (Vol. 8, pp. 316–321). Trieste, Italy.
- Thomson, R., & Lebiere, C. (2013). A balanced Hebbian algorithm for associative learning in ACT-R. In R. West & T. Stewart (Eds.), *Proceedings of the 12th international conference on cognitive modeling*. Ottawa, Canada.
- Thomson, R., Lebiere, C., Anderson, J. R., & Staszewski, J. (2015). A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture. *Journal of Applied Research in Memory and Cognition: Special Issue on Modeling and Aiding Intuitions in Organizational Decision Making*, *4*(3), 180–190.
- Towse, J. N., & Hitch, G. J. (1995). Is there a relationship between task demand and storage space in tests of working memory capacity? *The Quarterly Journal of Experimental Psychology*, *48*(1), 108–124.
- Towse, J. N., Hitch, G. J., & Hutton, U. (1998). A reevaluation of working memory capacity in children. *Journal of Memory and Language*, *39*(2), 195–217.
- Turner, M. L., & Engle, R. W. (1989). Is working memory capacity task dependent? *Journal of Memory and Language*, *28*(2), 127–154.

- Unsworth, N., & Engle, R. W. (2006). A temporal–contextual retrieval account of complex span: An analysis of errors. *Journal of Memory and Language*, *54*(3), 346–362.
- Vergauwe, E., Barrouillet, P., & Camos, V. (2010). Do mental processes share a domain-general resource? *Psychological Science*, *21*(3), 384–390.
- Vergauwe, E., Camos, V., & Barrouillet, P. (2014). The impact of storage on processing: How is information maintained in working memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *40*(4), 1072.
- Vergauwe, E., & Cowan, N. (2014). A common short-term memory retrieval rate may describe many cognitive procedures. *Frontiers in Human Neuroscience*, *8*, 126.