

The Pennsylvania State University

The Graduate School

College of the Liberal Arts

TESTING PREDICTIONS OF THE BINDING POOL MODEL OF
VISUAL WORKING MEMORY

A Dissertation in

Psychology

by

Garrett Swan

© 2017 Garrett Swan

Submitted in Partial Fulfillment
of the requirements
for the Degree of

Doctor of Philosophy

August 2017

The dissertation of Garrett Swan was reviewed and approved* by the following:

Brad Wyble
Associate Professor
Dissertation Adviser
Chair of Committee

Rich Carlson
Professor of Psychology

Frank Hillary
Associate Professor of Psychology

John Collins
Professor of Physics

Melvin Mark
Professor of Psychology
Head of the Department of Psychology

*Signatures are on file in the Graduate School

ABSTRACT

Throughout our daily lives, we use visual working memory (VWM) in both simple and complex tasks to briefly maintain visual information. How VWM works has been a hotly debated topic in psychology for many years. This thesis focuses on testing predictions generated from a computational model of VWM called the Binding Pool model. In this model, a memory is a set of binding pool nodes activated from concurrent activity of the visual features that compose an object and a token that individuates memories.

The first prediction focused on how similarity affects memory precision. The model predicts that highly similar and dissimilar stimuli are more precise than stimuli with intermediate similarity. In two experiments, the results supported the prediction of the model.

The second prediction focused trade-offs in retrieval precision as a function of the number of relevant features. The model predicts this trade-off should be present for multiple objects. In two experiments, participants reported the task irrelevant feature with coarse precision and were significantly less precise when the task irrelevant feature became relevant to retrieval, which support the prediction from the model. However, the ability to code features coarsely needed to be added to the model.

The third prediction focused on how repetitions are encoded into memory. The model predicts that repetitions will be less precise than a single object. In two experiments, participants were significantly more precise with repetitions than a single object and random multiple objects. This result failed to support the model.

A fourth prediction about the underpinning of the EEG correlates of VWM was also generated and tested using a delayed estimation task. The initial findings were mixed, but may provide insight into how the P3 components may relate to the disambiguation of targets from non-targets.

As a result of these experiments, there were significant changes implemented to how the model encodes information. Taken together, these new results provide additional benchmarks for models of memory and helped the shape the architecture of the Binding Pool model to become a more general model of memory.

TABLE OF CONTENTS

LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
ACKNOWLEDGEMENTS.....	ix
CHAPTER 1. Background and Introduction.....	1
Measuring Memory Quality.....	2
Overview of Thesis.....	6
CHAPTER 2. The Binding Pool Model.....	8
Types.....	11
Tokens.....	12
Binding Pool.....	13
Connectivity.....	14
Encoding and Retrieval.....	14
Encoding Multiple Objects.....	20
Parameter Fitting.....	21
Conclusions.....	22
CHAPTER 3. The Influence Of Similarity On Memory Precision.....	24
Experiment 1.....	26
Results And Discussion.....	28
Experiment 2.....	32
Results.....	33
Post-hoc analyses.....	34
General Discussion.....	35
Changes To The Mode.....	37
Type Activation Decoding.....	37
Conclusions.....	39
CHAPTER 4. The Storage of Task Irrelevant Features for Multiple Objects.....	40
Experiment 1.....	44
Results And Discussion.....	46
Experiment 2.....	52
Results.....	52
General Discussion.....	56
Changes To The Model.....	59
Adding Featural Attention.....	59
New Encoding Mechanism.....	62

Parameter Fitting For New Model.....	63
Conclusions.....	65
CHAPTER 5. The Precision of Repetitions.....	66
Experiment 1.....	67
Results And Discussion.....	70
Experiment 2.....	72
Results.....	74
General Discussion.....	77
Changes To The Model.....	78
Repetitions Bound To The Same Token.....	78
Boosting Type Activation.....	81
Conclusions.....	82
CHAPTER 6. Predicting ERPs Using The Binding Pool Model.....	83
P3 Components.....	84
CDA Component.....	85
Predicting ERPs With The Binding Pool Model.....	86
Experiment.....	87
Behavioral results.....	92
EEG results.....	93
General Discussion.....	96
Conclusion.. ..	100
CHAPTER 7. Conclusions Regarding The Binding Pool Model.....	101
Tested Predictions.....	101
Changes To The Model.....	103
New Predictions.....	106
General Discussion.....	108
Expanding The Binding Pool Model.....	110
Conclusion.....	112
References.....	113
Appendix A. Matlab Code For Binding Pool Model.....	122
Appendix B. Matlab Code For Binding Pool Model.....	129

List of Figures

Figure 1. Illustration Of Thesis Goals.....	6
Figure 2. Illustration Of The Binding Pool Model.....	9
Figure 3. Distributed Projections From Type To Binding Pool.....	12
Figure 4. Precision And The Number Of Active Binding Pool Nodes.....	14
Figure 5. Encoding In The Binding Pool Model.....	16
Figure 6. Retrieval Stage 1 In The Binding Pool Model.....	17
Figure 7. Retrieval Stage 2 In The Binding Pool Model.....	19
Figure 8. Decoding Type Activity.....	20
Figure 9. Error Distributions For Different Set Sizes From Binding Pool Model.....	21
Figure 10. Prediction From Binding Pool Model About Similarity And Precision....	25
Figure 11. Paradigm For Experiment 1 In Chapter 3.....	28
Figure 12. Parameter Estimates For Experiment 1 And Experiment 2.....	29
Figure 13. Effect Of Similarity On Retrieval Precision For Experiment 1	30
Figure 14. Paradigm For Experiment 2 In Chapter 3.....	32
Figure 15. Effect Of Similarity On Retrieval Precision For Experiment 2	33
Figure 16. Effect Of Serial Position On Similarity Results.....	35
Figure 17. Effect Of Two Vectors On Mean Vector Angle.....	38
Figure 18. Retrieved Type Activity As A Function Of Input Category.....	39
Figure 19. Different Encoding Types From Swan, Collins, And Wyble (2016).....	41
Figure 20. Paradigm For Experiment 1 In Chapter 4.....	46
Figure 21. Error Distributions For Experiment 1.....	47
Figure 22. Parameter Estimates For Experiment 1.....	48

Figure 23. Effects Of Trial On Recall Precision For Experiment 1.....	50
Figure 24. Error Distributions For Experiment 2.....	53
Figure 25. Parameter Estimates For Experiment 2.....	53
Figure 26. Effects Of Trial On Recall Precision For Experiment 2.....	55
Figure 27. Simulation Of Experiment 1 With Binding Pool Model.....	61
Figure 28. Illustration Of New Encoding Mechanism In Binding Pool Model.....	62
Figure 29. Simulation of Bays Et Al. (2009) With Binding Pool Model.....	65
Figure 30. Prediction About Precision Of Repetitions From Binding Pool Model.....	67
Figure 31. Paradigm For Experiment 1 In Chapter 5.....	68
Figure 32. Parameter Estimates For Experiment 1.....	71
Figure 33. Paradigm For Experiment 2 In Chapter 5.....	74
Figure 34. Parameter Estimates For Experiment 2.....	75
Figure 35. Simulation of Experiment 2 With Binding Pool Model Adjustments.....	79
Figure 36. Simulation Of CDA Component With Binding Pool Model.....	87
Figure 37. Paradigm For Experiment In Chapter 6.....	90
Figure 38. Predictions From Binding Pool Model For P3a, P3b, And CDA.....	92
Figure 39. ERPs of P3/P4 Electrode Time Locked To Stimulus Onset.....	94
Figure 40. ERPs of Pz Electrode Time Locked To Stimulus Onset.....	95
Figure 41. ERPs of Fz Electrode Time Locked To Stimulus Onset.....	95
Figure 42. ERPs of P3/P4 Electrode Time Locked To Pre-Cue Onset.....	99
Figure 43. Effect Of Number Of Features On Retrieval Precision.....	110

List of Tables

Table 1. Parameter Fits For Binding Pool Model from Swan and Wyble (2014)	22
Table 2. Parameter Fits For Binding Pool Model With New Encoding.....	64
Table 3. Parameter Estimates As Function Of Similarity In Chapter 5.....	81
Table 4. Parameter Estimates For Experiment In Chapter 6.....	92

Acknowledgments

There is a single name on the title page, but a dissertation is not the work of a single person. I wouldn't be here if not for the people around me. I know I am not always the most appreciative person, but I am sincerely grateful for everything everyone has done for me. I hope that someday, I can return the favor.

First, I would like to thank the inanimate objects in my life that have given me strength. Specially, I want to thank the miracle that is coffee, mustard, beer, Wegmans, Cadbury Mini Eggs, reddit, my Mini Cooper, and tortilla chips. I also want to thank the people in my life that do not know that they are people in my life; Kendrick Lamar, T. J. Warren, Becca who cuts my hair, Kang-ho Song, the team behind the Dark Souls series, George R. R. Martin, and that one guy whom makes delicious meatball subs at Wegmans. You know who you are.

I want to give a special thanks to my family. I am beyond lucky to have parents, Steve and Julie, that would give anything to help me succeed. Thank you Zach for finishing your dissertation, so that I can finally finish mine. Thank you Seth for setting the upper limit on what is the acceptable quantity of alcohol. Thank you Landon for properly showing me how to drive over a tree stump. And lastly, thank you Colter for going to Japan before me because that was totally super cool awesome.

I also want to thank everyone that I've shared some time with while in State College. Some of that time was ridiculous, like the Chicago Psychonomics hotel room and Chicago hot pot. Sometimes, it was enlightening, like trivia at Dark Horse. Occasionally, it was enthralling, like watching Game of Thrones together. More often than not, we single handedly keep the tatter tot empire afloat. Most of the time, the laughs and smiles came easy. The best times were weird. For some, we shared something much more than just friendship. No matter the place or time, I will never forget any of you and I will always have your back. Thank you, Rachel Bernier, Federica Bulgarelli, Hui Chen, Andrew Conyers, Dan Elbich, Tyler Garner, Victoria Gertel, Angela Grant, Julia Hershey, John Huhn, Libby Jones, Jennifer Legault, Natalie Motta-Mena, Giorgia Picci, Cory Potts, Cristina Roman, Ben Schloss, Caitlin Ting, Indira Turney, and all of those other people I can't remember.

I want to especially thank my lab mate, Chloe. Chloe, without you, I would have succumbed to science madness years ago. Having you in the lab makes the good times better and the tough times tolerable. And deep down, I know you will be a fine replacement lab barista.

I also want to thank my brothers from across the sea. Berry van den Berg, George Prichard, Nico Broers, Rob Nijenkamp, and of course the one and only Florian Sense. Though we only see each other once or so a year, it is always incredible and reminds me that there is still wonder and excitement on this Earth.

Thank you to my committee members, Rich Carlson, Frank Hillary, and John Collins. My research skills have come a long way in part from your guidance. I also want to thank the Alumni Association for awarding me with the Alumni Association Dissertation Award. And I also want to thank the Psychology Department. I absolutely appreciate your patience with my shenanigans, the noises I make with

my hands, and for my knack for always submitting documents a day late. Also, I would like to thank the National Science Foundation, the Australian Academy of Science, and Jennifer Fleming and David Shum for their assistance with my research conducted in Australia.

And finally, I want to thank my mentor and advisor, Brad Wyble. It wasn't always easy. In the deep recesses of my mind, there are calluses built from those long nights getting the Binding Pool model into shape for publication. He believed in me even when I didn't believe in myself. I leave your tutelage as a wise, kind, and Matlab fluent science person man.

-Science is whatever we want it to be – Dr. Leo Spaceman

Chapter 1. Background and Introduction

A task as simple as turning on a coffee pot or as complex as flying an aircraft requires the successful maintenance and retrieval of information from working memory. Working memory is primarily differentiated from other types of memory, such as sensory or long-term memory, by its limited storage capacity (Baddeley & Hitch, 1974). This attribute by itself, however, belies a construct that is much more complex and interesting. Characterizing this system remains an active debate in neuroscience and cognitive psychology (Suchow, Fougnie, Brady, & Alvarez, 2014; Ma, Husain, & Bays, 2014).

In order to better understand working memory, researchers have typically bifurcated working memory storage into visual and verbal working memory. These systems are usually treated as separate and independent because information retained in one system seemingly does not affect the capacity of the other (Baddeley & Hitch, 1974; c.f. Salmela, Moisala, & Alho, 2014). Early quantifications of the capacity of working memory, which were typically focused on verbal working memory, reported approximately 7 plus or minus 2 items could be held in (Miller, 1956). Though this number has fluctuated (e.g. 4, Cowan, 2001 or 2, Gobert & Clarkson, 2004), capacity is a fundamental topic in studying visual working memory (VWM), which is the limited capacity system for briefly maintaining visual information. Characterizing VWM capacity is important because VWM capacity has been shown to be indicative of a variety of cognitive abilities (Conway, Kane, & Engle, 2003).

Capacity in VWM is conventionally measured by sequentially presenting participants with two arrays of objects (e.g. color squares) separated by a blank screen, or retention interval (Phillips, 1974). Participants are asked to detect and report whether a change occurred in the stimuli in the second array, or test array, which happens on 50% of trials in most studies. In a typical change detection task, the number of objects is manipulated to measure the point at which the number of objects exceeds VWM capacity and hit rate declines. Several formulas have been developed to estimate VWM capacity as a function of the number of objects that

presented to the participant in the test array and the proportion of hits and false alarms (Pashler, 1988; Cowan et al., 2005). In addition to these formulas, there are other computations (e.g. Rouder et al. (2008) that account for guessing in a change detection task.

Inherent in these calculations of memory capacity is that the amount of visual information below capacity is maintained without any corruption or degradation. This assumption was best exemplified by a set of studies from Luck and Vogel (1997), which demonstrated almost ubiquitously that if the number of objects presented was below capacity, regardless of the complexity of the objects, participants' hit rate was near ceiling. This result culminated into what is commonly called the slot model of VWM, which presumed that objects could be maintained in discrete slots until there were no slots available for the remaining objects.

Measuring memory quality

In 2004, there was a shift in how VWM was measured. Wilken and Ma (2004) published a methodology whereby precision of a visual memory was recorded using a recall task called delayed estimation¹. In a delayed estimation task, participants are presented with a single array of objects (e.g. colored patches) much like the study array in change detection. However, unlike change detection, participants respond by moving a mouse cursor to a color wheel² containing the possible colors that could have been presented. Wilken and Ma then looked at the error, or the difference between the reported color and the cued presented color, as a function of the set size. The researchers found that as they increased the number of color

¹ Note that a similar methodology was used by Prinzmetal et al. (1997) to study how attention influences perceived color.

² If reporting direction or orientation, it may be a gray circle with a rotating object in the center (Swan, Collins, & Wyble, 2016). Other researchers do not present a wheel and have participants scroll through color space using a dial (e.g. Bays, Gorgoraptis, Wee, Marshall, & Husain, 2011). Despite the differences in the response screen, the vital component is the recall nature of the task.

patches, the standard deviation of the distribution of errors increased, which was interpreted as a decrease in the quality of the memory.

Unlike what would be predicted by the slot model, Wilken and Ma (2004) found that memory precision decreased from set size 1 to set size 2. This is a controversial result because both set size 1 and set size 2 are below memory capacity in traditional slot models. Slot models of memory would have predicted no difference in precision given that both objects could be successfully stored in separate memory slots. One possible explanation for this decrease in memory precision when set size is below capacity is an increase in spurious guesses as set size increases (Zhang & Luck, 2008). To account for this, the response distribution was analyzed using a mixture model with a central Gaussian component (von Mises),

$$\phi(x|\mu, \kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I(\kappa)} \quad [1]$$

where, κ is the dispersion, or the width, and μ is the mean of the distribution, and a uniform distribution. This mixture model was formalized as,

$$p(\hat{\theta}) = (1 - P_m)\phi_\sigma(\hat{\theta} - \theta) + P_m \frac{1}{2\pi} \quad [2]$$

with parameter $\hat{\theta}$ being the reported response, θ being the presented response of the cued color, P_m being the probability of guessing and σ being the precision of the central distribution. Both parameters were estimated using maximum likelihood estimation, whereby the set of parameters that maximizes the probability of finding that specific distribution are selected. When correcting for guesses, it appears that memory precision plateaus when exceeding memory capacity. However, memory precision still declined rapidly when the set size was below capacity. To explain these results using a slot model, Zhang and Luck (2008) posited that unused slots

(i.e. when set size is below capacity) could represent a single object multiple times, thereby increasing that object's precision and explaining why set size 1 results in a higher precision than set size 2. This model is referred to as the slots+averaging model.

Shortly after the correction for guessing proposed by Zhang and Luck, Bays et al., (2009) proposed an additional correction to the data to account for retrieval errors, which may arise more often in large set sizes. This was formalized as,

$$p(\hat{\theta}) = (1 - P_u - P_{nt})\phi_\sigma(\hat{\theta} - \theta) + \gamma \frac{1}{2\pi} + \beta \frac{1}{m} \sum_m^i \phi_\sigma(\hat{\theta} - \theta_i^\circ) \quad [3]$$

with additional parameters θ_i° being the presented response of the distractor and P_{nt} being the proportion of responses that are retrieval errors. Like the mixture model proposed by Zhang and Luck, maximum likelihood is used to estimate the best fitting parameters that characterize the distribution.

When using this mixture model, Bays et al. (2009) found that memory precision does continue to decline when the set size exceeds capacity. Bays et al. then proposed a resource model whereby VWM precision for a set of objects is dependent upon a fixed pool of resources. These resources are allocated to different memory representations, and in the case of a normal delayed estimation task, the resources are allocated proportionately. When there is a pre-cue for a specific object, that object will receive more resources, as evident from an increase in recall precision (Bays et al., 2009). Similarly, in sequential displays, there is a strong recency effect, which suggests that resources are allocated quickly (Gorgoraptis, Catalao, Bays, & Husain, 2011). A similar model that borrows from the resource model is called the variable precision model, which also has a fixed pool of resources to draw upon, but the resources instead are variably allocated to objects instead of proportionally in normal delayed estimation tasks (van den Berg, Shin, Chou, George, & Ma, 2012). This variability is evident across trials and objects (van den Berg et al., 2012; Fougnie, Suchow, & Alvarez, 2012).

As a result of the delayed estimation task, researchers have had to fundamentally change how VWM is theorized. Though some researchers have focused on which combination of slots and resources produces the best fit to the data (e.g. van den Berg, Awh, & Ma, 2014), other researchers have focused on developing explanatory accounts of the underlying mechanisms of VWM. However, most of these models do not provide a plausible neural mechanism for how resources may be distributed across objects in memory. For example, models where maintenance and the coding of the visual features are conflated as the same component (e.g. Wei, Wang, & Wang, 2012; Franconeri, Alvarez, & Cavanagh, 2013) would have difficulty representing exact repetitions because there is no explicit pointer or token to individuate objects. In addition, these models predict interference arises when objects have similar visual features, whereas several studies have found benefits in memory capacity (Lin & Luck, 2009) and precision (Oberauer & Lin, 2017) for increased similarity between objects. Other models, such as the probabilistic palimpsest model (Matthey, Bays, & Dayan, 2015) and an interference model by Oberauer and Lin (2017), store information as localist representations³, which may have difficulties with scaling (i.e. combinatorial explosion) when representing more complex objects that have many features.

The model that is the focus of this thesis is called the Binding Pool model (Swan & Wyble, 2014), which is a mechanistic computational hybridization of the slot and resource models. The Binding Pool model stores information as a conjunction of the visual features that compose an object called types and a pointer called a token. The information is maintained in a distributed network of neurons called the binding pool. With these three layers (types, binding pool, tokens), the Binding Pool model is able to simulate a variety of change detection and delayed estimation findings (see Swan & Wyble, 2014) and generates a variety of testable predictions.

³ Oberauer and Lin (2017) actually posit that the representation is in the form of synapses instead of neurons, which is less likely to be susceptible to combinatorial explosion.

Overview of thesis

The remainder of this thesis focuses primarily with testing and developing the Binding Pool model. Test, in this context, corresponds to comparing simulations of the model quantitatively and qualitatively to behavioral data. Converging data provides support for the mechanisms of the model. Alternatively, a failure to find convergence between data and model simulations demonstrates that the model is incorrect. However, instead of throwing the baby out with the bath water⁴, incorrect predictions inform limitations of the model and thereby promote development by establishing new benchmarks for the model to be able to explain. Then, a new prediction from the model are generated and tested in a pilot experiment. Developing and testing predictions of models is motivated by the scientific method, and is thus a vital part of understanding a construct (Figure 1).

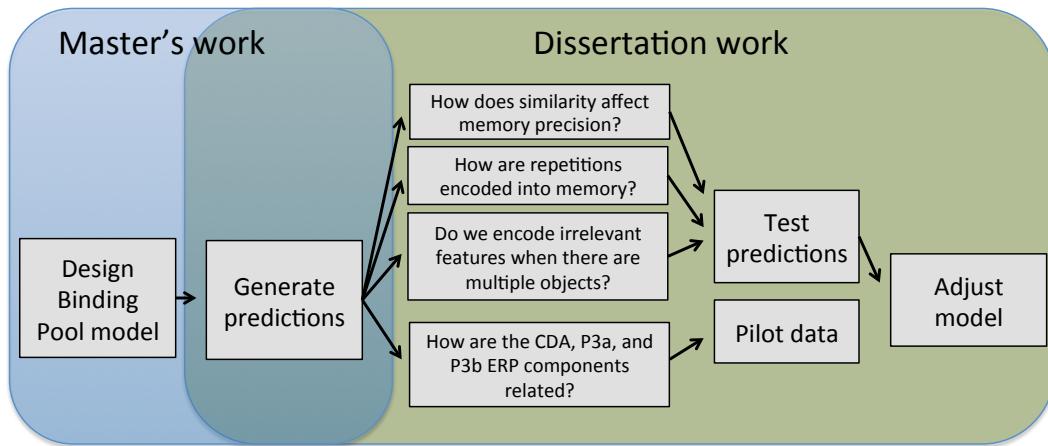


Figure 1: Illustration of the goals of this thesis

In this thesis, the mechanisms of the Binding Pool model are first detailed (see Appendix A for the Matlab code). Then, three studies are then described that test various predictions of the Binding Pool model. The three predictions focus on

⁴ Though colloquial, ‘throwing the baby out with the bath water’ is apt for describing what to do with a model after incorrectly predicting behavior. That is, entirely discarding a model because it was incorrect for a single prediction seems drastic in most circumstances without considering adaptions to the model.

how similarity in feature space affects memory precision, how task irrelevant features are encoded when multiple objects are presented, and how repetitions are encoded into memory. Testing these predictions is important to our understanding of how objects and features interact in VWM because there are many instances where multiple objects are briefly maintained (e.g. memorizing a grocery list or keeping track of airplanes as an air traffic controller). Following these experiments, an additional prediction is introduced that focuses on predicting how the relationship between different EEG signals and VWM precision

Testing these predictions is important for finding evidence to constrain the Binding Pool model and for helping to isolate how different memory processes are distributed throughout the brain. However, more importantly, the results of these experiments are informative for developing more general purpose theories of how memory works. Specifically, knowing exactly what pieces of attended information are encoded into memory and knowing how presented information interacts in memory are vital questions to understanding how working memory works.

Chapter 2. The Binding Pool model

The Binding Pool model simulates how neural resources are dynamically allocated across visual objects in a memory display and how the representations of those objects are retrieved from memory⁵. The Binding Pool model is conceptually a hybridization of the slot and resource models and borrows many mechanisms from the Simultaneous Type, Serial Token model of visual attention (Bowman & Wyble, 2007; Wyble, Bowman, &, Nieuwenstein, 2009). In the Binding Pool model, visual information is represented in memory as a distributed network of links between the features that make up a presented object and an index that individuates different representations, which are described as types and tokens, respectively (Figure 2). The links between the types and tokens are maintained in an amodal pool of nodes called the binding pool. When multiple objects need to be encoded, there is crosstalk between the different links, which results in an increase in interference between the different representations. This increased interference leads to worse precision and an increase in guessing and retrieval errors in delayed estimation tasks (Bays et al., 2009) and decreasing hit rate and increasing false alarms with increasing set sizes in change detection tasks (Keshvari et al., 2013)

⁵ Note that the Binding Pool model simulates just the binding of features into memory and the retrieval of those features. The model does not simulate the entirety of processes that govern working memory, such as understanding the task, object recognition, manipulation of objects in memory, cognitive control, affect, or learning. Though some of these processes could eventually be implemented, they are currently outside the scope of the model.

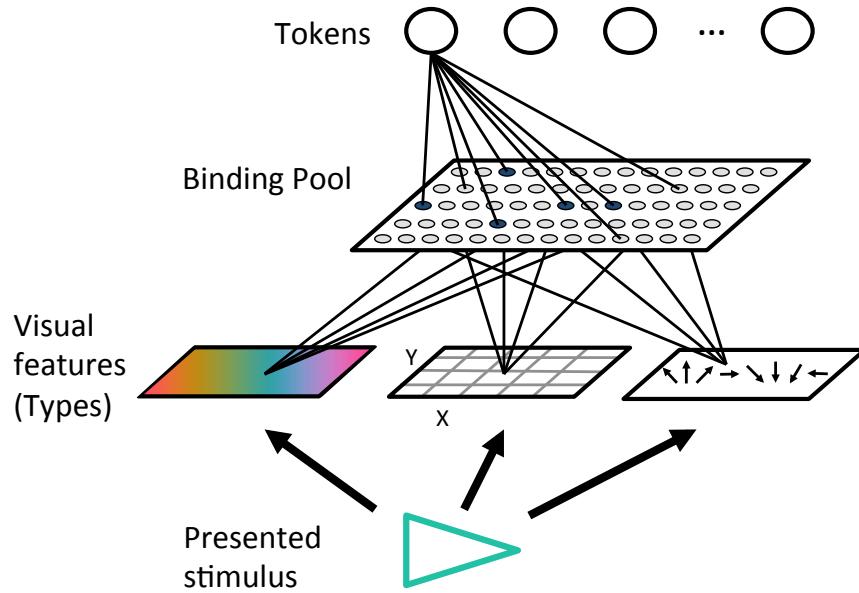


Figure 2: Illustration of the Binding Pool model. In this illustration, a colored oriented triangle is being encoded. The features of the triangle (i.e. color, location, and orientation) are being activated in the feature layers (types). The types are then projecting to the binding pool. The binding pool nodes connected to both these types and a token become activated, thereby binding those features into a single indexed representation

Before describing the model in more depth, it is important to establish the constraints that govern the mechanisms of the model. Here, constraints are the set of rules with which the model must adhere, usually derived from data. If, for example, a model of VWM is only constrained by the number of objects that could be maintained, then that model would not be useful when thinking about how stimuli of different complexities may influence memory storage, how precision may vary as a function of different task parameters (i.e. retro-cues), or how capacity is implemented in the brain. Therefore, it is reasonable to assume that the more constraints imposed on a model, the more likely that model reflects the underlying mechanisms of the construct. The following constraints were used when developing the architecture of the Binding Pool.

First, working memory representations need to be indexed in some way, such that individual features of specific objects can be retrieved. This has been demonstrated with the delayed estimation task, where a specific color is retrievable

from a color patch at a given location (e.g. Bays et al., 2009). Furthermore, multiple copies of information must be able to be encoded, and for that to be possible, repetitions must be individuated in some way⁶. Second, memory representations must be content addressable. In other words, any feature of a representation can retrieve any other feature of that representation. For example, Gorgoraptis et al. (2011) displayed multiple oriented color bars and used color to cue which bar's orientation needs to be retrieved. Content addressability occurs even when participants are not expecting to report information in a different way, such as unexpectedly switching from a recognition task to a recall task (Swan, Wyble, & Chen, 2017).

Third, there is interference between objects when multiple objects are encoded. For example, when two Gabor patches with different spatial frequencies are encoded, participants respond with a bias in the direction of the spatial frequency of the non-reported Gabor patch (Huang & Sekuler, 2010). In another example, similar colors are reported with better precision than non-similar colors (Oberauer & Lin, 2017), which further suggests multiple objects are interacting in memory. A fourth constraint is that information is rapidly encoded into VWM (Vogel, Woodman, & Luck, 2006). Information is also rapidly forgotten, such that a response on trial n is minimally influenced by the information presented on trial n-1 (Huang et al., 2010).

⁶ This problem of individuation similarly arises when developing neural architectures of language. Jackendoff (2002) brings up the example of maintaining this sentence in memory, "The big star is next to a small star". In order to do that, one has multiple repetitions of the word 'star' that need to be individuated in some way. One possibility is to have a node that represents 'star' at every possible position, but that seems computational intensive. Alternatively, there could be a set of pointers and a single representation of 'star'. This second view is similar to the token and type layers in the Binding Pool model.

Types

The content of a memory is a combination of the features (e.g. color, shape, size) of the originally presented or imagined information. In the experiments typically simulated using the Binding Pool model, types are conceptualized as distinct layers that represent abstract features, such as color hue, line orientation, and location, among other properties. However, the extent of types as a construct extends beyond simple visual features. For example, when remembering a scene, it is unlikely that the memory for that scene is represented as the basic physical characteristics. Instead, it is more likely the memory is constructed using more abstract concepts that exist in a hierarchy that increases in complexity (Orhan & Jacobs, 2014) and is learned from experience over a lifetime (Vanrullen, 2009).

In the model, type layers are implemented as discrete neural units, regardless of whether the feature is continuous (e.g. colors from a color spectrum) or categorical (e.g. identity of a letter). If a type layer represents a continuous feature, each node in that layer represents an evenly spaced point in feature space and a feature value that falls between those points is interpolated between multiple type nodes. In addition to this interpretation, if the feature space being represented is continuous, then neighboring type nodes share more overlapping connections in the binding pool than non-neighboring type nodes (Figure 3). For categorical features, such as the identity of a letter, there is no overlap between the connectivity patterns of neighboring type nodes.

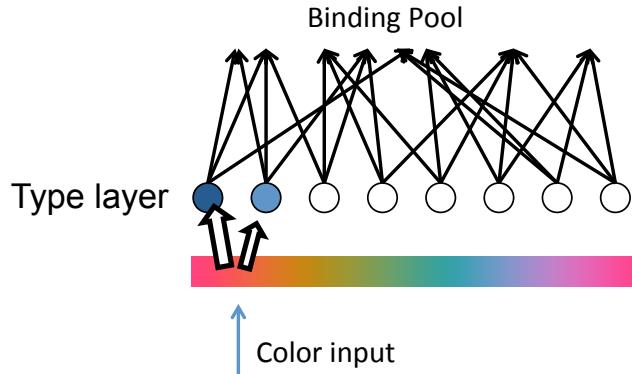


Figure 3: Illustration of a continuous type dimension. The white arrows with the black outline correspond to the interpolated activity between two type nodes when a presented feature value is between two type nodes. Neighboring type nodes share more overlapping projections than non-neighboring type nodes. The distributed nature of the binding pool is also illustrated here, as indicated by multiple type nodes connecting to the same binding pool node.

Tokens

Unlike types, tokens are non-stimulus specific, but instead provide a mechanism for individuating presented objects regardless of the attributes of those objects. It is assumed that during the presentation of stimuli, tokens are automatically assigned to different objects through the visual system's detection of closed contours, much in the same way as place tokens (Marr, 1976) or object files (Kahneman, Treisman, & Gibbs, 1992). Once an object has been assigned to a token, then that object's features could be bound to the token in the binding pool.

Similar to the idea of slots in the slot model, the number of available tokens determines how many objects could be encoded into memory. Unlike the fixed number of slots presumed in the slot model, the number of available tokens is assumed to vary according to the task, attentional control from one moment to the next, and individual capacity limits. For simplicity, on each trial, the number of tokens available is implemented in the model by drawing a number from a uniform distribution [2, 7]. If there are more presented objects than tokens, then the

remaining objects are not encoded into memory and thus have no influence on activity in the binding pool. This is one type of memory capacity built into the model.

Binding Pool

The links between different types and tokens occurs in the binding pool, which is implemented as a set of nodes that maintain information in the form of elevated activation. In its current implementation, only types and tokens influence activity in the binding pool through these links, which are distributed in nature (McClelland & Rumelhart, 1988; Murdock, 1983). The links are the only form of information storage in the model, meaning that the activity in the types and tokens is transient and dependent upon visual presentation. Once binding pool nodes become active through the binding of types and a token, that activity persists through self-excitation in the absence of further input. This type of mechanism allows the model to simulate rapid encoding and forgetting through excitation and inhibition of nodes (for more details regarding how these mechanisms may be implemented, see Wyble, Bowman, &, Nieuwenstein, 2009)

The second form of capacity limitation in the Binding Pool model is the fixed size of the binding pool. The size of the binding pool, or the number of total binding pool nodes, determines how many total binding pool nodes represent a stimulus, assuming a fixed proportion of active connectivity from the types and tokens. The more binding pool nodes represent a stimulus, the greater that stimulus's precision when retrieved (Figure 4).

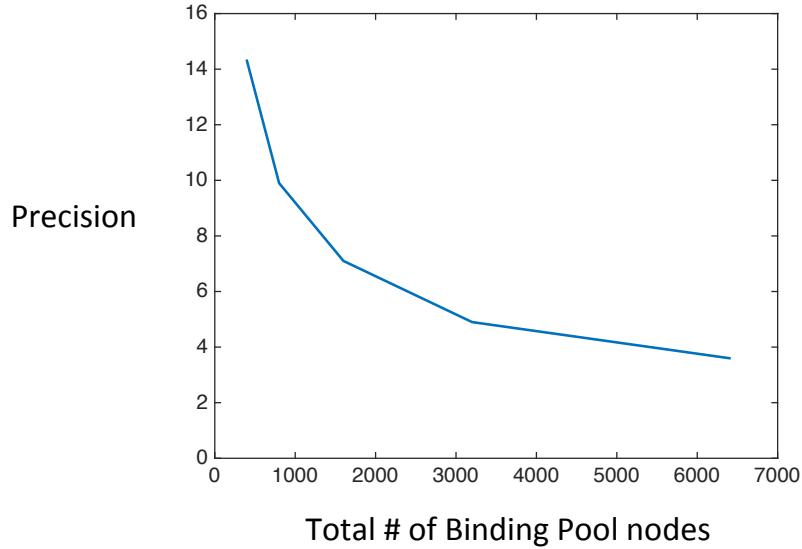


Figure 4: Precision as a function of the total number of binding pool nodes with the connection weights fixed.

Connectivity

Connectivity in the Binding Pool model is bidirectional, such that, a type or token node that activates a given binding pool node during encoding also receives input from that same binding pool node during retrieval. Each node in the type and token layers are connected to a proportion of the nodes within the binding pool. The connectivity between type layers that represent continuous feature dimensions is pseudorandomly determined (described in greater detail below), whereas the connectivity is entirely random for categorical type layers and the token layer.

Encoding and Retrieval

In order to illustrate how the model encodes and retrieves information, a delayed estimation task will be described with the presentation of a single to-be-remembered color patch where location is the retrieval cue.

First, connection matrixes are established⁷ between the type layers to the binding pool and the token layer to the binding pool. The activity in the connectivity matrix is either 1 or 0 and the proportion of active connectivity is determined by parameter, α . For categorical type dimensions, the connectivity is randomly established, such that there is no correlation in the projections between neighboring type nodes. When simulating feature dimensions that are continuous in nature (e.g. a color from a color wheel), the connectivity is pseudorandomly determined, such that neighboring type nodes share connections. This is done by randomly selecting two neighboring type nodes and then setting a binding pool node that has a common connection between the two type nodes to an equal value until a proportion (parameter L) of connections are in common.

Next, a stimulus color value is randomly drawn from a uniform distribution [1, 360]. Given that, for example, color is represented in this task as a continuous feature, the activation in the type layer may need to be interpolated between multiple type nodes. Location, in this example, is categorical, so just a single type node becomes active to represent the location of the stimulus.

Next, the number of available tokens is randomly drawn from a uniform distribution [2, 7]. In this illustration, a single object is always encoded, but if the number exceeded the number of available tokens, objects greater than the capacity would not be encoded. To encode each object in sequence, excitatory projections from each of the type layers converge at the binding pool using the pre-established pseudorandom connectivity patterns (Figure 5). These converging projections activate the subset of the binding pool nodes that receive input from the two type layers being represented in this task and the token layer. This is formalized as,

$$B_\beta = B_\beta + Z_t N_{t,\beta} \sum_{f=1}^m \sum_{g=1}^{n_m} X_{f,g} L_{f,g,\beta} \quad [4]$$

⁷ When simulating a study with the model, a single randomly generated connectivity matrix represents an individual participant, such that each participant has a different connectivity matrix.

where B is the set of binding pool nodes in the binding pool indexed by β , Z is the token layer indexed by the number of tokens, t , N is the connectivity matrix between the token layer and binding pool. The type layer, X , is indexed by the specific type layer, m , and the number of type nodes in that layer, n_m . The type to binding pool connectivity matrix is L .

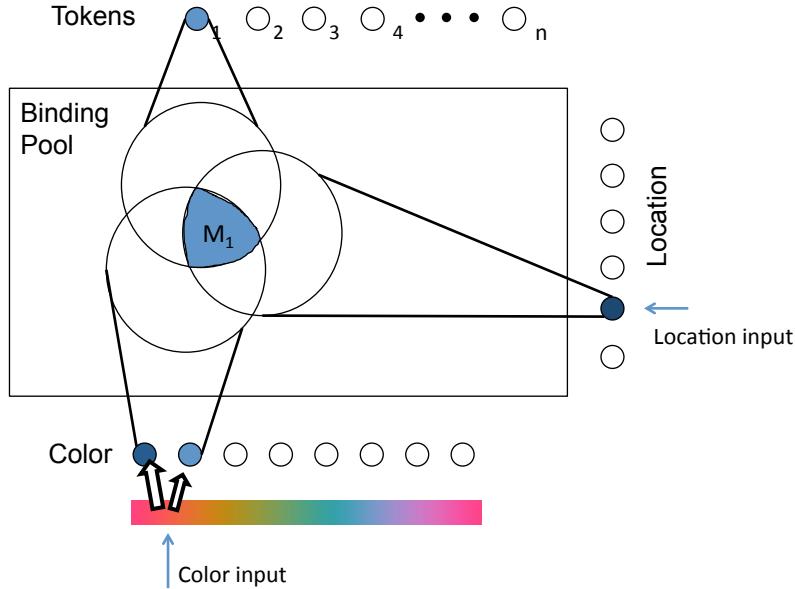


Figure 5: Illustration of how information is encoded into the binding pool. The inputted color and location features of a stimulus are activated in the model. These activations in the type layer are then projected into the binding pool and bound together with the projections of an activated token. The binding pool nodes connected to all projections (i.e. M_1) become active and represent the cell assembly for the object being remembered. Note that here, the activated binding pool nodes are spatially represented, but in reality, the binding pool nodes are distributed representations.

The active binding pool nodes represent the binding between the token and types and no longer require activation from the tokens or types to remain active. Now, the types can be used to encode the next object, if multiple objects are presented.

After all of the objects have been encoded, the only activity present in the model is activity in the binding pool. Next, information can be retrieved from the model. To do this, the model simulates retrieval in two stages. In the first step, a cue is used to retrieve the token associated with that cue (Figure 6). In this example, the

location of the presented stimulus is used to retrieve its color. To do this, the cue activates the corresponding feature value in the type layer and then projects activity to the binding pool. Next, the subset of binding pool nodes connected to the active type nodes projects to the token layer. This results in a noisy reconstruction of the activity in the token layer and is formalized as,

$$Z_t = \sum_{\beta=1}^{\omega} B_{\beta} N_{t,\beta} \sum_{f=1}^{m_{cue}} \sum_{g=1}^{n_{m_{cue}}} X_{f,g} L_{f,g,\beta} \quad [5]$$

where m_{cue} corresponds to the type layers that are acting as a cue and ω is the total number of binding pool nodes.

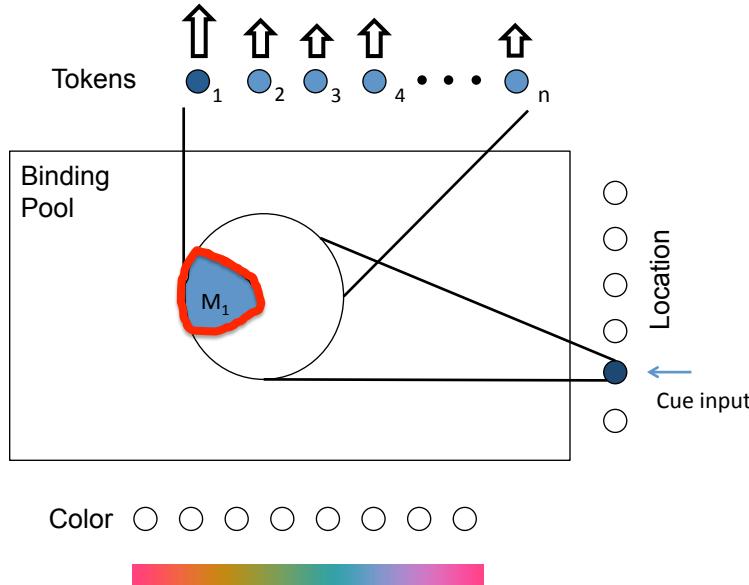


Figure 6: Illustration of the first stage of retrieval, which corresponds to the retrieval of a token. In this example, the cue activates the corresponding location feature in the type layer. These activated type nodes then project to the binding pool, activating a subset of these nodes. Those activated binding pool nodes (i.e. the nodes surrounded in red) are then used to activate the token layer. If a token is sufficiently more active than the next most active token, then that token is retrieved.

A successful token retrieval occurs when the most active token is sufficiently more active than the next most active token, determined by the token individuation parameter, ς . Most likely with set size 1, the activity in the binding pool results in the

correct token being retrieved, but occasionally, there will be a retrieval error whereby the wrong token is retrieved⁸. In other situations, if the model cannot sufficiently select a token, then the model simply selects a random feature value (i.e. guessing⁹). In this example, the model would simply randomly select a color value from [1, 360].

After the cue has been used to retrieve a token, then the model uses both the token and the cue to retrieve the desired type layer (i.e. color in this example) (Figure 7). To do this, the token projects to the binding pool. The subset of binding pool nodes connected to the token projections and the projections from the cue's active type nodes will then be used to reconstruct the desired type layer, which is formulized as,

$$X_{tar,f} = Z_t \sum_{\beta=1}^{\omega} B_{\beta} L_{tar,\beta} N_{t,\beta} \sum_{f=1}^{m_{cue}} \sum_{g=1}^{n_{m_{cue}}} X_{f,g} L_{f,g,\beta} \quad [6]$$

where X_{tar} corresponds to the type dimension that is being retrieved.

⁸ When the wrong token is retrieved, the model makes a retrieval error almost every time. More specifically, in simulation 10,000 trials of set size 4, the model failed to retrieve a token on 28% of trials and retrieved an incorrect token on 16% of trials. This is nearly the same proportion of trials that the mixture model estimates for guesses (.25) and for swaps (.19). When only using the trials in which the correct token is retrieved, the mixture model estimates swaps on .015 of trials and no guesses. When only using the trials in which a non-target token is retrieved, the mixture model estimates only swaps. Interestingly, the precision is 28 degrees compared to 24.5 degrees, which is the precision when looking at precision only when a correct token is retrieved. And lastly, as expected, trials where the model could not retrieve a token resulted in a uniform distribution (i.e. guess).

⁹ There is an active debate in the field whether low precision responses (i.e. responses far from the presented color value) are actual true guesses (Zhang & Luck, 2008) or very coarse memories (van den Berg et al., 2012). Knowing the difference would be useful in distinguishing between slot and resource models, but there is no clear distinction with regards to guessing yet (e.g. see van den Berg & Ma, 2014 for discussion of how plateau's in summary statistics for precision curves are not useful to distinguish slots and resources). Therefore, the Binding Pool model simply opts for true guessing.

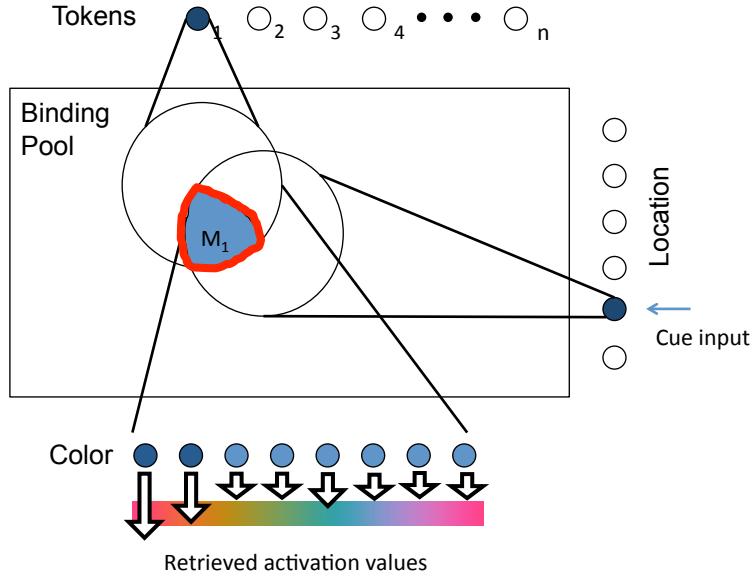


Figure 7: Illustration of the second stage of retrieval, which corresponds to the retrieval of a type. Here, a token has been activated and is projecting activity to the binding pool. The subset of nodes in the binding pool connected to both the active token and the type layer (i.e. the nodes surrounded in red) acting as a cue is then projected to the feature that the model is trying to retrieve (e.g. color). The activation that is retrieved is a noisy reconstruction of what was presented to the model.

The reconstructed activity in the retrieved type layer is noisy and is decoded into a continuous value by first treating the noisy reconstruction as a set of vectors where the angle of these vectors are their corresponding positions (Figure 8). These vectors are then converted into Cartesian space. Next, the vectors are summed, and the resultant mean vector has both a direction (i.e. an angle) and a length. The direction is taken as the retrieved color and the length is assumed to represent an internal metric of confidence in the retrieved value.

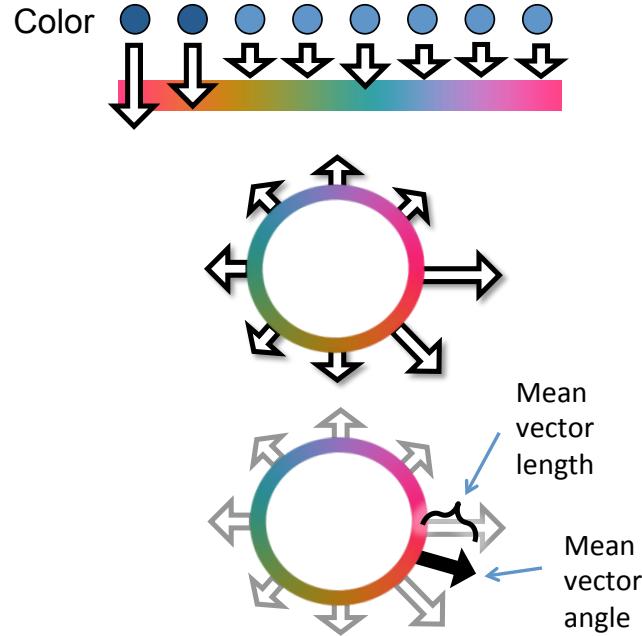


Figure 8: Illustration of the how the noisy reconstruction of the type layer after the second stage of retrieval is decoded. The activation values are treated as vectors along a continuous space. The retrieved feature value is the angle of the mean of these vectors. In addition to the mean of the vectors, there is also a length of the mean vector, which is interpreted as the confidence of the retrieval.

Encoding Multiple Objects

When multiple objects needs to be encoded, the model encodes each object serially until all objects are encoded, or the maximum number of tokens on this trial is reached. Otherwise, encoding occurs in the same way for each additional object. Importantly, the distributed nature of the binding pool means that binding pool nodes may overlap between multiple objects (Figure 9). This overlap between multiple object results in interference during memory retrieval, which could manifest as decreased precision and increases in retrieval errors and guesses.

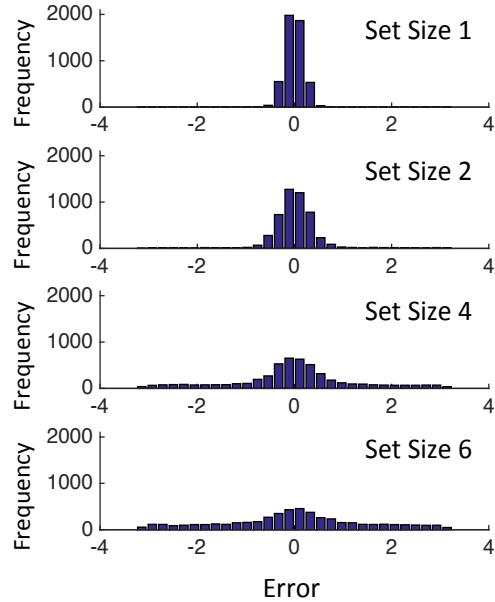


Figure 9: Response distributions from simulations of the Binding Pool model when the set size is 1, 2, 4, and 6 in 5000 trials per set size.

Parameter Fitting

To determine a single set of parameter values for the model, the parameter values were selected that minimized the root mean square error between two data sets simultaneously. In the first data set, estimates of precision, guessing, and retrieval errors in a delayed estimation task with set sizes 1, 2, 4, and 6 from Bays et al., (2009) were fit. In the second data set, the model was fit to hit and false alarm rate in change detection data from Keshvari et al., (2013) where the set size varied from 2, 4, 6, and 8. Additionally, the model was fit to the analysis by change magnitude, which illustrates the probability of detection as a function of set size and the difference between the study and test object that changed (Keshvari et al., 2013). When searching for the optimal parameters, a grid-search over the parameters in Table 1 was used.

There are 11 free parameters in the model (Table 1). Of the 11 free parameters, 4 parameters are specific to simulations of change detection. For all simulations described in this thesis, the parameters are fixed to the values described in Table 1, unless otherwise noted.

Table 1. Parameters of the Binding Pool model taken from Swan and Wyble (2014)

Parameter	Description	Value
Binding Pool Size (ω)	The amount of available nodes for storing links; as the size increases, so does the representational fidelity of stored information.	800
Type Layer Size (ρ)	The amount of type nodes in each feature space	10
Connection Sparsity (α)	The proportion of active connections between the each layer and the binding pool	0.45
Type Connection Overlap (L)	The proportion of added overlap between neighboring type nodes	0.30
Encoding Capacity [Ψ_ω]	The number of stimuli encoded per trial is drawn for a uniform distribution	[2 7]
Token individuation threshold (ζ)	If the ratio of the maximum retrieved token activity to the average retrieved token activity does not exceed this threshold, then the model will initiate 'guess' by choosing an unbound token.	0.016
Length threshold for change detection (κ_s and λ_s)	Two parameters (baseline and setslope) define the thresholds for detecting change based on the length of the retrieved vector (i.e. confidence).	$\Phi_L = 0.005 * \text{Setsize} + 0.09$
Deviation Threshold for change detection (κ_d and λ_d)	Two parameters define the thresholds for detecting change based on the deviation of the retrieved value from the probe value.	$\Phi_d = 0.038 * \text{Setsize} + 0.16$

Conclusions

The Binding Pool is a computational simulation of how visual features are combined to form a memory representation. This model utilizes a type-token mechanism that provides a means for representing specific features of a stimulus, while binding those features together to form an object using tokens. This

information is maintained in a distributed resource pool called the binding pool. The Binding Pool model has the ability to simulate a variety of empirical findings in the visual working memory literature, such as precision decrements in delayed estimation and trade-offs between hits and false alarms in change detection (see Swan & Wyble, 2014 for more details). Using the model, three predictions of the model about human memory were tested and are described in this thesis.

Chapter 3. The Influence of Similarity on Memory Precision

In many everyday situations, it is important to remember multiple pieces of information. Knowing how multiple objects in memory may or may not interact with each other has important implications when multiple objects need to be remembered with high degrees of accuracy, such as driving a car in high traffic situations. Early slot models of working memory (e.g. Luck & Vogel, 1997) theorize that objects are independently maintained, such that there would be little to no interference between representations when storing multiple objects. However, recent research looking at visual working memory (VWM) recall using delayed estimation have found trade-offs between the quantity of objects that need to be maintained and the quality of retrieval (Wilken & Ma, 2004; Bays et al., 2009). There are even trade-offs within a single presentation of stimuli (Brady et al., 2015). Brady et al., (2015) presented the same memory displays to 300 participants on Amazon's Mechanical Turk and found consistently different levels of precisions for different displays of colors. One possible cause that is driving these effects is the similarity between presented objects. Similarity, in this case, is the difference between multiple objects along a single feature dimension (e.g. in color space, red is more similar to orange than blue). The experiments in this chapter focuses on characterizes how similarity affects memory precision.

In some recent VWM studies (Lin & Luck, 2009; Sims, Jacobs, & Knill, 2012), more similarity between objects increased the probability of detecting a change relative to non-similar objects in change detection tasks. One explanation is that more similar objects are easily compressed into more efficient representations (Brady, Konkle, & Alvarez, 2009; Brady & Alvarez, 2011). The Binding Pool model (Swan & Wyble, 2014) predicts that this increase in accuracy is due to increased overlap between the representations, which means that the representations pull, or bias, retrieval towards each other. The model predicts that this biasing will affect

memory precision systematically, such that memory precision is highest when the two objects are highly similar (Figure 10).

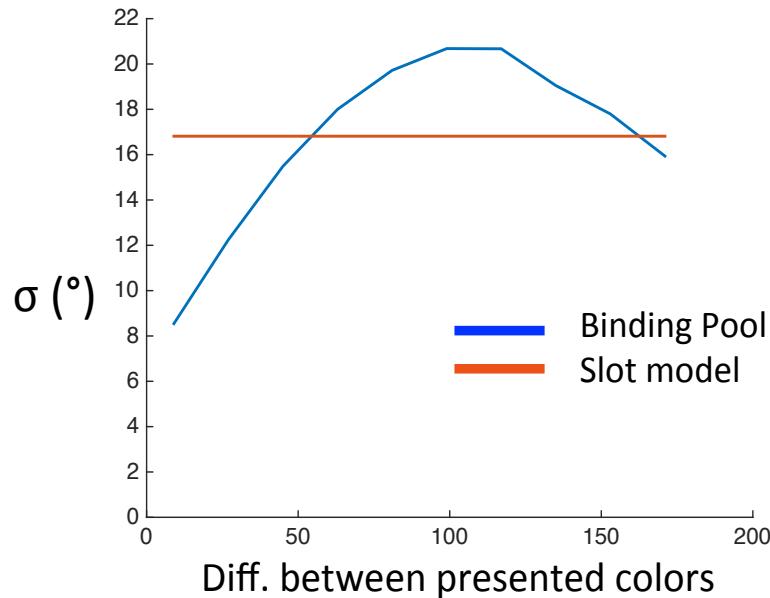


Figure 10: Prediction about how the similarity between two objects affects memory precision. Here, the Binding Pool model predicts an n-shaped function. On the other hand, models that keep representations in memory independent (e.g. slot model) may predict that similarity has no effect on response precision. The Binding Pool simulation had 3000 trials for differences in the absolutely value between 2 presented colors from 18 to 180 degrees in 18 degree increments.

Huang and Sekuler (2010) found evidence in favor of this prediction in a delayed estimation task where participants reported the spatial frequency of one out of two sequentially presented Gabor patches. They found that participants recalled a spatial frequency that is biased towards the spatial frequency of the non-reported Gabor patch. The Binding Pool model predicts that this bias would be present in other feature dimensions, such as color, as well.

In addition to predicting increased precision for highly similar stimuli, the model predicts that participants will also be precise for highly dissimilar stimuli compared to intermediate levels of similarity (Figure 10). This benefit for dissimilar

stimuli also results from the method of decoding retrieved activity in the model. That is, when the target and non-target are highly dissimilar, there is less biasing of the mean vector compared to when the objects have an intermediate level of similarity. The exact point at which this occurs depends upon the retrieval strength of the objects (see Changes to the Model section for more details).

To test this prediction, participants were presented with two color patches and precision was analyzed according to the similarity between the two color patches. If there were no relationship between precision and the similarity of the two colors, then it would suggest that there are limited interactions between information in memory. It may be the case that only some features interact, such as spatial frequency shown by Huang and Sekuler (2010). Though, given other results (e.g. Lin & Luck, 2009; Oberauer & Lin, 2017), this seems very unlikely. The Binding Pool model, alternatively, predicts an ‘n’ shaped function, such that precision is highest when the two colors are most similar, next highest when the colors are most dissimilar, and least precise at intermediate levels of similarity.

Experiment 1

Participants: Seventy participants were recruited from the online subject pool at Pennsylvania State University. Ten participants were omitted from analyses, described below. All participants were presented instructions in American English and signed an informed consent approved by the Pennsylvania State University IRB. Participants received course credit for their participation.

Apparatus: The experiment was run using Matlab 7.9.0 (build R2009b) with PsychtoolBox (Brainard, 1997; Pelli, 1997) scripts on computers running Windows XP (Microsoft, Redmond, WA). The screen resolution was set to 1024x768 at a 75-Hz refresh rate on cathode ray tube monitors with a diagonal screen size of 40cm. Participants were seated in a chair with an adjustable height and situated in chin rests located 50 cm from the monitor.

Stimuli: The stimuli consisted of color patches that could vary in color across trials. Color was selected from a series of 360 colors in a one-dimensional selection from CIE L*a*b* color space where L was fixed at 54 and a circle was drawn with a radius of 50.5 and origin centered at a = 22.5 and b = 11.5. The color patch had dimensions 3.6 by 3.6 of visual angle. Color patches appeared equidistant along an invisible circle with radius 4.9 degrees of visual angle.

Perceptual matching: Participants first completed 10 trials of a perceptual matching task. In this task, two color patches were presented along the horizontal meridian separated by 10.8 degrees of visual angle. Participants were instructed to match colors between the two stimuli by selecting a color value from a surrounding color wheel. Across trials, the rightmost color patch had a color that was uniformly sampled from 10 possible color values separated by 36 degrees to cover the extent of the wheel. When participants moved their mouse to the color wheel, the leftmost color patch changed to the color value of the location of mouse cursor. Participants were free to move the mouse until satisfied with their answer, which they could record by clicking the left mouse button. Note that this method of responding was used in the delayed estimation task. After responding, the next trial began 500ms later.

Delayed estimation: After completing the perceptual matching task, participants began the experimental trials (Figure 11). Participants were instructed to remember the color of two presented color patches. After 250ms of fixation time, two color patches appeared for 150ms. After the offset of the color patches, there was a retention interval for 900ms. Following the retention interval, participants were presented with the response screen. On the response screen, there was a color wheel made up of the 360 colors and a white box. A location cue (i.e. a white box) informs participants of the relevant color. Responses were made by moving the mouse cursor to the color wheel similar to responses in the perceptual matching task. After selecting a color, participants reported their confidence. Three white bars appeared; one centered at the location of their response and two equally spaced around the color wheel. When participants moved their mouse along the central white bar, the two outside bars moved along the color wheel (Figure 11).

Participants were instructed to create a range whereby the true value is a color inside of the two bars approximately 90% of the time. Note that the confidence metric data was collected for a separate set of analyses.

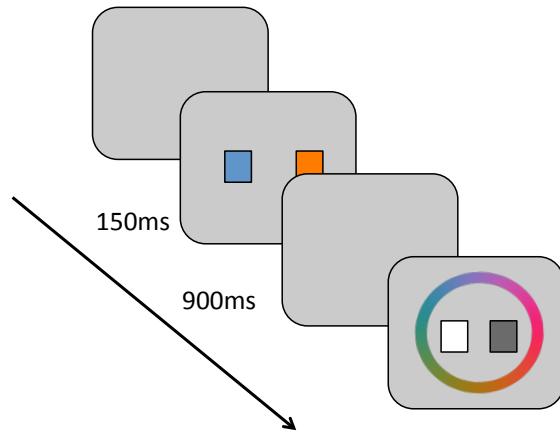


Figure 11: Illustration of the procedure of Experiment 1. Participants were asked to remember both presented colors. After the colors disappear, there was a short retention interval, followed by the recall screen. The location of the white box was the cue to which color needed to be recalled. Recall occurred by clicking on the color that best matches what was shown.

Participants ran the experiment for 26 minutes. Participants completed on average 139 trials (standard deviation = 42 trials). Note that the amount of trials does not significantly predict precision ($r = .16$, $p = .22$). Participants were not given a fixed number of trials in order to maximize the number of trials that could be completed within the available time slot (i.e. less than 30 minutes).

Results and Discussion

The dependent measure of interest was the difference between the reported color and the presented color of the cued color patch, which resulted in an error value in degrees. Over many trials, the errors created a response distribution. It is assumed that this response distribution is composed of multiple types of responses,

such as guesses and retrieval errors (Bays et al., 2009). To factor out these responses, the response distribution per participant was analyzed using a mixture model. This mixture model was composed of two circular Gaussian distributions called von Mises and a uniform component, which allowed for the precision of the memory response, guess rate, and swap rate to be extracted from the response distribution (see Chapter 1 for more details).

First, the data was fit to the mixture model to determine if any participants needed to be removed for having outlier parameter fits. Of the 70 participants, 2 were removed for having precision 2 standard deviations below the mean, 3 were removed for having swap rates 2 standard deviations above the mean, and 5 were removed for having guess rates 2 standard deviations above the mean. The resulting average parameter fits for the remaining 60 participants are displayed in Figure 12.

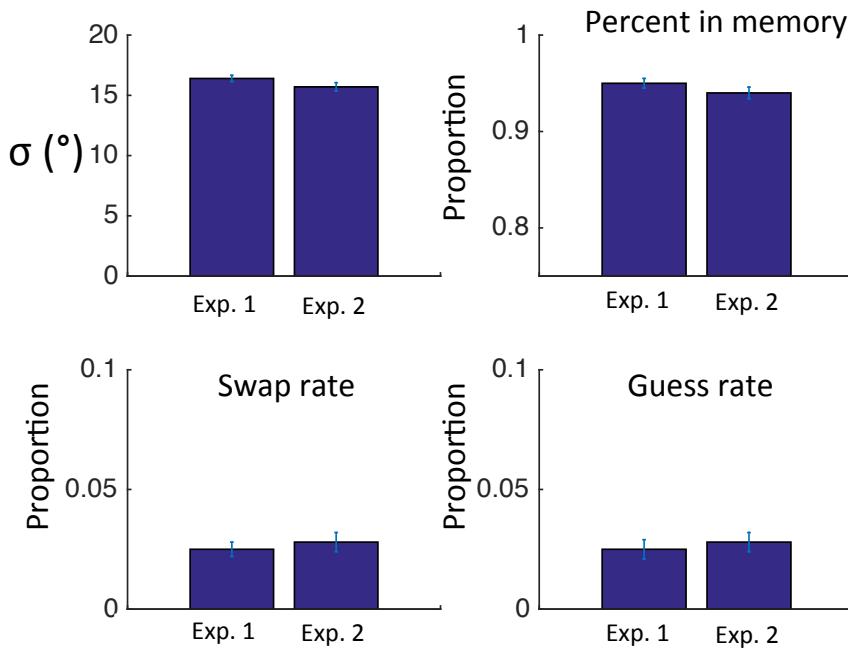
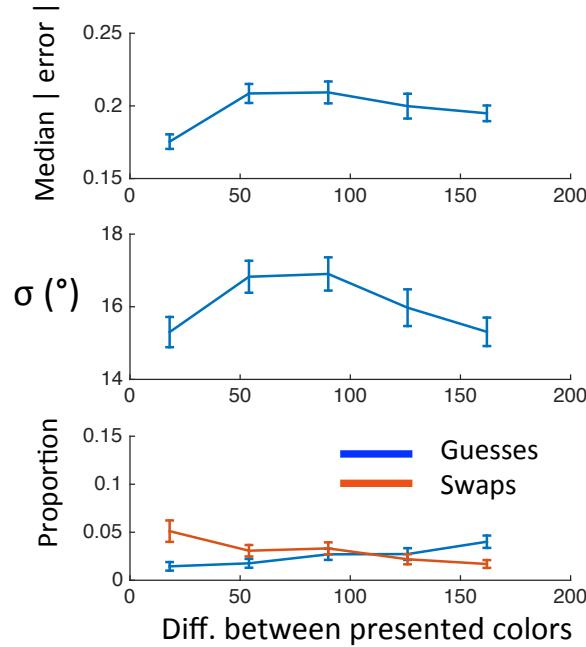


Figure 12: Parameter estimates from a mixture model (Bays et al., 2009) for Experiment 1 and Experiment 2. Error bars correspond to standard error.

In the following analyses, the focus was on the relationship between the similarity of the two colors and the precision of the response. The data were binned according to the absolute value of the difference between the two presented colors.

The bins went from 0-36, 37-72, 73-108, 109-144, and 145-180 degrees. These bins were selected to maximize the number of trials per bin when analyzing data across participants (mean = 28 trials per bin per participant). For each participant, the median absolute value of the error was calculated per bin (Figure 13a). In addition, the mixture model was also fit to the data within each bin¹⁰ (Figure 13b,c).



To determine whether there was a difference between any two bins at a time, paired t-tests were conducted on precision between two bins. First, the first bin was

¹⁰ It is worth noting that one may think that there may be parameter inflation when mixture modeling these data within the different bins. For example, given that the presented values are so proximal in the first bin, it may be difficult for the mixture model to decode swaps and guesses from the target distribution. However, when generating fake data, applying a similar binning process, and then putting those data through the mixture model, there is no observable parameter inflation. It is worth noting that in these generated fake data, that the median error of the simulated data has a positive slope (i.e. the median error increases as the two objects become less similar), which one would expect given that any simulated swaps in bins after the first bin would pull the median away from central distribution.

compared to the second bin to determine if participants were significantly more precise when the two colors were most similar. In a paired t-test, this was significant [$t(59) = 2.8, p < .01$], suggesting participants became less precise when the stimuli became less similar. This was also true when comparing the first bin to the third bin [$t(59)= 2.5, p < .02$]. However, the first bin was not significantly different from the fourth [$t(59)= 1.1, p = .29$] or fifth bin [$t(59)=.01, p = .99$]. In order to determine if participants improved precision when the two stimuli became more dissimilar, the fifth bin was compared to the second [$t(59)=2.8, p < .01$], third [$t(59)=2.8, p < .01$], and the fourth [$t(59)=1.2, p = .22$] bins.

In addition to the above analyses, a secondary analysis was conducted to determine if participants were improving their precision as a function of increasing dissimilarity between the presented colors. To do this, log likelihoods (LL) were compared between model fits where the slope could vary for precision, guess rate, and swap rate for the last 4 bins and a model where there is no slope between those parameters. When comparing between these two models, the LL of the slope model was -4241.5 and the LL of the no-slope model was -4259.1. An LL difference of 17.6 is significant even when considering the additional three free parameters in the slope model.

The above results replicate the findings for increased VWM capacity (Lin & Luck, 2009) and increased precision (Oberauer & Lin, 2017) when multiple objects are similar in feature space. In addition, these results show that highly dissimilar feature values had higher precision compared to immediate levels of similarity. To determine whether these results replicate, a second experiment was conducted where the stimuli were presented sequentially. By using a sequential presentation, the experiment better matches the methodology used by Huang and Sekuler (2010) and may minimize potential ensemble effects that may be present with simultaneous presentation.

Experiment 2

Participants – Seventy new participants were recruited from the online subject pool at Pennsylvania State University. Ten participants were omitted from analyses, described below. All participants were presented instructions in American English and signed an informed consent approved by the Pennsylvania State University IRB.

Stimuli and procedure – The stimuli and procedure were the same as Experiment 1 except where follows (Figure 14). After a 250ms prestimulus fixation, the first stimulus appeared for 150ms. After the offset of the first stimulus, there was a 500ms inter-stimulus interval. The second stimulus appeared for 150ms and was followed by a 900ms retention interval. The stimuli, like Experiment 1,

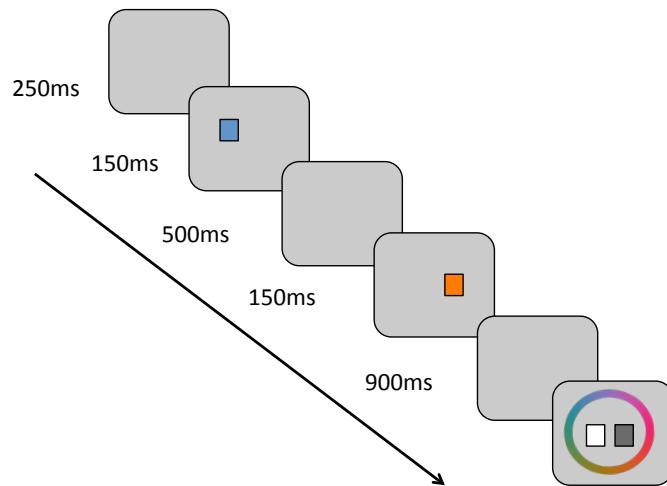


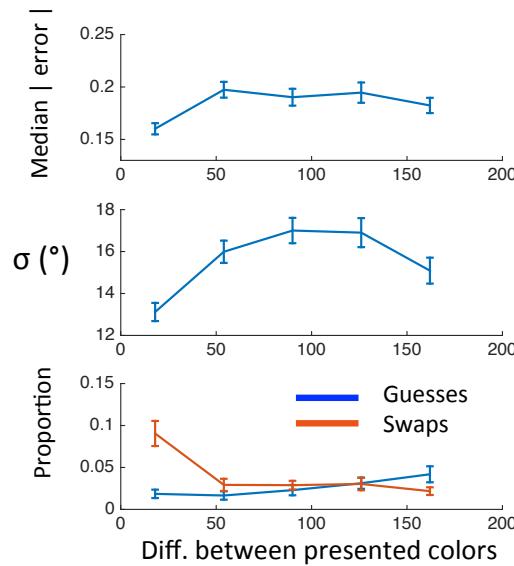
Figure 14. Illustration of procedure for Experiment 2. Note that the primary difference between Experiment 1 and Experiment 2 is that in Experiment 2, the stimuli are displayed sequentially.

appeared on an invisible circle, so location was still the feature used to cue participants as to which color needs to be recalled. The response screen was the same as the probe screen used in Experiment 1.

Results

Similar to Experiment 1, the data was first fit to the mixture model (Bays et al., 2009) to determine if any participants would need to be excluded. Of the 70 participants, 2 were removed for having precision below 2 standard deviations from the mean, 2 for guess rates 2 standard deviations above the mean, and 6 for swap rates 2 standard deviations above the mean. The remaining 60 participants' parameter estimates are displayed in Figure 12.

The same bins used in Experiment 1 were also used for Experiment 2 (Figure 15). To compare between different bins, paired t-tests were run for precision between any two bins. Like Experiment 1, bin 1 was significantly different from bin 2 [$t(59)=5.0, p < .001$], bin 3 [$t(59)=6.1, p < .001$], bin 4 [$t(59)=5.0, p < .001$], and, unlike Experiment 1, bin 5 [$t(59)=3.5, p < .001$]. These results generally replicate Experiment 1, insomuch that precision is improved for similarly colored objects. When comparing the last bin where the two color values were most dissimilar, bin 5 was not significantly different from bin 2 [$t(59)=1.2, p = .24$], but was significantly different from bin 3 [$t(59)=2.5, p < .02$] and bin 4 [$t(59)=2.6, p < .02$].



In order to determine if participants were incrementally becoming more precise as a function of increasing dissimilarity, the analysis used in Experiment 1 was also used here. When comparing the model where the slope was a free parameter ($LL = -2635.5$) to the model where the parameters had no slope ($LL = -$

2630.5), the difference in LL (LL difference = 5) was not significant when taking into account the additional free parameters in the more complex model.

Post-hoc Analyses

It is possible the above results may only be present for one of the serial positions. For example, it may be the case that the positional effects only arise for responses to one of the serial positions and that may be driving the differences in the similarity analyses. First, the effects of serial position were examined without taking into account similarity to see if there was a serial position effect on precision, swaps, and guess rate. When separating all of the trials by serial position, there was a significant difference in a paired t-test between SD depending on whether the first (mean: 16, SEM: .4) or second (mean: 15.1, SEM: .4) color was reported [$t(59)=2.4$, $p = .02$]. However, serial position on guess rate [$t(59)=1.1$, $p = .29$] or swap rate [$t(59)=1.1$, $p = .26$] was not significant. This benefit in precision for the most recent object replicates the finding from other VWM using sequential presentation (Gorgoraptis et al., 2011).

Next, the similarity effect was examined for the different serial positions. Figure 16 shows the relationship between precision and the difference between the two presented colors as a function of serial position. Note that this analysis was run collapsed over participants, given that halving the number of trials per participant would result in not enough trials for accurate parameter estimation (Lawrence, 2010). Based on visual inspection of Figure 16a, it appears as if serial position did not have an effect on the overall shape of the precision curve, given that both are n-shaped.

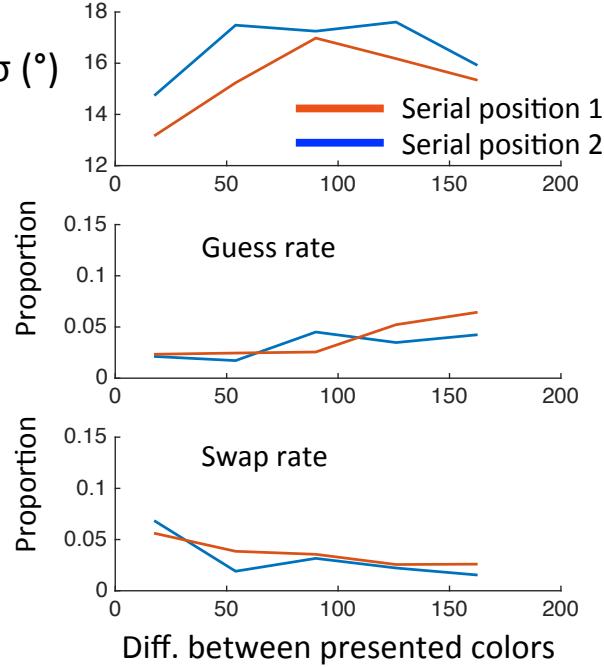


Figure 16: Serial position effects separated for Experiment 2. In the top figure, precision is displayed for serial position 1 (orange) and serial position 2 (blue). In the middle figure, the guess rate is displayed. In the bottom figure, the swap rate is displayed.

General Discussion

Two experiments were conducted to see how the similarity in color space between two objects affects memory precision. In both experiments, there was a benefit in precision when the two colors were similar (within 36 degrees of each other) compared to intermediate levels of similarity. This result replicates previous findings from the literature (Lin & Luck, 2009; Sim & Jacobs, 2012; Oberauer & Lin, 2017). One possible explanation is that when objects are similar, they are easily compressed into a single memory representation (Brady, Konkle, & Alvarez, 2009; Brady & Tenenbaum, 2013). This compressed representation may mean more resources are contributing to the individual representation, which enables better precision. This finding of increased precision for similar objects is consistent with the Binding Pool and other models of VWM (e.g. Oberauer & Lin, 2017). However, it

is important to note that theories where interference of storing multiple objects arises from overlapping sensory representations would have predicted worse precision for more similar objects (e.g. Franconeri et al., 2013).

In addition to finding increased precision for highly similar colors, there was also increased precision for highly dissimilar colors (above 144 degrees difference) in both experiments compared to intermediate levels of similarity. One possible explanation is that sufficiently dissimilar colors interfere less in memory. In this particular experiment, the colors on the opposite side of the color wheel are approximately opponent colors given that the color wheel is selected near the origin of L*a*b* color space where L was fixed. This may be the case, given that memory for colors is not uniformly precise (e.g. Bae, Ollkonen, Allred, Wilson, & Flombaum, 2014). Whether the benefits found in this experiment are simply because of the opponent colors would need to be explored in future experiment. For example, if participants needed to recall orientation, would the same benefits for highly dissimilar orientations be found? The Binding Pool model would predict increased precision regardless of the feature dimension (see section below).

It is worth noting that it appears as if participants were more precise in the most similar bin range in Experiment 2 compared to Experiment 1. When comparing precision across experiments with post-hoc independent t-tests, there was a significant difference [$t(118)=3.4, p < .001$] for the first bin. Whereas, for the last bin, there was not a significant difference [$t(118)=.3, p = .76$]. This suggests that participants were able to benefit from the similarity effect more in Experiment 2. It is likely this benefit was driven primarily by responses to the stimulus in serial position 2 (see Figure 16). It is likely that participants were able to use additional strategies to boost their precision. For example, when the two colors were highly similar, the color in serial position 1 was recoded as being relative to the color in serial position 2 (e.g. more blue or more red), resulting in a response more similar to a set-size 1 experiment.

Oberauer and Lin (2017) reported similar findings from an experiment where the set size was manipulated from 1 to 8. Similar benefits for similar colors were found when comparing the mean absolute value of the error to the mean

deviation in color space between the presented colors, but their analysis focused on the mean of the absolute value of the difference. The mean is influenced by swaps and guesses and these types of responses bias the mean towards higher deviations. The current experiments find the same increase in mean absolute value of the error as a function of bin¹¹. It is thus difficult to determine if the same benefit for dissimilar colors was present in their results across multiple set sizes.

Changes To The Model

The data from the experiments qualitatively support the predictions from the Binding Pool model. However, the magnitude of the n-shaped function found in the participant data was less than the predicted function from the Binding Pool model. In addition, given that the effect seemed to be diminished in Experiment 2 though still present, it is important to clarify the cause of the Binding Pool model's prediction regarding the effect of similarity on precision..

Type Activation Decoding

The way that the type activations are decoded into a single color response is playing the primary role in determining the effect that similarity has on performance. To test this, two vectors were placed along a circle. The vectors were moved apart along the circle and as the separation increased, the angle of the mean vector was recorded (Figure 17).

¹¹ The mean of the absolute value of the error has a similar profile to the median of the absolute value of the error, except that the mean of the absolute value of the error is linear across the similarity axis instead of n-shaped like the median.

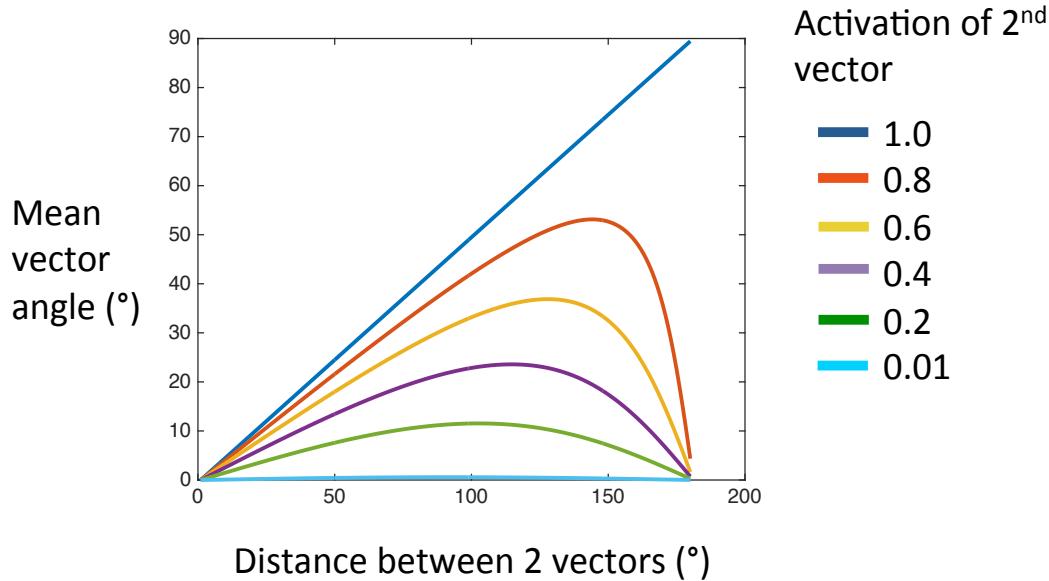


Figure 17: Results of moving one vector around a circle while another vector (activation = 1) is stationary and then taking the mean vector at each degree separation between the two vectors. The different colors correspond to different activations of the moving vector

This is similar to the reconstructed type dimension when presented with two objects¹². That is, type nodes that coded for the presented objects are more often more activated at retrieval than type nodes that were not activated during encoding. Figure 18 displays retrieved type activations as a function of whether they were activated during encoding as the target, non-target, or not activated at all for set size 2.

From these tests, the prediction of the model is driven primarily by the method of decoding the retrieved type activations. Therefore, one way to manipulate the similarity curve would be to adjust the relative proportions of activity in the retrieved type layer. For example, if the activity is raised by an exponent less than 1, then similarity's effect on precision is increased, and the effect

¹² The illustration in Figure 17 demonstrates how 2 vectors of different lengths pull a mean vector. It is important to note that in the Binding Pool model, there are 10 total vectors (i.e. the number of type nodes, determined by parameter ρ), and that while a set size of 2 often results in 2 vectors that are larger than the other vectors (Figure 18), this is still a simplification of the retrieved activity.

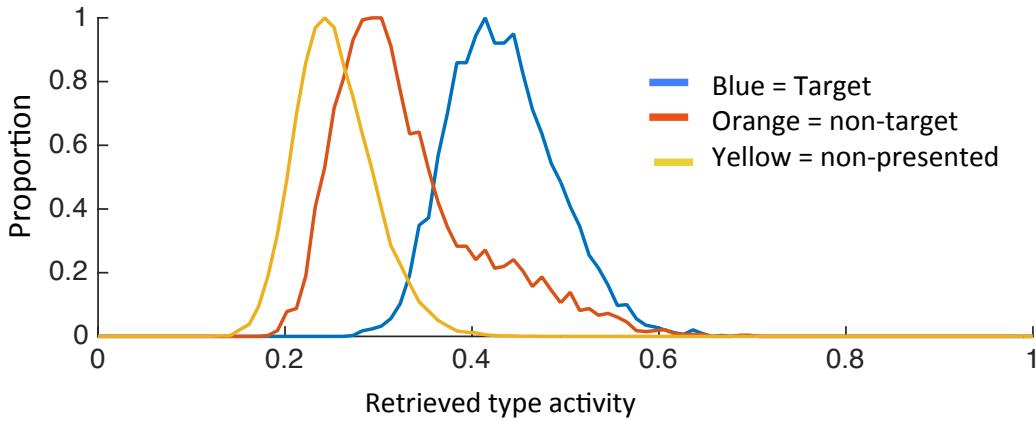


Figure 18. Type activity at retrieval separated by whether that type node was initially a target (Blue), non-target (Orange), or not-presented (Yellow) during encoding. Note that for display purposes, values at 0 (i.e. when the model guesses) were removed (approximately 3% of trials in this simulation of 5000 trials).

is decreased if the exponent is greater than 1. However, given that the results from Experiments 1 and 2 qualitatively match the prediction from the Binding Pool, the model does not need to undergo any adjustments.

Conclusion

In two experiments, participants were presented with two color patches either simultaneously (Experiment 1) or sequential (Experiment 2). When analyzing the data with respect to the similarity of the presented colors, an n-shaped distribution was found for both experiments. Participants were most precise when the two colors were highly similar or highly dissimilar compared to intermediate levels of similarity. One explanation is that similar colors recruit similar memory resources that may benefit precision, whereas dissimilar colors may have less interference. This result supports predictions from the Binding Pool model.

Chapter 4. The Storage of Task Irrelevant Features for Multiple Objects

Much of our understanding of visual working memory (VWM) is built on how well participants remember specific features of presented objects. For example, in most delayed estimation paradigms, participants are asked to remember a set of briefly presented colors and then later report one of the presented colors. Yet, it is necessarily the case that there are features irrelevant to the task that may be encoded, such as the shape or texture of a color patch when its color is the relevant feature. To fully understand memory capacity, it is important to know exactly what information is encoded about an attended object. Some researchers theorize that all features of an attended object are encoded into memory (e.g. Luck & Vogel, 1997), whereas others theorize that only features relevant to the task are encoded (e.g. Awh, Vogel, & Oh, 2006). A third possibility (Figure 19c) is that task irrelevant features are encoded at varying levels of coarseness (Swan, Collins, & Wyble, 2016). Knowing which of the above theories is more accurate has important implications about the underlying mechanisms of the working memory and its limited capacity. This topic has been explored in two ways. First, is there a cost to remembering more features of an object? If there is no cost, then it suggests that all features are encoded, whereas a cost suggests differences in the encoding of relevant and irrelevant features. And secondly, is the task irrelevant feature of an object being encoded? If it is not reportable, then it suggests that all features are not encoded into memory.

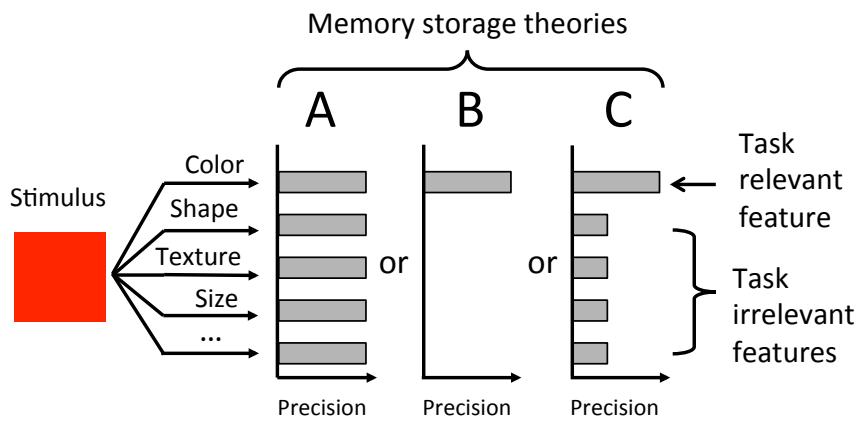


Figure 19: Figure taken from Swan, Collins, and Wyble (2016). In that article, the researchers found evidence in favor of theory C, which is that task relevant features are coded precisely while task irrelevant features were coarsely coded.

The first question regarding whether there is a cost to storing more features of an object has been explored previously with change detection. In typical change detection experiments, participants are presented with an array of objects (e.g. colored oriented lines) and must detect whether any objects in the first array have changed in a second array that appears after a brief retention interval. Luck and Vogel (1997) and Vogel et al. (Experiments 11 -16, 2001) compared hit rate between conditions where participants needed to remember the conjunction of features (e.g. color and orientation) to conditions where only a single feature was relevant (e.g. color or orientation) and found no effect of adding features. These results suggest that even in the single feature condition, participants were encoding irrelevant features because their hit rate did not change when the other features were task relevant. However, attempts to replicate these findings were mixed (e.g. Wheeler et al., 2002; Olson & Jiang, 2002), other data suggesting there is a cost to storing more complex stimuli (Alvarez & Cavanagh, 2004; Oberauer & Eichenberger, 2013; c.f. Awh, Barton, & Vogel, 2007), and other experiments showing no correlation between the ability to report two features of the same object (Bays, Wu, & Husain, 2011; Fougnie & Alvarez, 2011). For example, Oberauer and Eichenberger (2013) used more complex feature conjunctions (e.g. shape, size, color, orientation, thickness) and found that adding more features with a constant set size resulted in decrements in hit rate. These results from Oberauer and Eichenberger (2013) support a prediction from the Binding Pool model that adding features to memory

results in decrements in precision while holding the number of objects constant. This prediction arises from the fact that adding more features reduces the overall number of binding pool nodes representing a memory and less binding pool nodes means less precise retrieval. However, the incongruences in the literature make it difficult to conclude whether the Binding Pool model's prediction is supported or not. Determining whether participants can report the task irrelevant feature may elucidate some of these conflicting results.

With regards to the storage of task irrelevant features, there is some evidence of the storage of task irrelevant features using neurological imaging techniques, which have found brain activity in response to irrelevant features (Xu, 2010; Gao, et al., 2009). However, without explicitly measuring whether participants can report an irrelevant feature, one cannot conclusively say whether that feature was actually encoded into memory. For example, the brain activity may simply reflect perception of those task irrelevant features, but not the storage of those features into memory. The difficulty in explicitly measuring a memory for an irrelevant feature is that that feature becomes relevant to the task once it is reported. Therefore, there is only a single critical trial per participant¹³. Despite this difficulty, a number of recent publications using a surprise test methodology to explicitly probe an irrelevant feature have come to differing conclusions. For example, Eitam et al. (2015) found that participants could recognize an irrelevant shape of an attended object in a surprise forced choice recognition task. However, Chen and Wyble (2015) found that participants could not report the color of an attended letter or the identity of that letter, despite the identity being necessary to complete the task.

¹³ There are alternative methodologies that may increase the number of 'surprise trials'. For example, it may be possible to ask a participant to report a task irrelevant feature (e.g. color). Then, consistently ask them to report a different feature (e.g. orientation) over a large set of trials, which may reset the relevancy of the initially task irrelevant feature. Another potential methodology is to surprisingly ask participants to report different irrelevant features on different trials. However, participants may adapt different strategies in anticipation that they may be asked about a feature that was intended to be task irrelevant.

However, one of the limitations of these studies and others (Eitam, Yeshurun, & Hassan, 2013; Chen, Swan, & Wyble, 2016; Swan, Wyble, & Chen, 2017) is that a recognition task was used to probe the memory for the task irrelevant feature. Recently, it was shown when combining a delayed estimation task with a surprise test methodology that participants could recall an irrelevant feature of a single object, albeit with less precision than when that same feature was task relevant (Swan, Collins, & Wyble, 2016; Shin & Ma, 2016). These results could corroborate the findings from the recognition tasks described above. For example, participants may have seen the color red, but remembered orange. If orange was not a possible choice in a recognition task and red was a choice, participants may select red over alternative colors that were not similar to red. Thus, finding that participants could or could not report an irrelevant feature may depend upon the choices in a recognition task..

One potential limitation of Swan, Collins, and Wyble (2016) is that only a single stimulus was used. It may be the case that there are special properties when only a single stimulus is used. For example, Oberauer and Lin (2017; see Oberauer 2002 as well) posit that there is a focus of attention that selectively increases that object's precision in memory. Thus, it seems necessary to explore the generalizability of the findings from Swan, Collins, and Wyble by determining if task irrelevant features are also coarsely coded when there are multiple objects. This is tested using a similar methodology to Swan, Collins, and Wyble (2016), where one feature (e.g. color) is initially relevant and the irrelevant feature (e.g. direction) is reported in a surprise test. In addition to explicitly measuring precision of the irrelevant feature with a surprise test, the precision of the relevant feature may change following the surprise trial, which would demonstrate that the irrelevant feature was not being encoded while it was irrelevant. The proposed experiment tests a prediction of the Binding Pool model that encoding more features comes at a cost to memory precision, regardless of the number of objects presented.

Experiment 1

Participants: 130 participants were recruited from the subject pool at Pennsylvania State University. 30 participants were omitted from analyses, described below. All participants were presented instructions in American English and signed an informed consent approved by the Pennsylvania State University IRB. Participants received course credit for their participation.

Apparatus: The experiment was run using Matlab 7.9.0 (build R2009b) with PsychtoolBox (Brainard, 1997; Pelli, 1997) scripts on computers running Windows XP (Microsoft, Redmond, WA). The screen resolution was set to 1024x768 at a 75-Hz refresh rate on cathode ray tube monitors with a diagonal screen size of 40cm. Participants were seated in a chair with an adjustable height and situated in chin rests located 50 cm from the monitor.

Stimuli: The stimuli consisted of a colored arrow that could vary in color and direction across trials. Color was selected from a series of 360 colors in a one-dimensional selection from CIE L*a*b* color space where L was fixed at 54 and a circle was drawn with a radius of 50.5 and origin centered at a = 22.5 and b = 11.5. The arrow was constructed using a rectangle with the dimensions 5.4 and 1.2 visual angle and an isosceles triangle with a 1.7 degree base and 1.2 degree height. Direction was selected from 360 degrees as indicated by the arrowhead. The colored arrows appeared along the horizontal meridian separated by 4.4 of visual angle.

Perceptual matching: Participants first completed 20 trials of a perceptual matching task. The perceptual matching task trained participants on how to produce responses when asked to report color, and importantly, direction in the delayed estimation task. This block of trials was labeled Experiment 1 and participants were instructed to be as precise as possible. In this task, there were two blocks randomly allocated depending on the parity of the participant number. In the color matching block, two colored arrows were presented along the horizontal meridian separated by 10.8 degrees of visual angle. Participants were instructed to

match colors between the two stimuli by selecting a color value from a surrounding color wheel. Across trials, the rightmost color patch had a color that was uniformly sampled from 10 possible color values separated by 36 degrees to cover the extent of the wheel. When participants moved their mouse to the color wheel, the leftmost color patch changed to the color value of the location of the mouse cursor.

Participants were free to move the mouse until satisfied with their answer, which they could record by clicking the left mouse button. Note that this method of responding was used in the delayed estimation task. After responding, the next trial began 500ms later. In the direction matching block, the presented stimuli were the same, except that the color wheel was gray and the rightmost arrow varied by 36 degrees across trials. When participants moved their mouse to the outside wheel, the arrowhead of the leftmost arrow rotated to the angle of the cursor.

Delayed estimation: In the second block, participants were instructed that the researchers were interested in how precise their memories could be and that only a single feature is relevant (e.g. color). Participants were presented with two colored arrows with an independently sampled color and direction from 360 possible values. At the start of every trial, there was a 300ms prestimulus fixation screen. The arrows then appeared for 150 ms, followed by a brief mask of colored oriented lines for 100ms. After a 1000ms retention interval, participants were presented with the response screen. For the first 26 trials, participants were presented with a color wheel and consistently reported color. Responses occurred in the same manner as in the perceptual matching. On the 27th trial, participants were presented with a gray wheel, a gray square in the location of which direction needed to be retrieved, and a brief message that informed participants that this was a surprise question and that direction needed to be retrieved. Like the responses in the perceptual matching task, once participants moved their mouse cursor to the gray wheel, the arrow (which appeared in place of the gray square when the cursor exceeded the radius of the wheel) moved to the angle of the mouse cursor. The response wheel was centered as the location of one of the presented stimuli. For the remaining trials (i.e. 28 to 79), participants were randomly asked about either the color or the orientation of the one of the arrows.

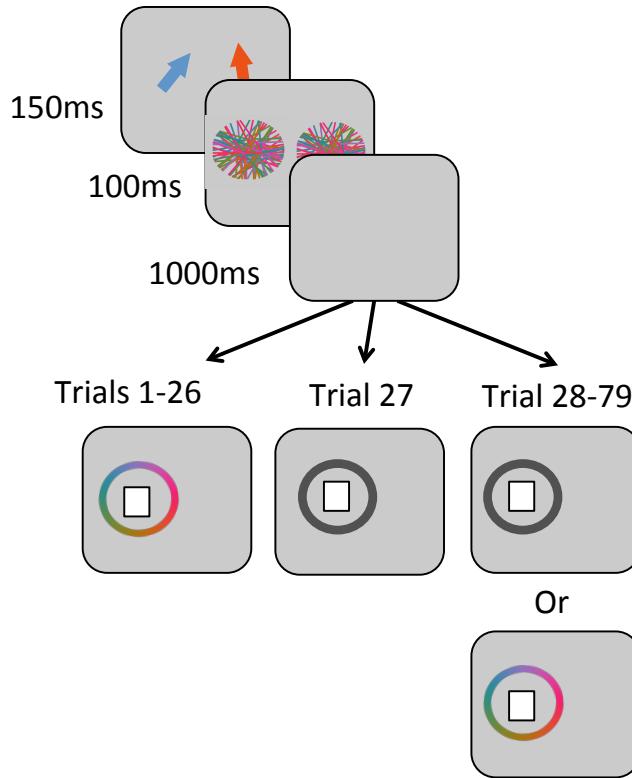


Figure 20: Illustration of the procedure of the task. Participants were asked to remember the color of the presented arrows. After the arrows disappear, there was a mask, a short retention interval, then the recall screen. The location of the white box is the cue to which color needs to be recalled. Recall occurred by clicking on the color that best matches what was shown. On trial 27, there was a surprise question that asked participants to recall the direction of one of the arrows (cued in the same way as before). Following trial 27, participants were asked about either the color or orientation of the different arrows.

Results and Discussion

Two steps were used to determine if participants needed to be excluded. This was the same set of exclusion criteria in Swan, Collins, and Wyble (2016). First, if participants had an error (i.e. the difference between the reported and presented angle) greater than 10 degrees in any of the last 3 trials of the perceptual matching task for the direction feature, they were excluded. This is an important criterion for exclusion because it minimizes the potential that inaccuracy on the surprise trial resulted from an inability to know how to respond. Of the 130 participants, 11

participants had poor performance on the last 3 trials of the perceptual matching trial. Secondly, 9 participants were excluded for having precision that was two standard deviations above the mean and 10 participants for having Pm (percent in memory) two standard deviations below the mean. These parameters were generated by first calculating the error of the trial type. It is assumed that the error is composed of multiple types of responses, such as guesses and retrieval errors (Bays et al., 2009). This mixture model is composed of two circular Gaussian distributions called von Mises and a uniform component, which allows for the precision of the memory response, guess rate, and swap rate to be extracted from the error distribution (see Chapter 1 for more details). The error distributions and the parameter fits of the remaining subjects are shown in Figures 21 and 22, respectively.

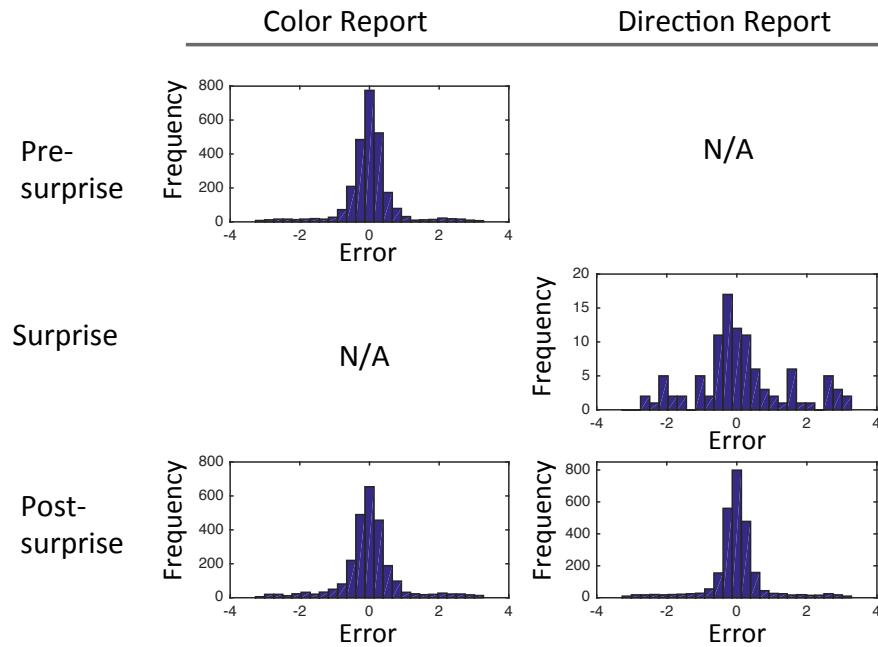


Figure 21. Error distributions for different conditions of Experiment 1. The first column corresponds to responses to the color of the colored arrows. The second column corresponds to responses to direction of the colored arrows. Note that in pre-surprise trials, participants only reported color and in the surprise trial, participant reported the direction.

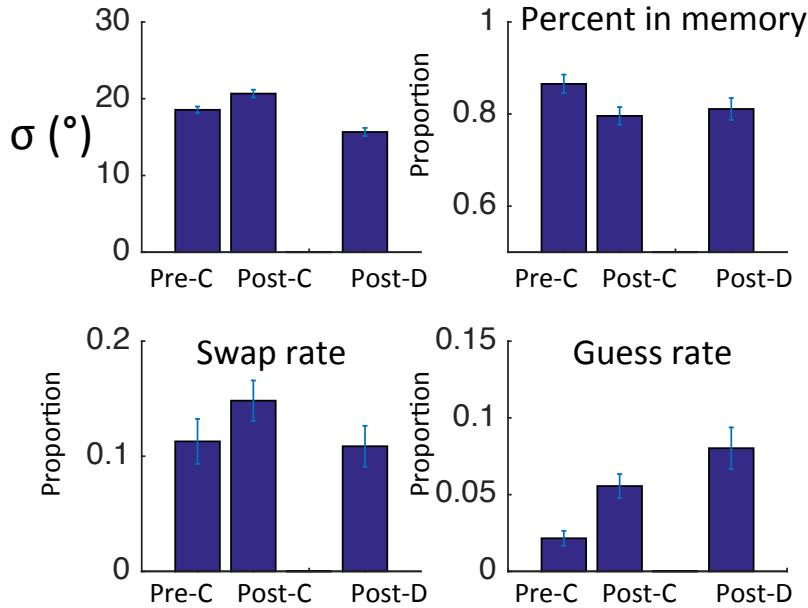


Figure 22. Parameter estimates across participants for Experiment 1. Error bars are standard error. Pre-C, Post-C, and Post-d refer to pre-surprise color, post-surprise color, and post-surprise direction trials, respectively.

First, the difference between the pre-surprise and post-surprise trials was compared to determine what effect making direction relevant in post-surprise trials had on color responses. For each participant, precision, guess rate, and swap rate were estimated for pre-surprise and post-surprise trials where color was reported by passing those trials into the mixture model (Bays et al., 2009). To see if there was a cost between the two conditions, the fitted parameters between pre and post surprise were compared with paired t-tests. There was a significant difference between pre-surprise and post-surprise trials for precision [$t(99) = 3.5, p < .001$], guess rate [$t(99) = 4.2, p < .001$], and swap rate [$t(99) = 3.4, p < .001$]. These results suggest that adding an additional relevant feature not only resulted in a decrement in memory precision, but also an increase in guesses and retrieval errors. This replicated the results from Swan, Collins, and Wyble (2016) and suggests that coarse coding of task irrelevant features was not limited to cases in which only a single object needs to be remembered.

One possibility in the decrement between pre-surprise and post-surprise trials is that participants were gradually getting worse on the task from a factor unrelated to the addition of a feature to memory load, such as fatigue. To test this hypothesis, the data were collapsed across trial (Figure 23). If it were the case that participants were getting worse as the experiment continued due to fatigue, a model (Model 1) where parameter estimates varied linearly across trials would fit the data better than a model (Model 2) where the parameter estimates do not vary (i.e. no slope in parameters). Alternatively, participants could be getting worse as a function of direction becoming task relevant, which would be a model (Model 3) with a set of parameter estimates for pre-surprise and post-surprise parameters. When these different models were fit to the data, Model 3 had the higher AIC value (-4111.2) than Model 2 (4141.5) and Model 1 (-4119). Given that Models 3 and Model 1 have the same number of free parameters (Model 1 = slope and y-intercept for SD, guess rate, and swap rat; Model 3 = 2 separate y-intercepts for SD, guess rate, and swap rate), an AIC difference of 7.8 significantly favors Model 3. This indicates that participants were not simply getting worse across the experiment and there participants had worse precision and increased guess and swap rates when direction became task relevant in post-surprise trials.

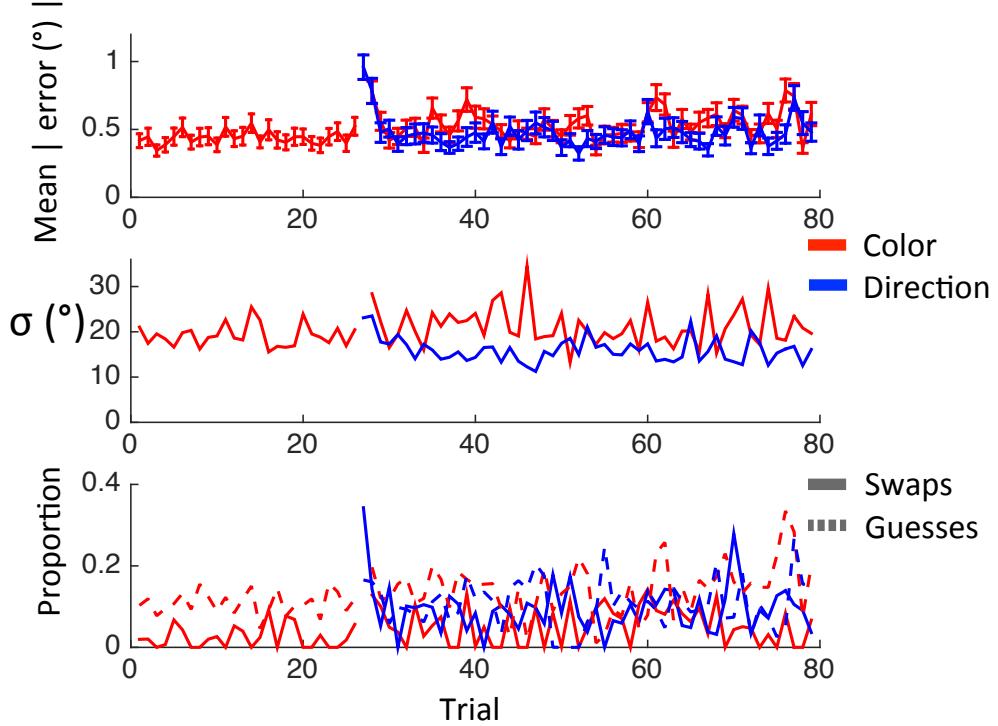


Figure 23. The effects of the trial and type of response are displayed here for Experiment 1. The top figure is the absolute value of the mean across trial. The error bars are standard error. In the middle figure, the precision across participants for different trials is displayed. In the bottom figure, the guess rate (solid lines) and swap rate (dotted lines) are depicted.

Next, direction responses were analyzed. Participants had a post-surprise direction precision of 15.7 degrees (SEM: .53), guess rate of .08 (SEM: .01), and swap rate of .11 (SEM: .02). To determine how precisely participants were encoding direction when direction was task irrelevant, the surprise trial data were analyzed next. The errors on the surprise trial were analyzed using the mixture model across participant (see Swan, Collins, & Wyble 2016 for details), which resulted in a precision of 23.1 degrees, a guess rate of .34, and a swap rate of .17. To determine how responses for a task irrelevant feature differ from when that same feature was relevant, the surprise trial fits were compared to the first post surprise trial response of the same feature dimension. The resulting parameter fits of the first post-surprise trial for direction were a precision of 20.9, a guess rate of .12, and a swap rate of .13. By comparing the surprise trial to the first post-surprise trial per

participant, the comparison was most conservative and useful for determining how quickly participants were able to adjust to new task requirements.

To determine if there was a difference in the model fits between the surprise and first-post surprise trials, a permutation test over differences in log likelihood (LL) was performed. First, the difference between the LL of the best fitting model to the surprise data was subtracted from the LL of the surprise data calculated using the parameters of the best fitting model for the first post-surprise trial. A difference of 12.6 LL was found and to determine whether this difference is significant, a permutation test was conducted. To produce a null distribution, the data between the two trials was permuted 1,000 times. For each permutation, the same LL difference was computed. A comparison of the observed difference and the empirically derived differences from the null distribution revealed that less than 1% of the permuted differences were larger than the observed difference. This same analysis was run in the reverse order, where first post surprise trial data was fit to the parameters best fit to the surprise trial data, and the same results were found (LL difference = 10.6, $p < .02$)

These results suggest that participants were coarsely coding the task irrelevant feature and that making that feature relevant had a cost on the precision of the already relevant feature. These results replicate the findings from Swan, Collins, and Wyble (2016) and support the prediction of the Binding Pool model that adding features reduces memory precision. In addition, these results help demonstrate that coarsely coding task irrelevant features may be a general property of memory and not a function of there being just a single object. Similarly, these results suggest that participants can individuate visual features across multiple objects. However, set size 2 was likely still below memory capacity. Would these same findings be present when the amount of information exceeds what is typically considered above capacity? In Experiment 2, the set size was increased to 4 objects.

Experiment 2

Participants: 125 participants were recruited from the subject pool at Pennsylvania State University. None of these participants had previously participated in Experiment 1. 25 participants were omitted from analyses, described below. All participants were presented instructions in American English and signed an informed consent approved by the Pennsylvania State University IRB.

Procedure: The stimuli and apparatus are the same except where noted. In the delayed estimation task, participants were presented with 4 arrows for 400 ms. The stimulus duration was increased to provide participants with enough time to encode all of the available objects. The arrows were located on the edges of an invisible stationary box with dimensions 4.4 by 4.4 degrees of visual angle. Like in Experiment 1, the location of the response wheel and the appearance of the response wheel cued the desired feature of one of the objects. The surprise trial started on trial 29 and the total number of trials was increased to 85 trials. Thus, there were 28 pre-surprise trials and 56 post-surprise trials (28 for color, 28 for direction).

Results

The participant exclusion criteria for Experiment 2 were the same as the criteria for Experiment 1. First, 9 participants were excluded for having poor performance on the perceptual matching. Secondly, 7 participants were excluded for having precision 2 standard deviations above the mean, 9 participants for percent in memory (P_m) 2 standard deviations below the mean. Error distributions and parameter fits for the resulting 100 participants are displayed in Figures 24 and 25 respectively.

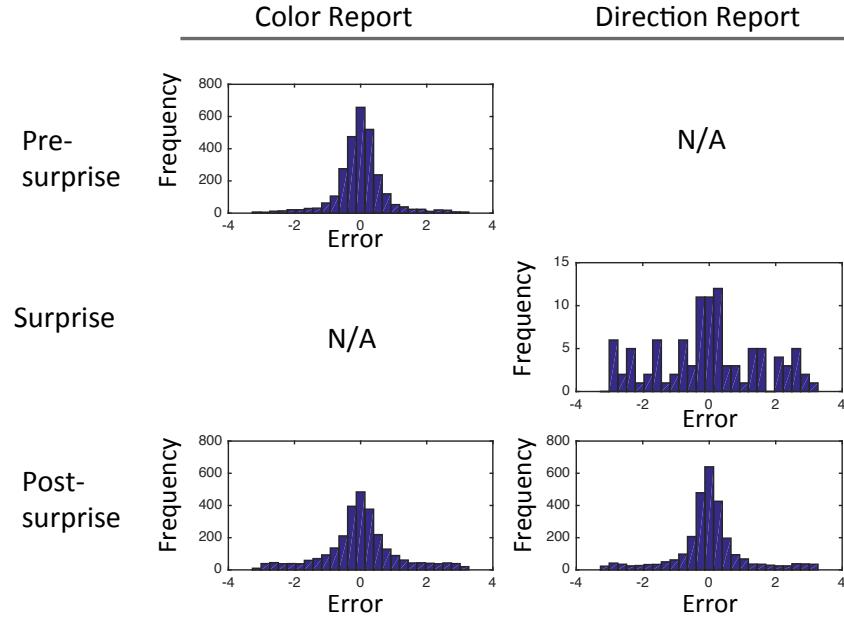


Figure 24. Error distributions for different conditions of Experiment 2. The first column corresponds to responses to the color of the colored arrows. The second column corresponds to responses to direction of the colored arrows. Note that in pre-surprise trials, participants are only reported color and in the surprise trial, participant report the direction

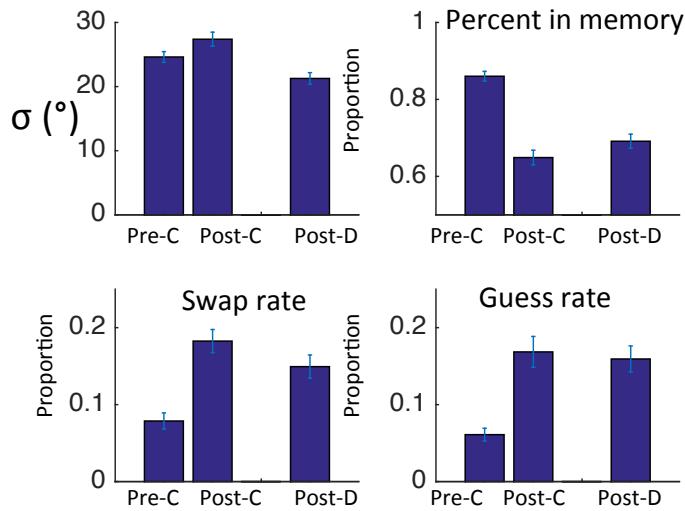


Figure 25. Parameter estimates across participant for Experiment 2. Error bars are standard error. Pre-C, Post-C, and Post-d refer to pre-surprise color, post-surprise color, and post-surprise direction trials, respectively

When comparing the pre-surprise and post-surprise trial parameter estimates, there was a significant difference a paired t-test for precision [$t(99) = 2.1$, $p < .04$], guess rate [$t(99) = 5.2$, $p < .001$], and swap rate [$t(99)= 6.4$, $p < .001$]. These results show a trade-off between the number of relevant features and memory quality, with greater effects on guess and swap rate than precision. However, it is worth noting that 28 trials may not be an adequate amount of trials for the mixing model fitting for response distributions with high variance (Lawrence, 2010), which may be the case here given that a set size of 4 potentially exceeds memory capacity.

In addition to comparing pre and post-surprise trials with paired t-tests, the across trial performance was also examined (Figure 26). Three models were fit to the across trial performance: Model 1 has parameter estimates varying across trial, Model 2 has a single parameter estimate across all trials (i.e. no slope), and Model 3 has a single set of parameter estimates for pre-surprise trials and a separate set for post-surprise trials. Model 3 fit the data the better ($LL = -6556.9$) than Model 2 ($LL = -6687.1$) and Model 1 (-6580.5). An AIC difference of 23.6 is significantly different even when accounting for the extra free parameters and thereby favors Model 3 over the other models. This analysis shows that it was unlikely that the trade-off between the number of relevant features and precision, guess, and swap rate was a result of participants gradually getting worse as the experiment continued.

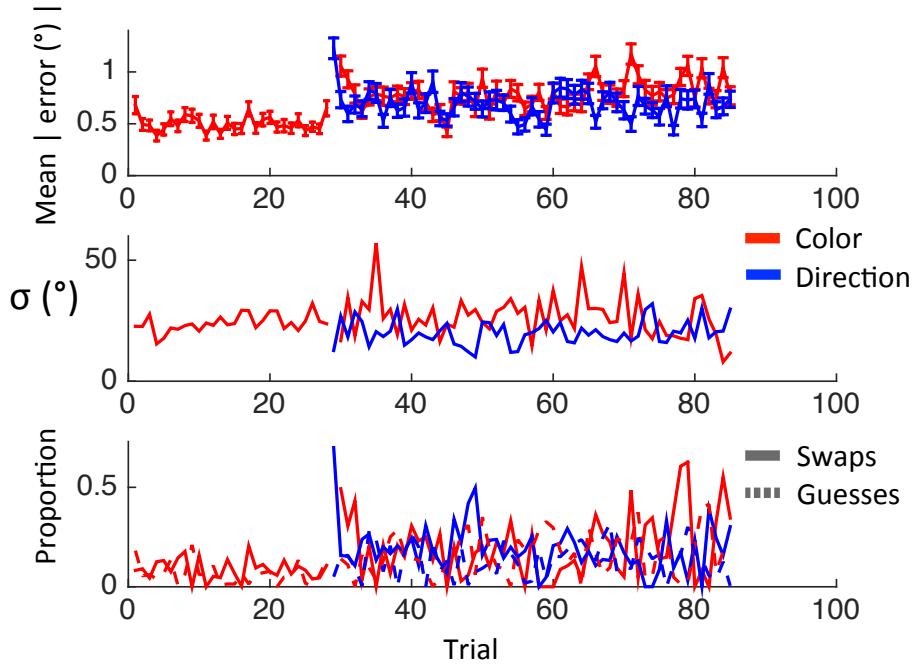


Figure 26. The effects of the trial and type of response are displayed here for Experiment 2. The top figure is the absolute value of the mean across trial. In the middle figure, the precision across participant for different trials is displayed. In the bottom figure, the guess rate (solid lines) and swap rate (dotted lines) are

Next, responses to direction were analyzed. Participants had a post-surprise direction precision of 21.2 (SEM: .9), guess rate of .16 (SEM: .017), and swap rate of .15 (SEM: .015). The mixture model was then fit to the surprise trial, which revealed parameter estimates unlike those found previously; some participants were highly precise (precision = 12.7 degrees), while most participants guessed (guess rate = .70) and few swapped (swap rate = .05). In Experiment 1 and in Swan, Collins, and Wyble (2016), participants responded coarsely on the surprise trial while improving almost immediately on post-surprise trials.

On the first post-surprise direction trial, participants' precision decreased to 29.3, but increased to 26.9 and 20.7 degrees for the second and third post-surprise direction trial. Percent in memory remained relatively more stable (.68, .78, and .63), insomuch compared to the average post-surprise trial direction percent in memory (.69, SEM: .018). In the likelihood analysis (described in Experiment 1),

there was a significant difference between the fits for the surprise trial compared to the first post-surprise surprise trial ($p < .001$) and vice versa ($p < .04$).

General Discussion

In two experiments, participants were unexpectedly asked to report a feature of an attended object. One goal of these experiments was to determine the memory quality of a task irrelevant feature when multiple objects are presented. An additional goal was to determine whether there was a cost to the initially task relevant feature when the initially task irrelevant feature becomes relevant to the task. These experiments are a continuation of Swan, Collins, and Wyble (2016), which used a single object, by exploring how increased set sizes affect memory precision for task irrelevant features.

In both experiments, participants were instructed to remember the color of multiple colored arrows and were surprisingly asked about the direction of one of the arrows half way through the experiment. There were 2 colored arrows presented in Experiment 1 and 4 colored arrows in Experiment 2. To determine the cost of adding a feature to the memory load, responses before the surprise trial (i.e. one relevant feature) were compared to responses following the surprise trial (i.e. two relevant features). When the task irrelevant feature became relevant to the task, there was a cost to precision for color when comparing pre-surprise trials to post-surprise color trials. This result replicates the findings from Swan, Collins, and Wyble (2016) and suggests that even when multiple objects are presented, there is a trade-off between the number of features being remembered and precision.

These results from Experiment 1 and Experiment 2 support the prediction of the Binding Pool model that adding features to memory decreases memory precision. This prediction arises from the way information is encoded in the model. When adding features to a memory representation, fewer binding pool nodes represent an object given that the active binding pool nodes for an object is the conjunction of all of the types and the token projections to the binding pool.

Therefore, with less binding pool nodes contributing to the representation, retrieval precision declines (Figure 4)

To determine how well participants were encoding the task irrelevant feature, the response on the surprise trial was analyzed. Responses on the surprise trial were coarser than responses to that same feature when that feature was relevant to the task in trials following the surprise trial, which again replicates the finding from Swan, Collins, and Wyble (2016). The results of the surprise trial for Experiment 2 had a different pattern of data than Experiment 1. Some participants were more precise on the surprise trial than the first post-surprise trials for direction and a large subset of participants guessed. The increased precision on the surprise trial is contradictory to the results in Swan, Collins, and Wyble and Experiment 1. There are several possibilities that may explain the differences in responses on the surprise trial between the experiments. First, it is possible that for some subset of participants, the four presented arrows formed a pattern (e.g. a square) meaningful to the participants on the surprise trial. This Gestalt may have 'popped out' to participants, resulting in the incidental encoding of the arrows' directions.

Secondly, it is possible that some participants were using different encoding strategies in the delayed estimation task. For example, both Fougner et al., (2016) and Donkin et al., (2016) found that participants could increase the number of reported objects (i.e. Pm) in lieu of precision. Some participants may have prioritized a couple of the arrows, resulting in more precise encoding of the arrows' features. However, this would suggest that those participants would have significantly more guesses in pre-surprise trials if those participants were ignoring some of the objects. When looking at the guess rate for participants whose response on the surprise trial was likely a response to the target¹⁴, it seems unlikely that

¹⁴ The likelihood of a response being a correct target, a swap, or a guess was divided by the sum of the likelihoods. This calculation provides results in probabilities that a given trial comes from either the target, uniform, or non-target distribution (Schneegans & Bays, 2016). For simplicity, the maximum probability was selected as the type of response for that participant. First, it is important to keep in mind that these are probabilities, so simply taking the max is leaving room a significant

these participants were guessing more than participants who had a less accurate response (difference of .002). However, it could be the case that these strategies were not being used at the beginning of the experiment and appeared at different trials for different participants in the pre-surprise trials.

Thirdly, it is also possible that there are not enough trials to accurately estimate the parameters of a response distribution with such high variance. For example, in 1000 simulations of 100 trials where 25% of the trials are Von Mises distributions with a precision of 27 degrees (i.e. intentionally coarser than 12.7 degrees), 70% of trials are guesses, and 5% of the trials are swaps, approximately half of the simulations resulted in a mixture model fit with a precision below 27 degrees and 7.3% of the simulations has a precision less than 12.7 degrees (i.e. the precision on the surprise trial for Experiment 2). This simple test suggests that it is possible by chance (though unlikely) to result in a high precision fit of random responses drawn from a coarse distribution with only 100 trials.

A fourth potential explanation is that participants simply had more time to precisely encode the task irrelevant feature. The stimulus duration was 150ms in Experiment 1 and in Swan, Collins, and Wyble and 400ms in Experiment 2. It is possible that with more time, participants' precision for task irrelevant features may improve similarly to how precision improves¹⁵ for relevant features with increased stimulus durations (Bays, Gorgoraptis, Wee, Marshall, & Husain, 2011). Though, 150ms used in Swan, Collins, and Wyble for a single object should have been enough to encode both the color and direction, presuming that features require only 50ms to be encoded (Vogel, Woodman, & Luck, 2006). Future experiments will need to be

amount of contamination between distributions. Secondly, one limitation of simply taking the max is that errors may be closely proportional for multiple types of responses. When looking at the data, only 6 responses on the surprise trial had probabilities that were ambiguous (i.e. a range of less than 20% between the 2 highest probabilities).

¹⁵ Precision does improve with increased stimulus durations, but precision does not infinitely improve (Bays et al., 2011). Bays et al., found that precision roughly plateaus around 350ms stimulus duration for different set sizes. Importantly though, the precision plateaus at different levels across set sizes, such that 1000ms presentation of four objects is still less precise than 1000ms of two objects.

conducted to determine how stimulus duration impacts the encoding of task irrelevant features.

In the trials following the surprise trial, participants improved dramatically in their ability to report direction. This result was found in Swan, Collins, and Wyble, and suggests that participants can rapidly allocate the precision with which features are encoded based on task demands. These task demands arise from changes in task relevance, given that in post-surprise trials, participants could be asked to report either the color or the direction of one of the arrows. However, given that participants were able to report the task irrelevant feature (albeit coarsely) on the surprise trial in Swan, Collins, and Wyble, Experiment 1, and in the first-post surprise trial of Experiment 2, it is not simply the case that an extra feature was added in the post-surprise trials. Instead, the Binding Pool model would need to be able to simulate trade-offs in precision and the number of relevant features while coarsely coding the task irrelevant features. To do this, task relevance needs to be added to the Binding Pool model.

Changes To The Model

In order for the model to simulate Experiment 1 described above, the ability to simulate task relevance needs to be added to the model. Relevance is implemented as featural attention in the model.

Adding Featural Attention

One way to implement featural attention is by changing the amount of activation in the type layer. It has previously been shown that attention to a feature changes the shape of the tuning curve of neurons corresponding to that feature. This research using monkey neurophysiology finds that the amplitude of the tuning curve changes as function of the amount of attention directed towards that feature (McAdams, & Maunsell, 1999). Implementing this in the model is quite simple.

Currently, activation is interpolated between two type nodes when that type dimension represents a continuous feature,

$$type_{floor}\left(\frac{Stimulus}{v}\right) = 1 - \frac{Stimulus}{v} - \left(floor\left(\frac{Stimulus}{v}\right) \right) \quad [7]$$

$$type_{ceil}\left(\frac{Stimulus}{v}\right) = \frac{Stimulus}{v} - \left(floor\left(\frac{Stimulus}{v}\right) \right) \quad [8]$$

where *floor* and *ceil* correspond to the type node that is rounded down and rounded up from the feature value, respectively. Stimulus refers to the feature value selected for the stimulus. The sum of the interpolation between two type nodes is equal to 1, but this value could be changed by,

$$type_{floor}\left(\frac{Stimulus}{v}\right) * \lambda \quad [9]$$

$$type_{ceil}\left(\frac{Stimulus}{v}\right) * \lambda \quad [10]$$

where λ is a weight parameter that corresponds to the amount of attention to a feature dimension. Full attention to a feature is implemented as $\lambda = 1$ and less attention is described as $\lambda < 1$.

To simulate Experiment 1, precision with two relevant features and one irrelevant feature was compared to precision with three relevant features.

Interestingly, the difference in task relevance (i.e. attention to a feature) on a third feature had no effect on the model's precision (Yellow; Figure 27¹⁶). This simulation

¹⁶ In this simulation, there were 100 participants generated from the model (i.e. matching the sample from Experiment 1) who each had a unique connectivity

makes sense. Regardless of a feature's relevancy, the same number of binding pool nodes is being activated. Thus, all that is being affected is the total amount of activation in the binding pool, but not the proportion of activity between different features. The model will need to undergo another change in order for feature relevance to affect memory precision and a feasible mechanism to change is how information is encoded into the binding pool.

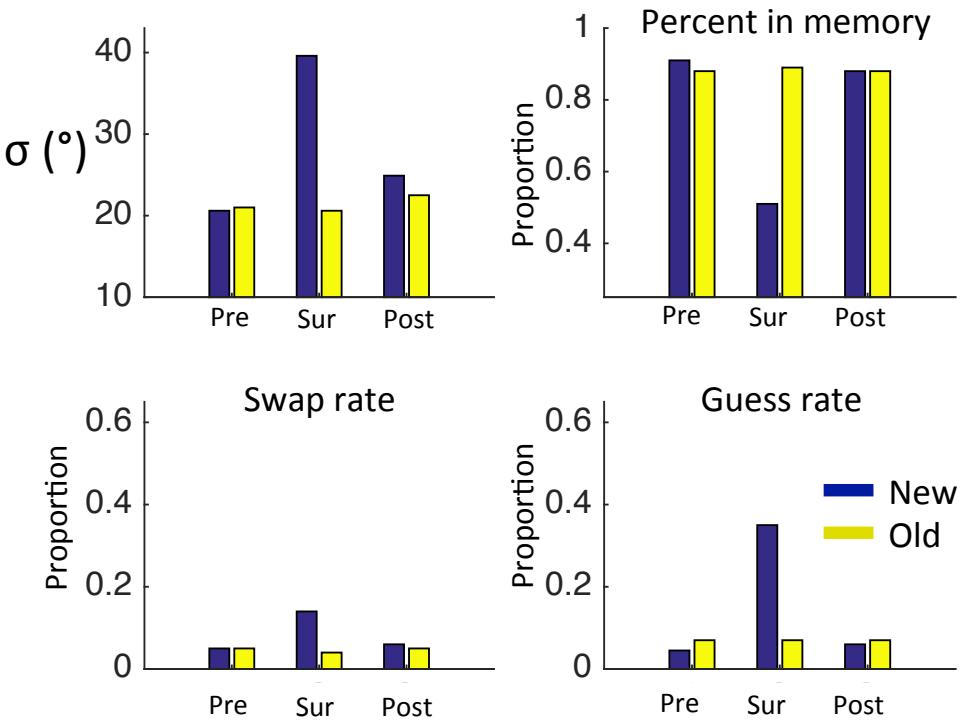


Figure 27. Simulation of Experiment 1 with a version of the Binding Pool model with the new encoding mechanism (Blue) and the old encoding mechanism (yellow), where parameter λ was fixed to 0.3 for the irrelevant feature in both simulations. See footnote 16 for more details about this simulation.

matrix. In 26 pre-surprise trials, there were 3 features and for a subset of the participants, the irrelevant feature was coarsely coded and for the other set, the feature was not encoded. Then, that feature was retrieved in the surprise trial. In 26 post-surprise trials, all features were precisely coded. Note that the largest standard errors were under 7 degrees for the surprise trial and less than .07 for the guess rate for the surprise trial.

New Encoding Mechanism

Encoding, as described in Chapter 2, was previously implemented as the conjunction between the type layers and the token that represents that object. An alternative encoding mechanism that will allow feature relevancy to affect precision is to bind each presented type to a token in sequential order (Figure 28)

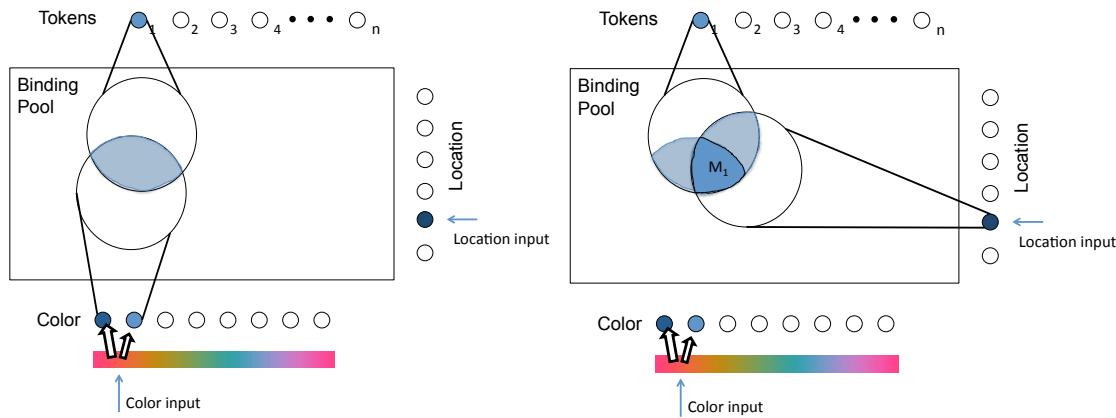


Figure 28. Illustration of how the new encoding mechanism works. Instead of all types and a token being bound to represent an object, there is a serial binding between individual types and a token. In this illustration, color is being bound to the token first. Then, location is being bound. The cell assembly for the encoded object (M_1) now includes all nodes shaded in blue with the darker blue corresponding to more active binding pool nodes and lighter blue corresponding to less active type nodes

With this new version of encoding (see Appendix B for Matlab code), the model can simulate the results from Experiment 1 (Blue; Figure 27). Importantly, the model simulates both the trade-off between the number of features that need to be encoded and the coarseness of an irrelevant feature. This model is able to simulate these results because of two additional properties that arise from this new encoding mechanism. First, when type layers are sequentially bound to a token, different binding pool nodes will have different amounts of activity depending upon the amount encoding strength of the type layer. For example, when a type layer has

a lower activity (i.e. its irrelevant), the binding pool nodes connected to both that irrelevant type layer and the token will have less activity than binding pool nodes connected to a relevant type layer and the token. This change enables the model to show the trade-off in precision when adding features to the memory load because there is less interference during retrieval when features are task irrelevant.

In addition, sequentially binding each type to a token means that some subset of binding pool nodes will only code for the storage of the task irrelevant feature. In addition, other binding pool nodes code for relevant features and perhaps even more nodes code for combinations of relevant and irrelevant features. This is of course is the result of the distributed nature of binding in the Binding Pool model. This heterogeneity allows the model to simulate the coarse coding of a task irrelevant feature, given that noise from the higher activated binding pool nodes connected to relevant features will interfere with retrieval of an irrelevant feature. Therefore, the model is able to simulate the coarse coding of irrelevant information. One advantage that arises from this new encoding mechanism is that the amount of activity in the binding pool increases when adding features to a representation. In the old version of the model, the activity in the binding pool was the conjunction of all features and the token. Therefore, with each additional feature being encoded, there are fewer binding pool nodes coding for that object. One potential limitation of the old model (Swan & Wyble, 2014) is that it suggests that an extremely complex stimulus with many features would be represented with fewer neurons than simpler stimuli. In the new version of encoding, the amount of activity in the binding pool increases when adding features to a representation (e.g. Luria, Sessa, Gotler, Jolicœur, & Dell'Acqua, 2010).

Parameter Fitting for New Model

The new parameter values (Table 2) were found by minimizing the root mean square error between the model's simulation and Bays et al. (2009). Most of the parameter values are similar to the parameters selected for the Swan and Wyble

(2014) version of the model¹⁷. For this fitting process, change detection was not simulated given that none of the simulations here are change detection experiments. Simulations of the new version of the model and the model from Swan and Wyble (2014) are displayed alongside Bays et al. (2009) in Figure 29.

Table 2. Parameters of the Binding Pool model with the changes to encoding and featural attention.

Parameter	Description	Value
Binding Pool Size (ω)	The amount of available nodes for storing links; as the size increases, so does the representational fidelity of stored information.	2500
Type Layer Size (ρ)	The amount of type nodes in each feature space	10
Connection Sparsity (α)	The proportion of active connections between the each layer and the binding pool	X ¹⁸
Type Connection Overlap (L)	The proportion of added overlap between neighboring type nodes	0.30
Encoding Capacity [Ψ_ω]	The number of stimuli encoded per trial is drawn for a uniform distribution	[2 7]
Token individuation threshold (ς)	If the ratio of the maximum retrieved token activity to the average retrieved token activity does not exceed this threshold, then the model will initiate ‘guess’ by choosing an unbound token.	0.015
Featural attention (λ)	The sum of the amplitude for a given type layer	1

¹⁷ One new, albeit unnecessary, parameter added to the model is that the binding pool activity after encoding is squared. This squaring is roughly equivalent to doubling the number of binding pool nodes to 5000. However, doubling the number of binding pool nodes increases the simulation duration 2.5 times.

¹⁸ The connectivity weight for the cue dimension (e.g. location in a normal color relevant delayed estimation experiment) is set to .5, which increases retrieval precision. For any feature not being used as a cue, the activity is set to .2. And lastly, the connectivity of the token layer was reduced to .12. Note that while there was an initial fitting process to determine these parameter values, these parameter values were fixed for all of the simulations described in this thesis unless otherwise noted.

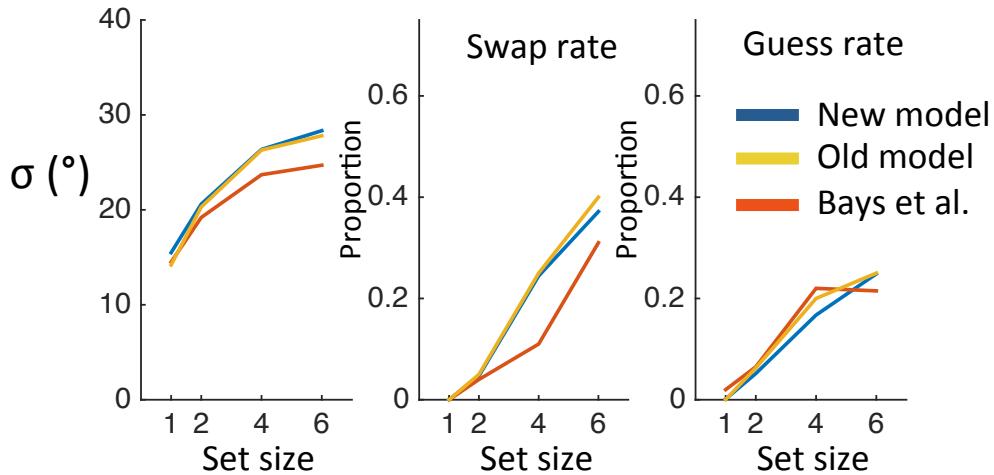


Figure 29. Simulation of delayed estimation from Bays et al. (2009) using the old version of the Binding Pool model (Swan & Wyble, 2014) and with the new version of the Binding Pool model described in Chapter 4. There were 5000 trials per set size in each simulation.

Conclusion

Two experiments were conducted to determine if participants coarsely code task irrelevant features when multiple objects are presented. When two objects were presented, the evidence replicated previous findings that participants coarsely code irrelevant features and participants exhibited a cost in memory quality on the initially task relevant feature when the task irrelevant feature became relevant to the task. In Experiment 2, four objects were presented to participants and the same trade-off in memory quality was found. However, most participants guessed on the surprise trial instead of showing a coarse, albeit accurate, response.

The data can be simulated with the Binding Pool model by manipulating the amplitude of activation during encoding, such that irrelevant features are encoded with less precision than relevant features.

Chapter 5. The Precision of Repetitions

Throughout the day, we often experience repetitions, or copies, of information. For example, when someone desires a donut at the bakery, comparing two of the same type of donut requires representing repetitions into memory. Being able individuate the representation of copies of information is a fundamental property of memory (e.g. problem of 2, Jackendoff, 2002). Understanding how repetitions of information are encoded was a popular area of research in the 1980s-90s with repetition blindness (Kanwisher, 1987; Kanwisher, Driver, & Machado, 1995). In these tasks, participants were rapidly presented with two sequential stimuli (e.g. letters, words, pictures). If the two stimuli were repetitions, participants' ability to recognize the second stimulus is diminished. An explanation of this phenomenon is that when perceiving individual objects, tokens (see Chapter 2) are assigned to each object. In rapid serial visual presentation and with repetitions specifically, the token account suggests that errors arise when there is a failure to bind the 2nd repetition to a token, so only a single object is perceived.

One way to better understand how repetitions of information are encoded is to measure the memory quality of a repetition using delayed estimation. Delayed estimation is a visual working memory (VWM) task that focuses on the retrieval precision of a memory. The most prototypical delayed estimation tasks involves remembering briefly presented colors and then reporting by selecting the specific color on a color wheel (e.g. Zhang & Luck, 2008). This metric is sensitive to differing qualities of a memory and may provide insight into differences between repetitions and non-repetitions in VWM.

When thinking about how the precise repetitions are, one approach is to think about the precision of repetitions compared to a single object. A single object would be the same type of information encoded (e.g. same color), but when repetitions are presented, there are multiple objects. It would also be useful to compare repetitions to the same number of objects that are not repetitions. The Binding Pool model provides one account about how these different conditions may

differ with regards to retrieval precision. Based on simulations from the model (Figure 30), participants should be most precise when presented with just a single object. Participants should then be more precise when presented with multiple repetitions than multiple randomly selected color values. The Binding Pool model makes this prediction because when there are multiple repetitions, each gets its own token. That is, there is nothing inherently difference about encoding repetitions compared to randomly colored objects except that the encoded features are the same. To determine if this is the case, two experiments were conducted to explore how repetitions are encoded into memory.

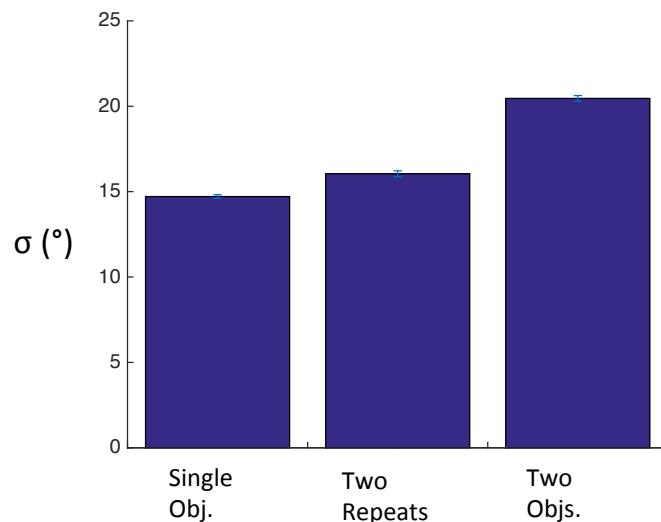


Figure 30 – Precision generated from simulations of the Binding Pool model when presented with a single color patch (Single Obj.), two repetitions (Two Repeats), and two randomly selected color patches (Two Obsjs.). Errors bars correspond to the standard error from 10 iterations of 2000 trials of each condition. Note that this is a simulation of Experiment 2 and that a simulation of Experiment 1 results in a similar pattern of data with regards to the different conditions.

Experiment 1

Participants: Thirty-four participants were recruited from the subject pool at Pennsylvania State University. Four participants were omitted from analyses, described below. All participants were presented instructions in American English

and signed an informed consent approved by the Pennsylvania State University IRB. Participants received course credit for their participation.

Apparatus: The experiment was run using Matlab 7.9.0 (build R2009b) with PsychtoolBox (Brainard, 1997; Pelli, 1997) scripts on computers running Windows XP (Microsoft, Redmond, WA). The screen resolution was set to 1024x768 at a 75-Hz refresh rate on cathode ray tube monitors with a diagonal screen size of 40cm. Participants were seated in a chair with an adjustable height and situated in chin rests located 50 cm from the monitor.

Stimuli: The stimuli consisted of a color patch that could vary in color across trials. Color was selected from a series of 360 colors in a one-dimensional selection from CIE L*a*b* color space where L was fixed at 54 and a circle was drawn with a radius of 50.5 and origin centered at a = 22.5 and b = 11.5. Each color patch had dimensions 3.6 by 3.6 of visual angle. Color patches appeared along an invisible circle with radius 4.9 degrees of visual angle. When 3 color patches were presented, they were equidistant from each other..

Perceptual matching: Participants first completed 10 trials of a perceptual matching task. In this task, two color patches were presented along the horizontal meridian separated by 10.8 degrees of visual angle. Participants were instructed to match colors between the two stimuli by selecting a color value from a surrounding color wheel. Across trials, the rightmost color patch had a color that was uniformly sampled from 10 possible color values separated by 36 degrees to cover the extent of the wheel. When participants moved their mouse to the color wheel, the leftmost color patch changed to the color value of the location of mouse cursor. Participants were free to move the mouse until satisfied with their answer, which they could record by clicking the left mouse button. Note that this method of responding was used in the delayed estimation task. After responding, the next trial began 500ms later.

Delayed estimation: After completing the perceptual matching task, participants began the experimental trials (Figure 31). At the start of each trial, there was a fixation screen for 250ms. Then, color patches appeared for 150ms. After the offset of the stimuli, there was a retention interval for 900ms. Then, the

response screen appeared. On the response screen, participants were asked to report one of the colors. Here, the location of a white box cues participants to which color needs to be reported. After making a response, feedback was presented to participants. The feedback screen consisted of the selected color and the actual presented color.

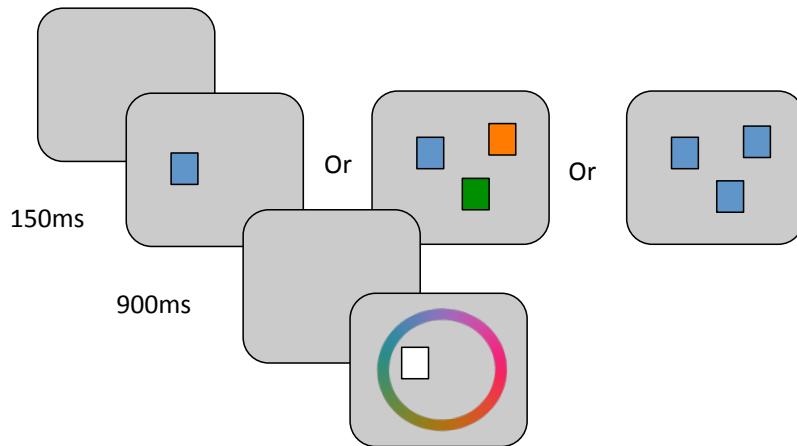


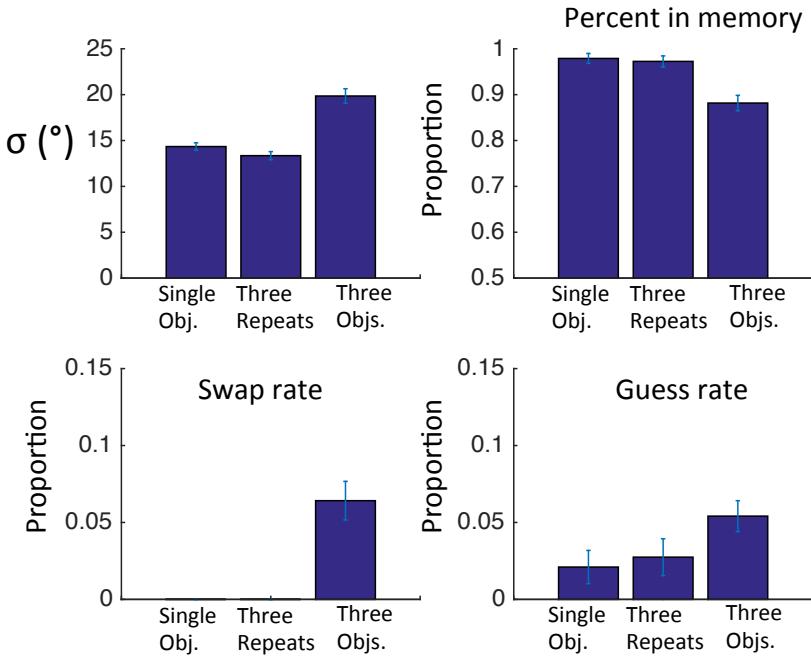
Figure 31 – Illustration of the procedure for Experiment 1. Participants were presented with either 1, 3 repetitions, or 3 randomly colored patches. After a brief retention interval, participants were cued by location to report one of the presented colors by selecting a color value along the color wheel.

There were three conditions, which were intermixed across trials. In the first condition (single object), participants were presented with a single color. In the second condition (three random), participants were presented with three randomly selected colors. In the last condition (three repetitions), participants were presented with three colors of the same selected color value. Participants ran the experiment for 26 minutes and completed an average of 233 trials (standard deviation = 78). Note that precision was not significantly correlated with the number of trials completed (lowest $p = .28$ and $r = .2$). Participants were not given a fixed number of trials in order to maximize the number of trials that could be completed within the available time slot (i.e. less than 30 minutes).

Results and Discussion

Error was calculated by taking the difference between the presented color and the reported color. It is assumed that the error is composed of multiple types of responses, such as guesses and retrieval errors (Bays et al., 2009). This mixture model is composed of two circular Gaussian distributions called von Mises and a uniform component, which allows for the precision of the memory responses, guess rate, and swap rate to be extracted from the response distribution (see Chapter 1 for more details). Note that the set size 1 condition and the repeated colors condition cannot have reportable retrieval errors, so only precision and guess rate could be estimated from error distributions generated from those conditions.

Participants with performance that resulted in unusual parameter estimates were excluded from the study. Of the 34 participants, 2 participants were removed for having proportion in memory (i.e. $1 - \text{guess rate} - \text{swap rate}$) less than 2 standard deviations before the mean. An additional 2 participants were removed for having trial counts below 80. Note that including the excluded participants does not change any of the following results. Parameter estimates are displayed in Figure 32 for the different conditions.



Given that the focus of the model's prediction is on precision across conditions, precision was the only parameter estimate analyzed. Precision across conditions was on average 14.1 degrees (SEM: 0.4) for the single object condition, 19.9 (SEM: 0.8) for the three-object condition, and 13.2 (SEM: 0.5) for the repeated object condition. To determine whether precision was different across conditions, paired t-tests were used. First, when comparing the repetition condition to the 3 object condition, there was a significant difference [$t(29) = 11.5, p < .001$], which indicates that participants were more precise on the repetition condition. The repetition condition was also significantly different from the single object condition [$t(29) = 2.7, p = .01$], which suggests that participants were more precise on the repetition condition than the single object condition. Lastly, the single object condition was significantly more accurate than the three-object condition [$t(29) = 8.7, p < .001$].

These results indicate that participants were most accurate when 3 repetitions were presented compared to a single object. Both the single object and three repetition conditions were more precise than three randomly presented colors. These results fail to support the prediction from the Binding Pool model,

which predicted that repetitions would be less precise than a single object, but more precise than three objects. It is possible that participants were able to use ensemble effects to improve their precision when three repetitions were presented simultaneously. That is, the three repeated copies presented simultaneously might have been perceptually integrated such that only a single large object was encoded into memory (Brady, Konkle, & Alvarez, 2009). In Experiment 2, participants were presented with objects sequentially. By sequentially presenting the objects, participants may be less able to benefit from grouping objects together. In addition to presenting objects sequentially, the set size in Experiment 2 was reduced to two objects to minimize guesses and retrieval errors.

Experiment 2

Participants: Thirty-three new participants were recruited from the subject pool at Pennsylvania State University. Three participants were excluded, described below in the analysis section. All participants were presented instructions in American English and signed an informed consent approved by the Pennsylvania State University IRB.

Stimuli and Procedure: The stimuli and procedure were the same for Experiment 1, except where noted. The first color patch appeared at fixation for 150ms followed by a mask of randomly colored patches that onset immediately after the offset of the stimulus for 100ms. Next, there was an interval for 500ms. Then, the second color patch appeared at fixation, followed by a mask, and then the retention interval for 500ms. The response screen consisted of a white box with a number inside of it (either 1 or 2) and a color wheel surrounding the white box. Participants were instructed to recall the color that was either presented first (i.e. when a '1' was located in the box) or the second color patch (i.e. when a '2' was located in the box).

There were four conditions (Figure 33), which were randomly shuffled across trials. In the first two conditions, participants were presented with sequential

displays of 2 color patches with colors drawn from the same color space as Experiment 1. In Condition 1, the colors were randomly selected. In Condition 2, the color patches were exactly the same (i.e. repetitions). In Conditions 3, only a single stimulus appeared and that stimulus appeared in the first serial position (as described above). Note that without a second stimulus, there was only a single mask following the offset of the single stimulus and the mask was followed by a retention interval of 1250ms. In Condition 4, only a single stimulus appeared and that stimulus appeared in the second serial position. There was no mask preceding the presentation of this stimulus. The design of the experiment was 2 (set size: 1 or 2) x 2 (report first or second object) x 2 (repeat or non-repeat), resulting in a total of 18 trials per condition and 144 total trials. Note that for set size 1, the third factor (i.e. repeat or non-repeat) was negligible. Also note only 11 out of the final 30 participants completed all 144 trials. The experiment was timed to conclude at 26 minutes. The average number of trials for those that did not complete the full set was 114 trials (standard deviation 17) with the fewest being 95 trials.

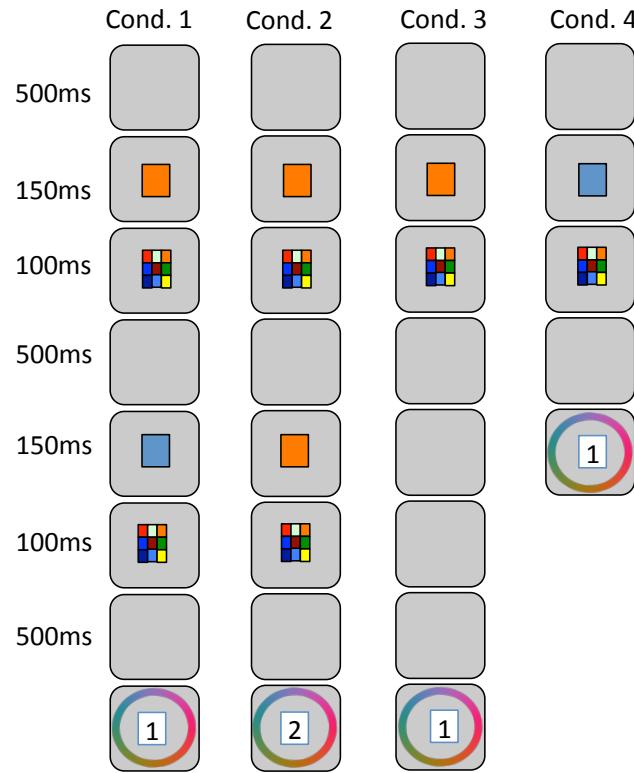


Figure 33 – Illustration of the procedure for Experiment 2. Here, participants were either asked to remember the color of either one or two presented color patches. There were 4 conditions: two randomly selected colors (Cond 1), two repeated colors (Cond 2), a serial position 1 report with a single color (Cond 3), and a serial position 2 report with a single color (Cond 4). Participants were cued by serial order (i.e. 1 means report the first object). Like in Experiment 1, participants responded by selecting a color value along the color wheel.

Analysis

The standard mixture model was used to estimate precision and the guess and swap rates. Of the 33 participants, 2 were excluded for having percent in memory estimates two standard deviations below the mean and another participant was excluded for having fewer than 75 total trials. Note that including the excluded participants changes none of the following results. This resulted in a total of 30 participants. Parameter estimates are displayed in Figure 34.

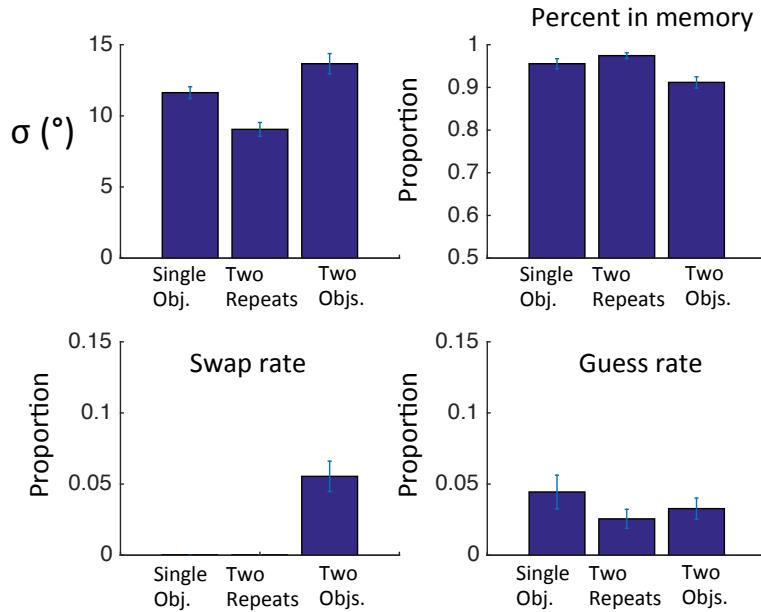


Figure 34 – Parameter estimates for Experiment 2. Note that for both the repetition and single object conditions, there cannot be swaps. Error bars are standard error.

As in Experiment 1, the primary parameter of interest was precision across conditions. Unlike Experiment 1, Experiment 2 had an additional factor of serial position. First, precision was compared with serial position factored across and the set size 1 conditions combined. The average precision for Condition 1 (set size 2) was 13.3 degrees (SEM: 0.7), for Condition 2 (repeat) was 8.9 (SEM: 0.5), and combined across Conditions 3 and 4 was 11.5 (SEM: 0.4). Like Experiment 1, paired t-tests were used to compare precision across conditions. First, when comparing the repetition condition to the 2 object condition, there was a significant difference [$t(29) = 7.6, p < .001$], which indicates that participants were more precise on the repetition condition. The repetition condition was also significantly different from the single object condition [$t(29) = 6.2, p < .001$], which suggests that participants were more precise on the repetition condition than the single object condition. Lastly, the single object condition was significantly more accurate than the two-object condition [$t(29) = 2.9, p < .01$].

While precision was higher in the repetition condition than the single object condition in the above analyses, it is possible that that was only the case because the single object conditions were combined. It is possible that Condition 3 (single object presented in serial position 1, where the retention interval is 1250ms) was driving the difference in precision for the combined single object conditions. Therefore, it is important to compare the repetition condition to Condition 2 (single presented object in serial position 2). If there is no difference, it may suggest that participants are treating the repetition condition as if a single object was presented. However, there is a significant difference in precision [$t(29)=3.3, p < .01$], which suggests that participants' precision improved when presented with repetitions.

One possibility that may explain the higher precision in the repetition condition is that this was similarity benefit, which has been previously reported in the literature (Lin & Luck, 2009; Oberauer & Lin, 2017). If that is the case, there may be a difference in precision as a function of the serial position. For example, participants may be encoding the color in serial position 2 relative to the color in serial position 1. There were not enough trials to accurately estimate the parameter fits when splitting the repetition condition by serial position (i.e. less than 20 trials per fit). Instead, a permutation test was conducted on the parameter estimates to the error distribution collapsed across participant for the different serial positions. The estimated precision for serial position 1 and serial position 2 was 9.8 and 9.1 degrees, respectively. The log likelihood of the best fitting model to the serial position 2 data is 147.7. When that same data is used to compute LL with respect to the best fitting model for serial position 1, the resulting log likelihood (LL) is 143.9. To determine whether this LL difference of 3.8 is significantly different, the data between the two conditions was permuted. Each permuted data set was fit to the mixture model and the same LL difference was computed. Out of 1000 iterations, .22 had a greater difference, which suggests there is not enough evidence to determine whether there is a difference in responses between the serial positions.

Another way to compare the repetition benefit to the similarity benefit is to see how well participants in the set size 2 random condition performed at different degrees of similarity compared to precision on the repetition condition. Responses

collapsed across participant were put into bins from 0-36, 37-72, 73-108, 109-144, and 145-180 degrees depending upon the absolute value of the difference between the presented colors. Like in Experiment 2 in Chapter 3, participants were more precise at the more similar bin (10.1 degrees) and the most dissimilar bin (14.3 degrees) than intermediate levels of similarity (15.1, 15.5, and 16.3, respectively). These results replicate the findings of precision having an n-shaped function when binned by similarity in Chapter 3. Also, participants in the most similar bin were less precise by 1.2 degrees than the average precision of the repetition condition. It seems likely that participants' precision continues to increase as two objects become more similar and are most precise when the two objects are repetitions.

General Discussion

Here, two experiments were conducted to determine how well repetitions were encoded into memory. The results of the experiments showed that participants were more precise when presented with repetitions than a single color. This was true for both simultaneous presentation (Experiment 1) and for sequential presented (Experiment 2).

Finding better precision for repetitions compared to a single object is perhaps consistent with models that represent memory strength as activity in the feature layers themselves (Wei, et al., 2012). In such models, a repetition would theoretically increase the strength of activation in that sensory space. However, it would be difficult to imagine how these models would differentiate repetitions as separate objects, which is a key advantage of the Binding Pool model. Yet, the results from the above two experiments fail to support the prediction of the Binding Pool model, which predicted that participants should be less precise when remembering repetitions than a single object. The Binding Pool model makes this prediction because multiple repetitions are treated as separated objects, thus each receiving their own token. By encoding each repetition to a different token, there is increased

interference in the model. It is thus important to consider different adjustments¹⁹ to the Binding Pool model that will allow the model to simulate these results.

Changes To The Model

As is the case with any other prediction, incorrectly predicting the data means that the model is incorrect and needs to change in order to accommodate new results. Two mechanisms that could be added to the model that improve the model's precision for repetitions is to bind both repetitions to a single token or to increase input activity in the type layer for the repetition. Both of these implementations were individually applied to the Binding Pool model, which already now incorporates the changes described in Chapter 4 regarding the new encoding mechanism.

Repetitions Bound To The Same Token

The first implementation that increases precision for repetitions is to bind repetitions to the same token. This change to the model is implemented by first setting a difference threshold whereby a repetition is detected. For example, if the absolute value of the difference between the presented objects is less than 2.5 degrees, which is approximately the precision of participants in a perceptual matching task (Swan, Collins, & Wyble, 2016), then the model binds those objects to the same token. Therefore, the repetition condition always results in the two objects being bound to the same token. This results in more precise memory of the

¹⁹ Since participants were more precise for repetitions than a single object, it does appear as if participants recognized the presentation of a repetition. If participants had no idea a repetition had been presented, participants may have been equally precise for the repetition and single object condition because participants would only be encoding a single repetition in the repetition condition. With that said, it may be the case that participants perceive a repetition, but do not store information about a repetition being presented.

repetitions because there are more binding pool nodes selective to that specific feature value and the bound token (Figure 35).

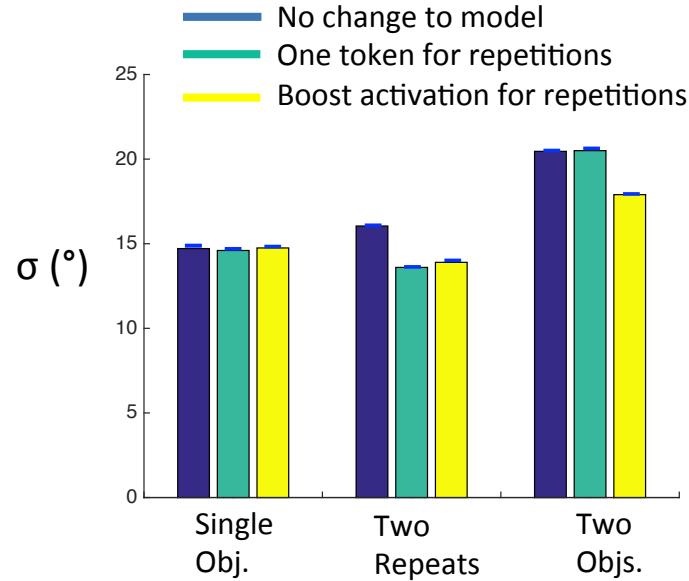


Figure 35. Binding Pool model simulations where the model is encoding a single colored object, repetitions, and two randomly colored objects. Note that the original implementation of the model (Blue), the data incorrectly predicts that participants are less precise for repetitions compared to a single object. In both proposed changes to the model (i.e. binding both objects to the same token (Green) and boosting activation during encoding (Yellow)), the model is able to simulate the precision benefit of repetitions. Errors bars correspond to the standard error from 10 iterations of 2000 trials of each condition.

This implementation of binding multiple objects to the same token is similar to how the STST model handles a visual attention effect called lag-1 sparing in the attentional blink literature. In attentional blink paradigm, targets and distractors are presented to participants in rapid succession (often 10 items per second), and participants must detect and report the targets that appear in the stream of distractors. When two targets appear with an intervening distractor, an effect arises called the attentional blink (for review, see Martens & Bays, 2010), where the probability of reporting the second target diminishes if participants successfully report the first target. Interestingly, when two targets appear back to back, the ability to report both targets is increased. The STST simulates this result by binding

both targets to the same token. This implementation allows the STST model to simulate order-effects associated with lag-1 sparing. That is, participants often swap the order of the targets, such that participants report the first target as appearing second and vice versa.

The Binding Pool model has the same effect when binding multiple objects to the same token. This was determined by running a simulation of the repetition condition with three features being encoded per object. The third feature, which is neither the reported nor the cue feature, was able to repeat between the two objects. In the simulation, when the third feature repeated, the two objects were bound to the same token²⁰. In the resulting simulation, the model's swap rate increased dramatically to 50% when the two objects were bound to the same token. This occurred because now the token is connected with binding pool nodes activated from features of both objects.

In Experiments 1 and 2 in this Chapter, participants make a single response and report the feature (i.e. color) that is repeated between multiple objects, so the current experiments cannot be used to infer whether participants are swapping other object features. However, in Chapter 4 Experiment 1, participants were instructed to remember the color and direction of two colored arrows in the second half of the experiment. On each trial, participants did not know which feature (i.e. color or direction) would be probed in a delayed estimation task. Therefore, this data set may provide some insight into whether there are more swaps when the non-reported feature was similar between the two stimuli.

Parameter estimates collapsed across participant for color responses and direction responses as a function of whether the similarity²¹ between the non-

²⁰ This simulation presumes that any single repeated feature between two objects results in both objects being bound to the same token. This seems highly unlikely. However, this simulation does provide an upper bound on what to expect if objects are bound to the same token. Future experiments will need to be conducted to determine exactly what criteria is necessary for objects to be bound to the same token, if that is the mechanism that is happening here.

²¹ Ideally, only repetitions (or less than 2 degrees absolute value difference) would be used. However, this would result in too few trials for the analysis. Instead, the threshold for determining whether participants may bind objects to the same token

reported feature are displayed in Table 3. From this preliminary comparison, it appears as if there were more swaps when the non-reported feature was similar in feature space than when the non-reported feature was not similar (2.3% increase for direction and 8% increase for color). This result seems to qualitatively support, at least for color, the prediction from the Binding Pool model and may be a suitable implementation for representing repetitions. However, additional experiments will need to be run in order to determine more precisely when multiple object will be bound to the same token.

Table 3. Parameter estimates from Experiment 1 in Chapter 4 divided by whether the non-reported feature was similar (see footnote 16) or dissimilar

		Similar				Dissimilar	
	σ (°)	Pnt	Pu		σ (°)	Pnt	Pu
Color	21.9	.19	.04		22.3	.11	.08
Direction	16.3	.11	.07		15.8	.087	.1

Boosting Type Activation

A different mechanism for increasing precision for repetitions in the Binding Pool model is by boosting activation in the type layer when that type is being bound to a token. Specifically, this boosting would happen for the second repeated feature. Like in the previous implementation, the model would first need to determine whether the two presented objects are repetitions or not (i.e. the absolute value of the difference is less than 2.5 degrees). Then, if there were a perceived repetition, the model would boost activity in that type layer for the encoding of the second repetition. This type of implementation is similar to how the model can be used to represent task irrelevant features. That is, a parameter λ that is normally fixed to 1 would be adjusted. Although, unlike task irrelevant features, λ would be increased to a value greater than 1 when simulating this boosting of activation. When this boosting is implemented in the model, the model is able to simulate the increase in

was 24 degrees for color and 18 degrees for direction, which is the respective average precision across participants. This threshold is conservative and designed to maximize the number of trials in the ‘repeated’ condition.

precision for repetitions (Figure 35) because the binding pool nodes connected to the repetitions have higher activation during retrieval.

One way to find evidence in favor of this implementation is to unexpectedly ask participants to report the number of objects that were presented. Since boosting of the type activation does not change the number of tokens, participants should be able to report the number of presented objects. However, if the repetitions are bound to the same token, it may be the case that participants have repetition blindness (Kanwisher, 1987), which would result in participants only reporting a single object instead of repetitions.

Conclusion

In two experiments, repetitions were found to be more precise than a single object. This result was found for both the simultaneous presentation of 3 objects (Experiment 1) and also the sequential presentation of 2 objects (Experiment 2). This increase in precision is inconsistent with a prediction from the Binding Pool model. In order for the Binding Pool model to simulate these results, the model needs to be adjusted. Two possible implementations were presented; binding repetitions to the same token or increasing the activation strength of the repeated feature dimension. Future research will need to be conducted to determine which of the proposed implementations needs to be incorporated into the model.

Chapter 6. Predicting ERPs using the Binding Pool model

Visual working memory (VWM) is a temporary storage system for visual information. Recently, VWM has been explored using both recognition (i.e. change detection, e.g. Luck & Vogel, 1997) and recall (i.e. delayed estimation, e.g. Bays et al., 2009). One interesting difference between the results of these methodologies is that recall measures of VWM show a trade-off between the number of objects and memory precision. These same trade-offs are often not found with change detection experiments, which typically find performance near ceiling until capacity is exceeded (e.g. Luck & Vogel, 1997; c.f. Keshvari et al., 2013). The Binding Pool model provides one computational explanation of VWM that can account for the findings using VWM recall and recognition (Swan & Wyble, 2014).

The Binding Pool model simulates the storage of information as the activation of nodes in a binding pool that are connected to the visual features of an object and an object file like representation called a token (for more details, see Chapter 2 or Swan & Wyble, 2014). Information can be retrieved from the binding pool by first using binding pool nodes connected to projections from a cue to retrieve a token. Once a token is retrieved, binding pool nodes connected to input from the cue and the token are used to retrieve the desired feature. These components (i.e. visual features called types, tokens, and binding pool) and mechanisms enable the Binding Pool model to simulate both change detection and delayed estimation experiments (see Swan & Wyble, 2014, and chapters 3-5). However, assessing the quality of a model is not just dependent upon the ability to simulate data, but also the ability to make testable predictions. One domain where the Binding Pool model may provide useful insight is in research using event-related potentials (ERPs) to explore the neural correlates of VWM. Finding evidence that the Binding Pool model can predict ERPs would give the Binding Pool model additional constraints and enable the model to make insights into the mechanisms that may underlie these neural correlates.

The ERPs most relevant to the encoding and storage of information are the P3 and the contralateral delay activity (CDA), which are receiving increasing attention in the VWM literature. Here, each component is described in the context of working memory.

P3 components

The P3 component, which is generally thought to be related to working memory updating, is found in a variety of cognitive tasks and is typically differentiated into two components; the P3a and the P3b. Both components are bilateral, but there are interesting differences in what elicits them. First, the P3a seems to be related to the presence of an oddball stimulus. For example, a reliable P3a is found when a relevant letter appears intermixed randomly within a stream of digits (Dell'Acqua et al., 2015). The P3a peaks around 300ms after the onset of the stimulus and is largest in midfrontal electrodes. The P3b seems to be related to the consolidation of visual information into VWM (Vogel, Luck, & Shapiro, 1998; Akyürek, Leszczyński & Schubö, 2010). Unlike the P3a, the P3b typically peaks around 450ms after the onset of the stimulus and is largest in midparietal electrodes. Another difference between the P3a and P3b is that the P3a seems to be primarily affected by the number of targets (Dell'Acqua et al., 2015), whereas the P3b is affected by the number of targets and also target complexity (Berti, Geissler, Lachmann, & Mecklinger, 2000).

The Binding Pool model predicts, based on the anatomical and functional distinctions, that the P3a may reflect the allocation of tokens and the P3b may reflect activity in the type layers during encoding. Given this distinction, the Binding Pool model predicts that changing the number of relevant features would only affect the P3b, which could be testable with a methodology similar to Swan, Collins, and Wyble (2016). In Swan, Collins, and Wyble, the number of relevant feature was manipulated and resulted in a measureable trade-off between the number of relevant features and memory precision.

CDA component

The CDA component is thought to reflect the maintenance of information in VWM (Vogel & Machizawa, 2004). The CDA is a lateralized component found in parietal and occipital electrodes and persists for several hundred milliseconds (i.e. from 350ms to approximately 1000ms) during the retention interval following the onset of the stimuli. The amplitude of the CDA is calculated by taking the difference between the electrodes contralateral to the visual hemifield cued at the start of a trial from ipsilateral electrodes.

The amplitude of the CDA has been found to be insensitive to a variety of task manipulations, such as the spatial proximity of the presented objects (McCollough, Machizawa, & Vogel, 2007), the perceptual difficulty of the objects (Ikkai, McCollough, & Vogel, 2010), the number of relevant locations (Ikkai, McCollough, & Vogel, 2010), and eye gazes (Kang & Woodman, 2014). By ruling out these other explanations, many researchers have come to the conclusion that the CDA specifically tracks the number of objects held in memory (Luria, Balaban, Awh, & Vogel, 2016).

In addition to the above manipulations, the CDA is seemingly unaffected by the number of relevant features (Luria, & Vogel, 2011), which suggests that the CDA is simply a reflection of the number of encoded objects. However, this conclusion is controversial and seems to depend upon which features are relevant to the task (Woodman & Vogel, 2008; Gao et al., 2009). For example, Woodman and Vogel reported CDA amplitude while manipulating the number of relevant features (i.e. color and orientation) of either 2 or 4 orientated color lines. They found that the CDA amplitude was the same for the conjunction and orientation relevant condition, but the CDA amplitude was lower for the color relevant condition.

If the CDA measures memory load, then presumably the CDA in the Binding Pool model would be interpreted as the sum of the activity in the binding pool after all objects have been encoded. Both the number of objects and the number of

relevant features affects the activity in the binding pool. Therefore, the Binding Pool model predicts that the number of relevant features should affect the CDA. Perhaps more sensitive measures of feature relevance (e.g. Swan, Collins, & Wyble, 2016) may elicit differences in CDA amplitude as a function of the number of relevant features.

Predicting ERPs with the Binding Pool model

When considering how the Binding Pool model could make predictions regarding ERPs, it is worth noting that there is no ‘time’ in the model, such that the model cannot represent the cascade of events required to actually simulate a time series. However, the model nevertheless offers a potential explanation of the underlying neural mechanisms of the CDA, P3a, and P3b components, since it does make a distinction between different stages of memory stages.

With regards to the CDA and P3b, the model predicts that these components should be correlated given that both components are related to the number of objects and number of relevant features. The model also predicts that only the number of objects and not the number of relevant features would affect the P3a, given that the model predicts P3a to reflect token allocation. In order to describe how the model makes these predictions, an example of encoding multiple color patches is described below.

When stimuli are presented, there is a top-down signal that assigns tokens to each object. This top-down signal would elicit a P3a ERP, which would have greater amplitude with increasing number of objects to encode. Next, there is a binding process between the tokens and the types in the binding pool. This requires some control processes not currently simulated in the Binding Pool model, but conceivably, this would result in changes to the amplitude of the P3b if more objects and more features were encoded. These predictions regarding the P3a and P3b are motivated from Wyble et al. (2009), which uses types and tokens to describe visual attention and is the precursor to the Binding Pool model. After encoding the objects,

the activity that is being maintained in the binding pool would correspond to the CDA (Figure 36). In this case, both the addition of relevant features and the addition of objects would increase activity in the binding pool.

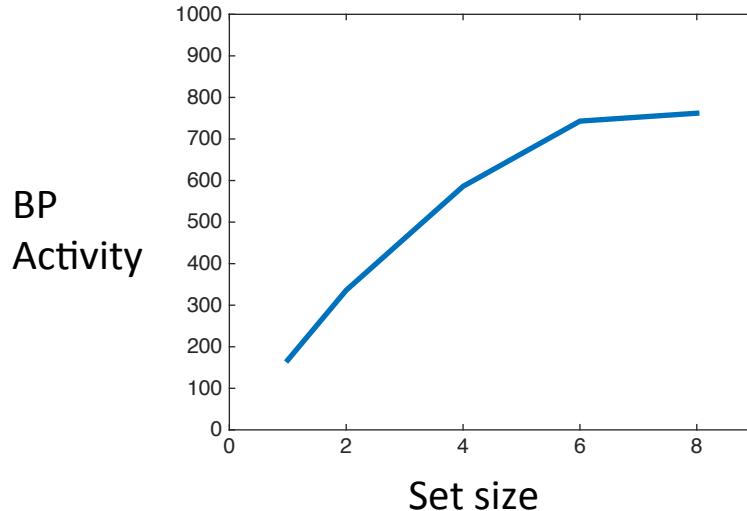


Figure 36. Simulation of the CDA as a function of set size with the Binding Pool model with 5000 trials per set size. The CDA, in this simulation, is a reflection of the amount of activity in the binding pool. Here, the sum of activity in the binding pool after all of the objects have been encoded is displayed.

These predictions were tested in a pilot experiment that resembled the methodology used in Swan, Collins, and Wyble (2016). By adopting this methodology, both the number of relevant features and objects were manipulated. One advantage of this methodology is that it was shown to be sensitive with regards to precision to changes in the number of relevant features. Thus, the cost in precision could be used as a predictor when analyzing the amplitude of the ERPs.

Experiment

Participants: Twelve participants were recruited from the subject pool at Pennsylvania State University. All participants were presented instructions in American English and signed an informed consent approved by the Pennsylvania State University IRB. Participants received course credit for their participation.

Apparatus: The experiment was run using Matlab 7.9.0 (build R2009b) with PsychtoolBox (Brainard, 1997; Pelli, 1997) scripts on computers running Windows XP (Microsoft, Redmond, WA). The screen resolution was set to 1024x768 at a 75-Hz refresh rate on cathode ray tube monitors with a diagonal screen size of 40cm. Participants were seated in a chair with an adjustable height and situated in chin rests located 91 cm from the monitor.

Stimuli: The stimuli consisted of a colored arrow that could vary in color and direction across trials. Color was selected from a series of 360 colors in a one-dimensional selection from CIE L*a*b* color space where L was fixed at 54 and a circle was drawn with a radius of 50.5 and origin centered at a = 22.5 and b = 11.5. The arrow was constructed using a rectangle with the dimensions .2.9 and .7 visual angle and an isosceles triangle with a .96 base and .7 height. Direction was chosen from 360 degrees as indicated by the arrow head.

EEG Recordings: A 32-channel sintered Ag/AgCl electrode array mounted in an elastic cap according to the 10-20 system (Fp1, Fp2, F3, F4, Fz, F4, F8, FT7, FT8, FC3, FC4, FCz, T3, T4, C3, C4, Cz, TP7, TP8, CP3, CP4, CPz, T5, T6, P3, P4, Pz, O1, O2, and Oz) (QuikCap, Neuroscan Inc.) was used, with the tip of the nose used for a reference. VEOG and HEOG electrodes were places on the lower and upper orbital ridge of the left eye as well as the outer canthus of each eye respectively. The amplifier was a Neuroscan Synamps with a band pass filter of 0.05-100 Hz and a sampling rate of 500 Hz. For analysis, the data was reduced offline to 250 Hz. Prior to the start of the perceptual matching, the impedance for all electrodes was lowered to 5 k Ω or less. All ERPs were time locked to the onset of the colored arrow stimuli and a three second epoch (1 second prior to the onset and 2 seconds after the onset) was used for each trial. Baseline activity from a window -200ms to 0ms relative to the onset of the stimuli was subtracted from each trial. If the average difference between HEOL and HEOR electrodes in a moving 32ms time window exceeded 20 μ V, the trial was judged as having contained a horizontal eye movement and was rejected from the analyses. If the difference between VEOU and VEOL electrodes increases by more than 100 μ V, the trial was marked as having contained an eyeblink and was also rejected from the analyses. Additionally, if any

channel exceeded an absolute value of 100 μ V during the epoched window, then that trial was additionally rejected. All artifact rejection and EEG analyses were conducted with EEGLab 13.3.2b (Delorme & Makeig, 2004) and custom MATLAB functions.

Perceptual matching: Participants first completed 10 trials of a perceptual matching task. The perceptual matching task ensured that participants knew how to produce responses when asked to report color, and importantly, direction. This block of trials was labeled Experiment 1 and participants were instructed to be as precise as they could be. In this task, there were two blocks randomly allocated depending on the parity of the subject number. In the color matching block, two colored arrows were presented along the horizontal meridian separated by 6.1 degrees of visual angle. Participants were instructed to match colors between the two stimuli by selecting a color value from a surrounding color wheel. Across trials, the rightmost color patch had a color that was uniformly sampled from 10 possible color values separated by 36 degrees to cover the extent of the wheel. When participants moved their mouse to the color wheel, the leftmost color patch changed to the color value of the location of mouse cursor. Participants were free to move the mouse until satisfied with their answer, which they could record by clicking the left mouse button. Note that this method of responding was used in the delayed estimation task. After responding, the next trial began 500ms later. In the direction matching block, the presented stimuli were the same, except that the color wheel was gray. When participants moved their mouse to the outside wheel, the arrowhead rotated to the angle of the cursor. Participants reported direction by clicking with the left mouse button.

Delayed estimation: In the second block (Figure 37), participants were instructed that the researchers were interested in how precise their memories could be and that only a single feature would be relevant (e.g. direction). Participants remembered either 1 or 2 colored arrows of 4 possible colors arrowed where the color and direction for all 4 arrows was randomly sampled from 360 possible values. At the start of every trial, there was a 1.5s prestimulus fixation screen. Next, there was a 200ms pre-cue that informed participants whether they would need to

remember just one arrow (i.e. circle with a single quadrant in white) or both arrows (i.e. circle with a hemisphere white). When participants needed to remember both arrows, the cued arrows were also either on the right or left side of the screen. Following the cue, there was a 400ms interval before the stimuli appeared.

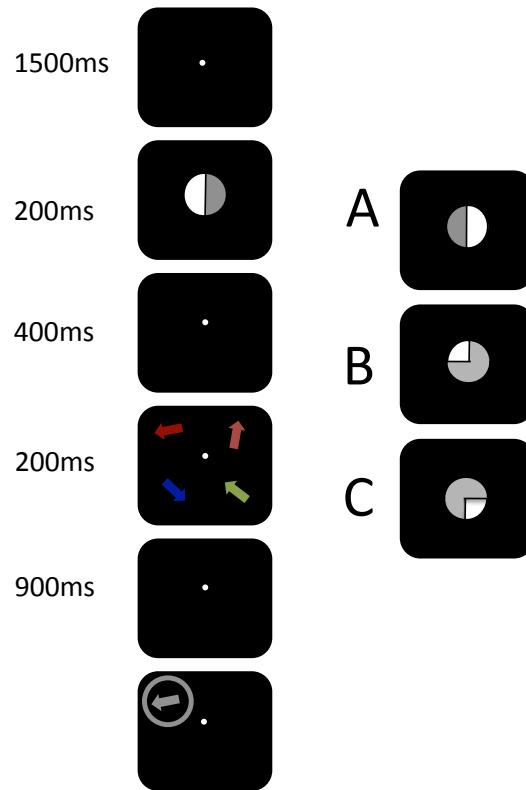


Figure 37. Illustration of the procedure for the experiment. Participants were first presented with a pre-cue that told participants which object or objects would be relevant. In the main illustration, the cue indicates that both arrows on the left are relevant. Examples of other pre-cues are displayed in A (to right arrows are relevant), B (top left arrow is relevant), and C (bottom right arrow is relevant). After the pre-cue, participants were presented with 4 colored arrows and then the response screen followed the presentation of the stimuli. Participants responded by moving their mouse to the wheel.

The arrows then appeared for 200ms on an indivisible square with sides that were 4.8 by 4.8 degrees of visual angle. After a 900ms retention interval, participants were presented with the response screen. For the first 88 trials, participants were presented with a gray wheel and consistently reported direction. On the 89th trial, participants were presented with a color wheel, a gray square in

the location of which arrow's direction needed to be retrieved, and a brief message that informed participants that this is a surprise question and that direction needs to be retrieved. The origin of the response wheel was centered at the location of one of the arrows and informed participants which feature from which object needed to be retrieved

Like the responses in the perceptual matching task, once participants moved their mouse cursor to the wheel, the stimulus reflected the corresponding feature value of the location of the mouse cursor. For the remaining trials (i.e. 90 to 265), participants could be asked about either the color or the direction of the one of the arrows.

The design of this experiment was thus 2 (left or right) by 2 (up or down) by 2 (set size 1 or 2) in the first half of the experiment for 88 trials and with an additional 2 factors (response feature: color and direction) for the remaining 176 trials. The surprise trial was always a color response, but the other factors were randomly selected given that primarily comparison of interest was pre-surprise versus post-surprise trials.

This experiment tested each prediction of the model (Figure 38). First, the model predicted that the P3a amplitude would only be affected by the set size manipulation because the P3a is a marker for token initiation. Second, the model predicted that the P3b and CDA would both be affected by the increase in relevant features and the increase in set size. And lastly, the P3b and CDA would be positively correlated, given that both components are related to the amount of information stored in the binding pool.

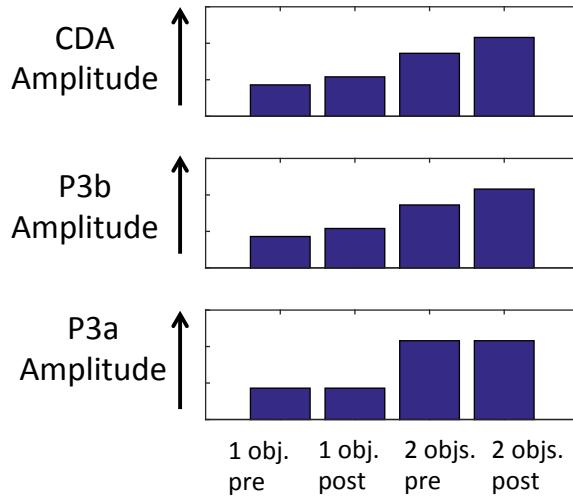


Figure 38. Predictions from the Binding Pool model about the amplitude of the CDA, P3a, and P3b as a function of the number of relevant objects and features. Obj refers to the number of presented objects. Pre and post correspond to pre and post-surprise trials, which refers to only a single feature being relevant in the pre-surprise and two relevant features in the post surprise trials.

Behavioral Results

First, the errors for direction responses were calculated by subtracting the reported direction response from the actual direction. Next, estimates of precision, guess rate, and swap rate were calculating by using a mixture model (Bays et al., 2009). One participant was removed for having a swap rate near 50% in the post-surprise trials, suggesting that the participant may have been ignoring the cue in post-surprise trials. For the remaining 11 participants, parameter fits are displayed in Table 4.

Table 4. Parameter fits for color responses

	σ ($^{\circ}$)	Pm	Pnt	Pu
Pre – Set size 1	16.6 (1.0)	.97 (0.02)	.014 (0.01)	.045 (0.01)
Pre – Set size 2	22.5 (1.2)	.94 (0.02)	.012 (0.01)	.038 (0.02)
Post – Set size 1	14.5 (0.6)	.94 (0.04)	.005 (0.01)	.054 (0.04)
Post – Set size 2	20.8 (1.5)	.86 (0.03)	.073 (0.02)	.06 (0.02)

Given that there are only 11 participants and that this is a pilot study, statistical comparisons were not run on the parameter estimates. However, there was a clear decrement in precision when comparing set size 1 to set size 2 overall. It was also worth noting that participants' precision seemed to improve in the post-surprise trials, which is unlike the findings from similar experiments (Experiments 1 and 2 in Chapter 4; Swan, Collins, and Wyble 2016). Given that there were only 11 participants, this was likely random variance. A couple of alternative explanations do not seem likely. For example, in Swan, Collins, and Wyble (2016), starting with direction (as was here) still resulted in a trade-off between the number of relevant features and precision. In addition, a trade-off was found in Experiment 1 in Chapter 4 for set size 2, so it does not seem likely that this was simply driven by differences in set size.

EEG results

The ERPs were examined next. Participants who had less than 25 trials remaining after the EEG exclusion criteria (see Methods section) were omitted from the ERP analyses. Out of the 11 participants used above, only 7 remained.

First, the CDA component was examined for electrode pairs P3 and P4. The contralateral, ipsilateral, and difference waveforms are displayed in Figure 39. The CDA amplitude was calculated by taking the mean amplitude from 300ms to 1000ms following the onset of the stimulus. This time range is typical of other CDA papers (e.g. Luria, Balaban, Awh, & Vogel, 2016). The mean amplitudes were .04 (SEM = .3), -.26 (SEM = .3), -.37 (SEM = .4), and -.21 (SEM: .17) for pre-surprise set size 1, pre-surprise set size 2, post-surprise set size 1, and post-surprise set size 2. There appears to be some minor effects of increasing the load beyond pre-surprise set size 1, but with this few participants, statistical tests were not conducted. The minimal differences in the mean amplitudes, which could also be observed in the waveforms (black) in Figure 39, indicate that there were not dramatic changes to

the CDA amplitude as a function of the number of objects and number of relevant features.

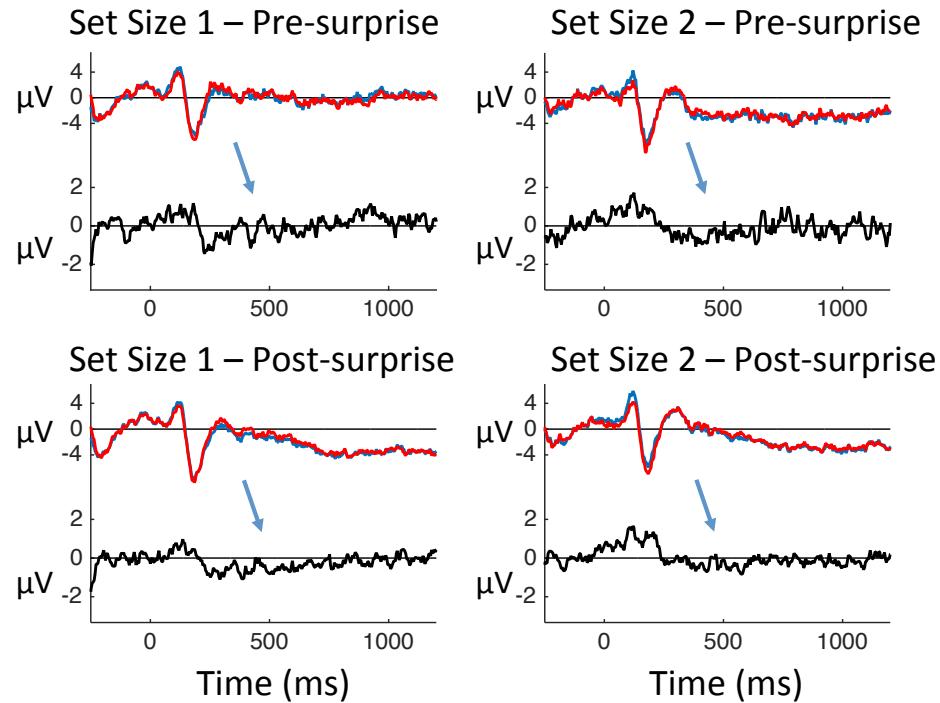


Figure 39. ERP time locked to the onset of the stimulus for electrode pair P3/P4. Contralateral (Blue) and Ipsilateral (Red) are presented in the top portion of the figure and the difference wave (Black) is presented below it. The arrow indicates where a CDA may be expected in the difference waves.

Next, the P3 components were examined for electrode Pz (Figure 40) and electrode Fz (Figure 41). The amplitude of the P3b was calculated by taking the mean amplitude from 250ms to 550ms following the onset of the stimulus for electrode Pz, which is the typical time frame of the P3b (e.g. Polich, 2007). The mean amplitudes were 1.7 (SEM = .8), -1.2 (SEM = 2.2), .3 (SEM = 1.5), and 1.2 (SEM = 1.3) for pre-surprise set size 1, pre-surprise set size 2, post-surprise set size 1, and post-surprise set size 2. Like the mean amplitudes of the CDA, there does not seem to be any significant effects of condition. In addition, visual inspection of the ERP in Figure 40 does not appear characteristic of a typical P3b.

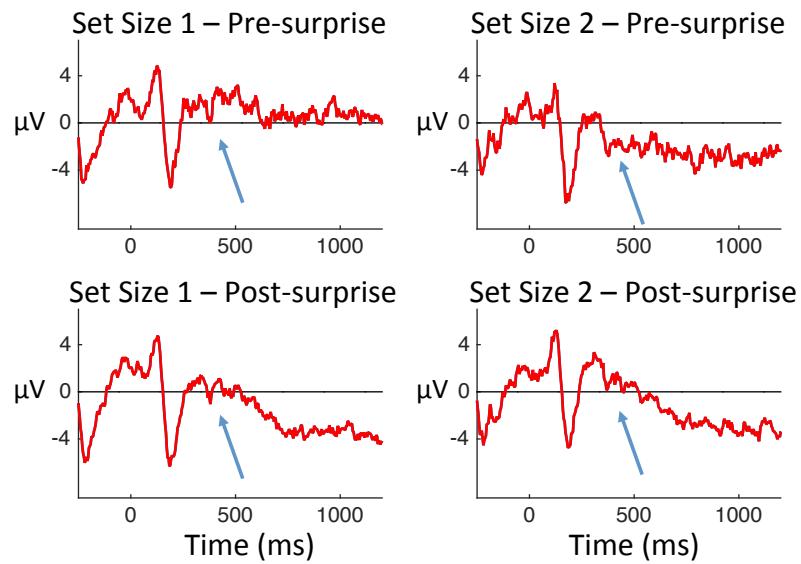


Figure 40. ERP time locked to the onset of the stimulus for electrode Pz to examine P3b. The arrow indicates where the P3b should be expected.

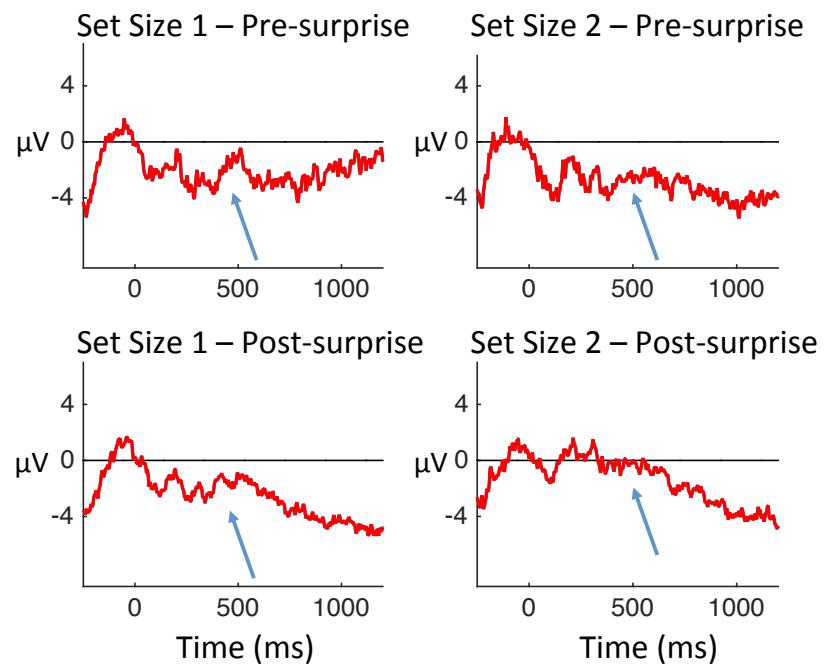


Figure 41. ERP time locked to the onset of the stimulus for electrode Fz to examine P3a. The arrow indicates where the P3b should be expected.

The amplitude of the P3a was calculated by taking the mean amplitude from 250ms to 550ms for electrode Fz. The mean amplitude were -2.4 (SEM = 1), -2.8 (SEM = 1.2), -1.8 (SEM = .9), and -.1 (1.2) for pre-surprise set size 1, pre-surprise set size 2, post-surprise set size 1, and post-surprise set size 2. Like the above findings with the CDA and P3b, there does not seem to be any significant effect of condition on the amplitudes and the visual inspection of Figure 41 does not resemble a typical P3a.

General Discussion

A pilot experiment was conducted to determine if EEG correlates associated with VWM would be affected by adding features to the memory load. This experiment utilizes a methodology similar to a previously published paper (Swan, Collins, & Wyble, 2016) that found a trade-off between the number of features and precision. In this experiment, participants were asked to report just the direction of colored arrows and halfway through the experiment, there was a surprise trial that asked participants to report the color. Following the surprise trial, either feature could be cued, which means that pre-surprise trials (1 relevant feature) can be compared to post-surprise trials (2 relevant features) to determine the effects of adding a feature to memory. In addition to the number of features, the set size was also manipulated in both pre-surprise and post-surprise trials.

The results of the experiment showed a clear set size effect on memory precision, such that participants were less precise when 2 objects compared to a single object. Interestingly, participants did not exhibit a trade-off in precision when the task irrelevant became relevant to the task. This result conflicts with previous results that have found a cost to adding a feature to memory (Swan Collins Wyble, 2016). Note, that this trade-off has been found when direction was the initially relevant feature (Experiment 2 in Swan, Collins, & Wyble, 2016) and for set sizes greater than 1 (e.g. Chapter 4, Experiment 1). One possibility for the lack of a trade-off between precision and number of relevant features is that participants had

difficulty ignoring the task irrelevant feature (Marshall & Bays, 2013) or task irrelevant objects (Vogel & Machizawa, 2004). Perhaps a simplification of the experiment, such as removing the set size manipulation, may help participants more precisely allocation featural attention. However, with so few participants, it is difficult to make any strong claims.

Perhaps the more interesting results from this experiment concern the ERPs. There did not seem to be a change in the amplitude of the CDA as a function of set size, which conflicts with the literature (Luria, Balaban, Awh, & Vogel, 2016) and predictions from the Binding Pool model. Most researchers do not typically include set size 1 and set size 2 manipulations, so it is difficult to determine if this is typical. One study that did include a set size 1 and set size 2 manipulation did find a CDA difference between set size 1 and set size 2 when the stimuli were similar (i.e. minimal differences between features) in color and shape, but not when the stimuli were dissimilar (Diamantopoulou, Poom, Klaver, & Talsma, 2011). It could be that with more trials, a difference in CDA amplitude between set size 1 and set size 2 may be more evident in this experiment.

In addition to not finding a CDA difference, characteristic ERPs for P3a and P3b were not observed. This is quite shocking when considering the wealth of evidence in support of the P3 components reflecting some part of working memory updating (for review, Polich, 2007). In the experiment presented here, one would expect that working memory updating would be necessary to encode the task relevant features of the multiple objects, especially since participants were able to do the task. When examining other similar EEG studies looking at VWM (see Figure 1 in Luria et al., 2016), most researchers typically only display difference waveforms. Upon closer inspection of the few studies that do display contralateral and ipsilateral waveforms, most also do not find ERPs that have the characteristics of the P3 components (e.g. Kuo, Stokes, & Nobre, 2012; Störmer, Li, Heekeren, & Lindenberger, 2013; Diamantopoulou et al., 2011). The lack of P3 components in the experiment described here and in other VWM papers has important implications about what cognitive processes the P3 components actually reflect.

One possibility is that the P3 components are a reflection of the disambiguation of the key feature when presented with stimuli. A key feature is the feature that defines the target relevant to the task (Botella, Barriopedro, & Suero, 2001; Remington & Folk, 2001). In the experiment described here, the key feature was the location of the objects, which was defined by the pre-cue that appears before the onset of the stimuli. The stimuli also contain the response features (i.e. color and direction), which were the features that were reported in the recall screen. The stimuli also contain the key feature (i.e. location), but since the pre-cue informs participants which stimuli are the targets, there is no uncertainty about which objects are task relevant when the stimuli appear. Therefore, the P3b should appear if time-locked to the onset of the pre-cue and not the onset of the stimuli. The waveform time locked to the cue for the P3/P4 electrode is displayed in Figure 42. Here, a P3b does seem to be present²² and, upon closer inspection, there appears to be a P3b in Diamantopoulou et al. (2011) as well. Given these observations, the P3a and P3b may simply be signal travelling from frontal to posterior regions that indicate which targets to bind the response feature to.

²² The mean amplitudes are 3.2 (SEM = 1.0), 5.1 (SEM = 1.5), 3.6 (SEM = 0.8), and 4.4 (SEM = 1.6) for pre-surprise set size 1, pre-surprise set size 2, post-surprise set size 1, and post-surprise set size 2 for the contralateral electrode from 250ms to 550ms. Like the above analyses, not much can be gleaned from these data with so few participants, but importantly, these data resemble P3b amplitudes much more than the same amplitudes when time locked to the stimuli onset.

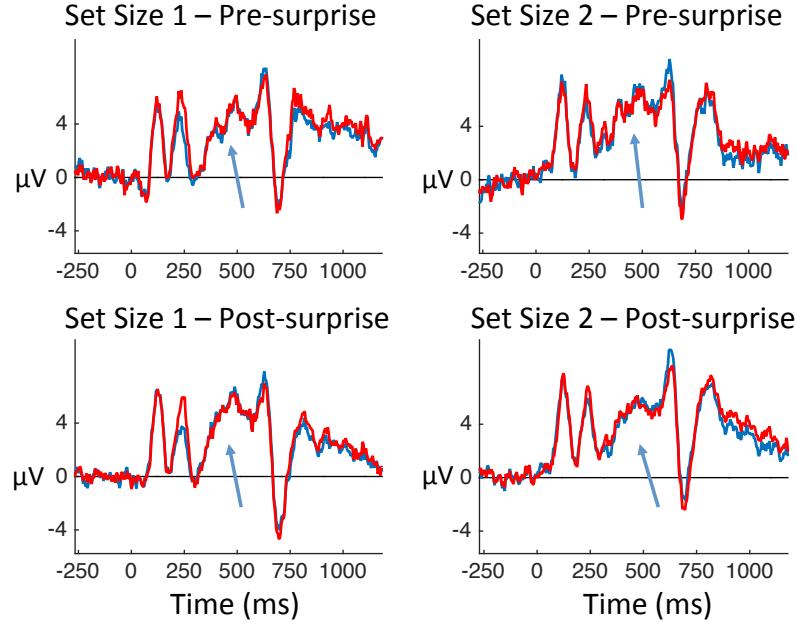


Figure 42. ERP time locked to the onset of the pre-cue for location for electrodes P3/P4, with both Contralateral (Blue) and Ipsilateral (Red) waveforms. The arrows indicate where the P3b would be expected.

This theory is consistent with other research where a P3 component is found time locked to the stimulus when the key and response features are the same, such as free recall of words (Karis, Fabiani, & Donchin, 1984). This theory is also consistent with research that finds a P3 component time locked to the stimulus when the key and response features are different, such as in oddball paradigms (Donchin, Ritter, & McCallum, 1978). In both of the above cases, a participant does not know whether a stimulus is a target or distractor until processing the key feature of the stimulus. In the experiment here, the pre-cue informs participants which stimuli are targets, so the stimuli themselves are not necessary in disambiguating which stimuli are targets or non-targets. In this sense, this theory remains consistent with the Binding Pool model if disambiguating targets from non-targets initiates the allocation of tokens. One way to test this hypothesis concerning the P3a and P3b would be to use the same paradigm described above, but include non-target distractors in the stimulus display. For example, if task irrelevant objects were also presented along with the stimuli, then there may be a P3 component when time-locked to the onset of the stimuli because participants would need to

disambiguate which stimuli are targets. An example of this would be Experiment 1 in Vogel, McCollough, and Machizawa (2005), where participants filtered out task irrelevant objects from the relevant visual field.

Conclusion

Here, the relationship between the ERP correlates of working memory and the memory load was explored. In a pilot experiment, participants were first consistently reporting the direction of either 1 or 2 colored arrows and then reported either the color or direction after a surprise trial. Behaviorally, there was a difference in precision across set sizes, but there was no trade-off in precision when comparing pre-surprise and post-surprise trials. Furthermore, the CDA component, which is presumed to reflect memory load, did not seem to be affected by either the load manipulation or the number of relevant feature manipulation. In addition, a P3a and P3b were not found when looking at the waveform time-locked to the onset of the stimuli. However, a P3b does appear when time-locked to the onset of the cue, which may indicate that participants were using the cue to start working memory processes before the stimuli even appeared. These initial results do not seem promising for this specific methodology, but additional participants may add further clarity to these results.

Chapter 7. Conclusions regarding the Binding Pool model

Visual working memory (VWM) describes the process where visual information is briefly maintained for later retrieval. Most research investigating VWM in the last few decades was conducted under the framework that information is maintained in individuated memory slots (Luck & Vogel, 1997). However, research investigating VWM recall have elucidated trade-offs between the number of objects that need to be remembered and the quality of those objects (Wilken & Ma, 2004; Bays et al., 2009). Specifically, this trade-off occurs even below memory capacity. This finding has resulted in the need for more sophisticated models of working memory to increase our understanding of the underlying mechanisms, which can help researchers better minimize memory errors in real life and connect VWM to other cognitive capabilities (Conway, Kane, & Engle, 2003).

This thesis has focused on testing specific predictions of a computational model of VWM called the Binding Pool model (Swan & Wyble, 2014). In order to determine if the Binding Pool model is an adequate description of how VWM works, several predictions of the Binding Pool model were tested. Correctly predicted behaviors validate the mechanisms of the model, while incorrectly predicted behaviors help to improve the model by indicating what aspects of the model are inaccurate. Therefore, the outcome of these tests is informative regardless of whether the model is supported or not.

Tested Predictions

In Chapter 3, two experiments were conducted to explore the Binding Pool model's prediction that the similarity of presented objects has differing effects on memory precision. The model predicts that memory precision when encoding two similar objects and two highly dissimilar is more precise than when encoding two

objects of intermediate similarity. This prediction was tested in two delayed estimation experiments where participants reported 1 of 2 presented colors. The colors were randomly selected and presented either simultaneously (Experiment 1) or sequentially (Experiment 2). In both experiments, participants exhibited patterns of data similar to the prediction of the model. That is, participants were more precise when the two colors were similar (i.e. within 36 degrees) and dissimilar (i.e. greater than 144 degrees) than when the two colors were at an intermediate level of similarity (between 36 and 144 degrees). This result has two implications. First, increased precision for similar objects replicate previous findings that more similar objects improves memory capacity (Lin & Lin, 2009; Sims, Jacobs, & Knill, 2012) and precision (Oberauer & Lin, 2017). Secondly, finding improved precision for highly dissimilar colors suggest that opponent colors or objects of high dissimilarity may be less susceptible to interference in memory. These findings support the prediction from the Binding Pool model.

In Chapter 4, two experiments were conducted to explore the Binding Pool model's prediction that encoding more features of an object comes at a cost to memory precision. In previous research (Swan, Collins, & Wyble, 2016), it was found that task irrelevant features were coarsely coded and that precision trade-offs with the number of relevant features. This trade-off in memory precision supports a prediction from the Binding Pool model, but it is important to determine if coarse coding and the trade-off in precision are general properties of VWM or if there is something unique about only encoding one object. In two experiments, the same procedure conducted in Swan, Collins, and Wyble was performed with set size 2 (Experiment 1) and set size 4 (Experiment 2). At set size 2 and set size 4, there was a cost to memory precision associated with making a task irrelevant feature relevant. In Experiment 1, the task irrelevant feature was more coarsely coded than when that same feature was relevant to the task. However, when four objects were presented in Experiment 2, participants were primarily guessing on the surprise trial. These results generally replicate the finding from Swan, Collins, and Wyble (2016) and the trade-off between the number of features and precision supports the prediction from the Binding Pool model.

In Chapter 5, two experiments were conducted to explore the Binding Pool model's prediction regarding the storage of repetitions into memory. The model predicts that retrieval precision is more precise when only a single object is presented compared to the presentation of repetitions. This prediction was tested using both simultaneous (Experiment 1) and sequential (Experiment 2) presentation of objects in two delayed estimation experiments. In both experiments, participants were more precise when reporting a color in the repetition conditions than the single color and multiple random colors conditions. This result was incorrectly predicted by the Binding Pool model and highlights a limitation of the model. In order for the model to simulate these results, there needs to be a change to the underlying mechanisms.

The data from these experiments provide new benchmarks for the Binding Pool model. The model was able to simulate higher precision for highly similar and dissimilar relative to intermediate levels of similarity and the cost in memory precision associated with adding features to the memory load. However, the model failed to predict better precision for repetitions than a single object and that task irrelevant features are coarsely coded. In order to simulate all of the above findings, the original Binding Pool model (Swan & Wyble, 2014) needs to be adjusted.

Changes To The Model

The first adjustment arises from the results of Swan, Collins, and Wyble (2016) and Chapter 4 regarding the coarse representation of task irrelevant features. The model correctly predicted that adding features to the memory load would come at a cost to memory precision because each additional feature constrains the number of binding pool nodes to represent the object. However, this assumed that the task irrelevant features were not being encoded. The experiments in Swan, Collins, and Wyble and Chapter 4 found that participants were encoding the task irrelevant features, albeit with coarse precision. Coarse coding could be implemented by increasing the width of the activated response distribution, while

keeping total amplitude the same. However, this change is not supported by data measuring how neurons in the visual cortex respond to relevance (McAdams & Maunsell, 1999). A more neurophysiological accurate change to the model is to adjust the strength, or height, of the activity, but that does not affect retrieval precision. Therefore, the model needs to be adjusted. One way to make this change is to alter how information is encoded into the model.

In the new version of the model, a memory is still the pattern of activity on a set of binding pool nodes connected to the active types and token. However, instead of projecting all of the types simultaneously, each type is sequentially²³ bound to a token. When coupling this change to encoding with the ability to strengthen or weaken the activity of a type layer corresponding to differing levels of attention to that feature driven, the model is able to simulate the effects from Chapter 4. These changes result in coarse retrieval because retrieval from a task irrelevant feature is more susceptible to interference from the distributed encoding of more activated relevant features. For example, a colored arrow is presented to the model. The color and location of the colored arrow are task relevant, but the direction of the arrow is task irrelevant. First, the type node corresponding to the location²⁴ of the arrow is highly activated and projected to the binding pool. Binding pool nodes connected to both the token and that location type layer are then activated. Next, the same process occurs for the color type layer. This results in some binding pool nodes activated from both projections, some from color type layer projections, some from location type layer projections, and other binding pool nodes not activate at all. Lastly, the type node corresponding to the direction of the arrow is minimally activated, reflecting task irrelevance, and is projected to the binding pool. This results in a diverse pattern of activity in the binding pool (Figure 28 in Chapter 5);

²³ Nothing in the data presented in this thesis suggests that participants are sequentially binding features to the token. In fact, theories about how features are integrated suggest that it happens in parallel (e.g. Treisman & Gelade, 1980). Future iterations of the Binding Pool model will focus on how to store information in parallel.

²⁴ In this example, location is coded first. However, there is no significance to the order of which features are encoded.

some nodes minimally activated that received activity from the direction type layer, some nodes highly activated that received activity from all type layers, and many other combinations. Task irrelevant features are retrieved with coarse precision because binding pool nodes coding for the task irrelevant feature are less activated than other activated nodes. The trade-off in precision from making the task irrelevant feature relevant arises from increased noise as a result of increased activity in the binding pool.

The second change to the model is in response to participants reporting repetitions more precisely than a single object. The second change also involves a change to the encoding of information in the binding pool. In the original version of the model, two repeated objects (i.e. two color patches with the same color) were treated like any two objects; each copy was bound to its own token. In two experiments (Chapter 5), it was found that repetitions were more precisely reported than a single object. In order for the model to account for these data, there are a couple of potential adjustments. First, one change that improved precision for repetitions is to bind the second object to the same token as the first object if it is a repetition. One consequence of binding two objects to the same token is increased binding errors for non-repeated features. For example, if the model binds two arrows with the same color but different orientations to the same token, the model makes substantially more retrieval errors when retrieving the direction of one of the arrows compared to simulations where the objects have separate tokens. This could be tested by having participants report either the color or direction of one of two presented colored arrows and then looking at responses as a function of the similarity of the non-reported feature (see discussion Chapter 5). However, one limitation of this approach is that it assumes that objects are bound to the same token if any single feature is repeated between two objects, which may not be the case.

Another consequence of this implementation is that participants would experience repetition blindness given that only a single token would be activated despite being presented with repetitions. Repetition blindness is the failure to recognize the repetitions of information as separate events, despite the multiple

presentations of that information (Kanwisher, 1987). For example, in Experiment 2 in Chapter 5, participants were sequentially presented with 2 color patches that could be exact repetitions. On some of the repetition trials, participants may fail to notice that a second repetition was presented. This could be tested in a surprise trial that asks participants to report the number of objects that were presented. Specifically, this surprise trial would occur on a repetition trial. If it is the case that participants fail to recognize that two objects were presented, then it provides evidence for the model's implementation of binding multiple objects to the same token.

A second option for simulating increased precision for repetitions relative to a single object is to increase activation during encoding when presented with a repetition. This would be akin to participants getting a second opportunity to more precisely encode the information. For this to be the case, participants would need to be aware²⁵ of the repetition. In the experiment proposed in the previous paragraph, participants would see repetitions and then unexpectedly be asked how many objects appeared. If it is the case that participants notice the repetition and use the second presentation to strengthen the memory for that object, then participants should be accurate in determining how many objects were presented.

New Predictions

Besides testing predictions, another important aspect in determining the quality of a model is to continue developing new predictions. In this thesis, the Binding Pool model was used to generate a prediction regarding event-related potentials (ERPs) that are thought to reflect different VWM processes. The P3a and P3b components are believed to reflect processes that update the contents of working memory (Polich, 2007). On the other hand, the contralateral delay

²⁵ The reason the participant would need to be aware is that the mechanism for increasing the activation for the repetition is the same mechanism that decreases the activation of task irrelevant features. And importantly, there would still be multiple activated tokens.

activation (CDA) is thought to reflect the amount of objects that are being maintained during a retention period (Luria et al., 2016). Relating these components has not previously been done using the same VWM paradigm, which provides an opportunity for the Binding Pool model to make predictions about how these components may be related to each other. Though the Binding Pool model does not have the ability to simulate the cascade of electrophysiological events to fully generate an EEG waveform, there are discrete memory stages that have different states of activity. For example, tokens are activated during encoding, but do not have any activity during maintenance. If a component were a reflection of token activity, it would be present during the encoding of the object, but not during maintenance. The model predicts that the P3a, which is frontal positivity, is related to the allocation of tokens when presented with multiple objects. The P3b and CDA correspond to the amount of information present during encoding in the type layers and in the binding pool, respectively. Therefore, changes in the number of relevant features should affect the P3b and CDA, but not the P3a.

This prediction was tested in a pilot EEG experiment that resembled the methodology used in Swan, Collins, and Wyble (2016). Participants were instructed to remember the direction of either 1 or 2 colored arrows and participants reported the direction consistently. In the second part of the experiment, the color of the arrow became relevant and participants needed to remember both features. The results of the experiment were mixed. With regards to memory precision, participants were less precise for set size 2 than set size 1, but there was not a trade-off between precision and the number of relevant features. With regards to the ERP analysis, the CDA had relatively the same amplitude across the different conditions, despite the number of objects and number of relevant feature varying. Interestingly, the P3 components were not prominent when the ERP was time locked to the onset of the stimuli. However, when time locked to the location pre-cue, there were clearer P3 components. This suggests that the P3 components might not reflect working memory encoding for the object itself, but possibly the encoding of the feature that differentiates the target from non-targets. In this experiment, location is the defining feature, so the cue before the stimuli provided the

disambiguation. Collecting more participant data may elucidate some of these mixed findings.

General Discussion

The conclusions from the results of the predictions tested in this thesis have important implications with regards to how visual working memory works. When remembering two objects, there is systematic interference between the two objects in memory. This interference, however, is not simply a worsening as the two objects become less similar. Instead, the pattern of interference is more complex, suggesting multiple mechanisms at play.

First, there is better precision when the relevant feature of the two objects is highly similar. One potential mechanism is that similar objects are more easily grouped together (Brady et al., 2009), which may minimize interference during retrieval and enable the storage of more information. Secondly, there is better precision when the relevant feature of the two objects is highly dissimilar relative to intermediate levels of similarity. One explanation is that highly dissimilar objects have less overlapping connections, thus interfere less in storage. These results support the prediction from the Binding Pool model (Swan & Wyble, 2014). Theories of memory that store information in the feature layers would have difficulty explaining these data (Franconeri et al., 2013) because more similar representations would result in greater levels of interference.

In addition to finding better precision when two objects are presented with similar features, there is also increased precision when repetitions are presented. In Chapter 5, two delayed estimation experiments were conducted that found participants could report a repetition more precisely than a single object. In the Binding Pool model, this benefit does not arise simply from presenting two copies to the model. In the model, there must be some other mechanism driving this increased precision for repetitions. One adjustment is that repetitions are bound to the same token, which is akin to the grouping mechanism when presented with

similar objects. A second adjustment could be increased activation when encoding the second copy as if participants recognize the repetition and strengthen the encoding process. These different hypotheses could be differentiated by surprisingly asking participants to report how many objects were presented when presented with repetitions. A failure to individuate the repetitions may indicate that participants were binding the repetitions to the same token. However, being able to successfully report that multiple repetitions were presented may suggest a different encoding strategy.

In the experiments conducted in Chapter 4, it was also found that visual features of different objects also interact in memory. For example, in Experiment 1 in Chapter 4, participants consistently reported the color of two colored arrows. In a surprise test, participants were surprisingly asked to report the direction of the arrow. In trials following the surprise trial, either feature (color or direction) could be cued to recall. Like in Swan, Collins, and Wyble (2016), there was a cost to memory precision of the initially relevant feature when adding the task irrelevant feature to the memory load. This result suggests that the memory resources encoding color are the same or at least partially overlapping resources used to encode direction, which support models of memory that utilize a single memory pool for objects and features (Swan & Wyble, 2014; Matthey, Bays, & Dayan, 2015). However, models that store objects in separate feature maps (e.g. Schneegans & Bays, 2017) would have difficulty explaining this trade-off in precision as a function of adding features to the memory load.

When attending to an object, it is not just the relevant features that are encoded into memory. Results from Chapter 4 and Swan, Collins, and Wyble (2016) suggest that task irrelevant features are also encoded in VWM tasks. For example, participants in Experiment 1 in Chapter 4 consistently reported the color of two color arrows. In a surprise trial, participants were asked to report the direction of one of the arrows and their response was coarse relative to when that feature was relevant to the task. One of the implications of this is that other task irrelevant features (e.g. size, shape, texture) that were not measured in Chapter 4 and in Swan, Collins, and Wyble (2016) were likely also coarsely coded. This suggests that trade-

offs between memory quality and the number of features are not about adding features to the memory load, but instead changing task irrelevant features to task relevant. In other words, while task relevant features interact in memory, task irrelevant features interact weakly. Thus, the number of task irrelevant features should then have minimal affect on memory precision for task relevant features. In a simulation of the Binding Pool model where the number of task irrelevant features was increased (Figure 43), precision of the task relevant feature was minimally affected by increasing numbers of task irrelevant features. For example, the model is able to simulate the encoding of a color patch's shape, size, texture, and many other task irrelevant features while still precisely remembering the task relevant color of the color patch.

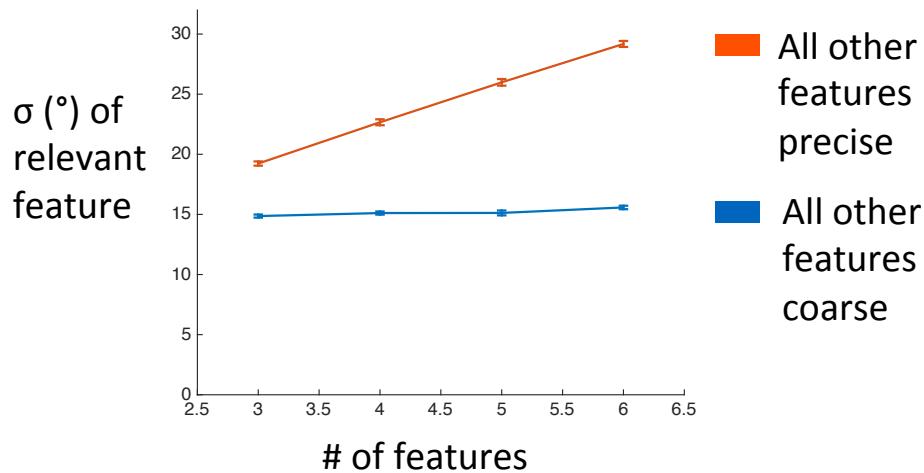


Figure 43. Adding features to memory load. Error bars are standard error.

Expanding The Binding Pool Model

In this thesis, it was shown that when participants are remembering multiple objects (e.g. colored arrows), the number of objects and also the number of relevant features interact in memory and affect memory precision. However, participants can also remember other types of information, such as faces and scenes. It is unlikely that these other stimuli are represented in memory as simple features, such as a scene being broken down into many individual objects and features. It is thus

important to consider how more complex stimuli may be encoded into VWM (Orhan & Jacobs, 2014). Vanrullen (2009) theorizes that there are two types of bindings in memory. First, there are hardwired bindings that are formed through experiences throughout life. These hardwired bindings start as lower level features, such as orientation, color, and motion. Higher-level representations, such as faces or scenes, are conjunctions of these lower level features that become their own representations with experience. For example, Japanese characters (e.g. kanji) to individuals who cannot read Japanese may represent the character with many lines, whereas someone who can read Japanese may have a single representation of that character. These hardwired bindings are akin to the type layers in the Binding Pool model.

In addition to hardwired bindings, there are also on-demand bindings. These bindings are mediated by attention to form transient representations of previously unknown or novel feature conjunctions. When remembering a single object, the result of these on-demand bindings is a single memory representation. This process is highly similar to how multiple type representations are combined in the binding pool using a token to individuate different objects. Therefore, by using more complex type layers, the Binding Pool model could be used to generate simulations of memories for scenes, faces, or other complex stimuli. The Binding Pool model provides a computational framework for integrating features within a level (e.g. color and orientation) and across levels (e.g. color and a face) into a memory in a single memory pool.

This framework may provide an explanation of other memory phenomena as well. For example, inaccuracies in eyewitness testimony (review Wells & Olson, 2003) may result from the coarse coding of task irrelevant features. One technique that helps accuracy in eyewitness testimony is to go through a Cognitive Interview (Wells, Memon, & Penrod, 2006), which is a procedure that an investigator uses when interacting with a witness. Part of this procedure is extensive retrieval, which involves retrieving as much information about the scene as possible from different perspectives and in different chronological orders. This is akin to using multiple

retrieval cues in a VWM task, which has shown to improve precision in participants and in the Binding Pool model (Rajsic, Swan, Wilson, & Pratt, 2017).

Another possible phenomenon that the Binding Pool model may provide insight is effects of memory on aging. Aging typically coincides with difficulties in forming new episodic memories (Wilson, Gallagher, Eichenbaum, & Tanila, 2006) and reduced memory precision (Peich, Husain, & Bays, 2013). As a function of aging, there is evidence that the number of synapses declines (Hedden & Gabrieli, 2004) instead of the number of neurons declining (Burke & Barnes, 2006). In the Binding Pool model, reduced synapses is akin to reduced connections between types and tokens to the binding pool, which reduces memory precision. When the connection weights are decreased, there are fewer active binding pool nodes representing the memory and thereby worse precision. This suggests that memory loss associated with the loss of synapses as a function of aging occurs during encoding instead of retrieval, which has been found in some source memory tasks (e.g. Glisky, Rubin, & Davidson, 2001). Thus, the Binding Pool model provides a mechanistic explanation of memory loss as a function of aging.

Conclusion

The Binding Pool model is a simulation of how visual information is bound together to form a memory representation. Three predictions of this model were tested, setting new behavioral constraints on the scope of the Binding Pool model. Several of the predictions resulted in changes to the model. Specifically, the underlying mechanisms of how visual features are bound to a token were changed to enable the model to simulate varying degrees of encoding strength. In addition, different implementations were proposed to enable the model to encode repetitions more precisely than a single object. The contribution of this work is two-fold. First, these new data not only set benchmarks for the Binding Pool model, but also other models of memory. Secondly, these data have resulted in adaptations to the Binding Pool model to make it a more general model of memory.

References

- Akyürek, E. G., Leszczyński, M., & Schubö, A. (2010). The temporal locus of the interaction between working memory consolidation and the attentional blink. *Psychophysiology*, 47(6), 1134-1141.
- Alvarez, G. A., & Cavanagh, P. (2004). The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological Science*. 15(2). 106-111
- Awh, E., Barton, B., & Vogel, E. K. (2007). Visual working memory represents a fixed number of items regardless of complexity. *Psychological Science*. 18(7). 622-628
- Awh, E., Vogel, E. K., & Oh, S. H. (2006). Interactions between attention and working memory. *Neuroscience*, 139(1), 201-208.
- Baddeley, A. D., & Hitch, G. J. (1974). Working memory. In G. H. Bower (Ed.), Recent advances in learning and motivation Vol. VIII, (pp. 47-90). New York: Academic Press.
- Bae, G. Y., Olkkonen, M., Allred, S. R., Wilson, C., & Flombaum, J. I. (2014). Stimulus-specific variability in color working memory with delayed estimation. *Journal of Vision*, 14(4), 7-7.
- Bays, P. M., Catalao, R. F. G., & Husain, M. (2009). The precision of visual working memory is set by allocation of a shared resource. *Journal of Vision*. 9(10). 1-11
- Bays, P. M., Gorgoraptis, N., Wee, N., Marshall, L., & Husain, M. (2011). Temporal dynamics of encoding, storage, and reallocation of visual working memory. *Journal of vision*, 11(10), 6-6.
- Bays, P. M., Wu, E. Y., & Husain, M. (2011). Storage and binding of object features in visual working memory. *Neuropsychologia*, 49(6), 1622-1631.
- Berti, S., Geissler, H. G., Lachmann, T., & Mecklinger, A. (2000). Event-related brain potentials dissociate visual working memory processes under categorial and identical comparison conditions. *Cognitive Brain Research*, 9(2), 147-155.
- Botella, J., Barriopedro, M., & Suero, M. (2001). A model of the formation of illusory conjunctions in the time domain. *Journal of Experimental Psychology: Human Perception and Performance*, 27(6), 1452.
- Bowman, H., & Wyble, B. (2007). The simultaneous type, serial token model of temporal attention and working memory. *Psychological Review*. 114(1). 38-70
- Brady, T. F., & Alvarez, G. A. (2011). Hierarchical encoding in visual working memory: ensemble statistics bias memory for individual items. *Psychological*

Science, 22(3). 384-92

Brady, T. F., & Alvarez, G. A. (2015). Contextual effects in visual working memory reveal hierarchically structured memory representations. *Journal of vision*, 15(15), 6-6.

Brady, T. F., Konkle, T., & Alvarez, G. A. (2009). Compression in visual working memory: using statistical regularities to form more efficient memory representations. *Journal of Experimental Psychology: General*, 138(4), 487.

Brady, T. F., & Tenenbaum, J. B. (2013). A probabilistic model of visual working memory: incorporating higher order regularities into working memory capacity estimates. *Psychological Review*, 120(1). 85-109

Brainard, D. H. (1997). The psychophysics toolbox. *Spatial vision*, 10, 433-436.

Burke, S. N., & Barnes, C. A. (2006). Neural plasticity in the ageing brain. *Nature Reviews Neuroscience*, 7(1), 30-40.

Chen, H., & Wyble, B. (2015). Amnesia for Object Attributes Failure to Report Attended Information That Had Just Reached Conscious Awareness. *Psychological science*, 0956797614560648.

Chen, H., Swan, G., & Wyble, B. (2016). Prolonged focal attention without binding: Tracking a ball for half a minute without remembering its color. *Cognition*, 147, 144-148.

Conway, A. R., Kane, M. J., & Engle, R. W. (2003). Working memory capacity and its relation to general intelligence. *Trends in cognitive sciences*, 7(12), 547-552.

Cowan, N. (2001). The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behavioral and Brain Science*, 24, 87-185

Cowan, N., Elliott, E. M., Saults, J. S., Morey, C. C., Mattox, S., Hismajatullina, A., & Conway, A. R. (2005). On the capacity of attention: Its estimation and its role in working memory and cognitive aptitudes. *Cognitive psychology*, 51(1), 42-100.

Dell'Acqua, R., Dux, P. E., Wyble, B., Doro, M., Sessa, P., Meconi, F., & Jolicœur, P. (2015). The attentional blink impairs detection and delays encoding of visual information: Evidence from human electrophysiology. *Journal of cognitive neuroscience*.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*, 134(1), 9-21.

- Diamantopoulou, S., Poom, L., Klaver, P., & Talsma, D. (2011). Visual working memory capacity and stimulus categories: a behavioral and electrophysiological investigation. *Experimental brain research*, 209(4), 501-513.
- Donchin, E., Ritter, W., & McCallum, W. C. (1978). Cognitive psychophysiology: The endogenous components of the ERP. *Event-related brain potentials in man*, 349-411.
- Donkin, C., Kary, A., Tahir, F., & Taylor, R. (2016). Resources masquerading as slots: Flexible allocation of visual working memory. *Cognitive psychology*, 85, 30-42.
- Eitam, B., Yeshurun, Y., & Hassan, K. (2013). Blinded by irrelevance: Pure irrelevance induced “blindness”. *Journal of experimental psychology: human perception and performance*, 39(3), 611.
- Eitam, B., Shoval, R., & Yeshurun, Y. (2015). Seeing without knowing: task relevance dissociates between visual awareness and recognition. *Annals of the New York Academy of Sciences*, 1339(1), 125-137.
- Fougnie, D., & Alvarez, G. A. (2011). Object features fail independently in visual working memory: Evidence for a probabilistic feature-store model. *Journal of vision*, 11(12), 3-3.
- Fougnie, D., Cormiea, S. M., Kanabar, A., & Alvarez, G. A. (2016). Strategic trade-offs between quantity and quality in working memory.
- Fougnie, D., Suchow, J. W., & Alvarez, G. A. (2012). Variability in the quality of working memory. *Nature Communication*, 3, 1229
- Franconeri, S. L., Alvarez, G. A., & Cavanagh, P. (2013). Flexible cognitive resources: competitive content maps for attention and memory. *Trends in cognitive sciences*, 17(3), 134-141.
- Gao, Z., Li, J., Liang, J., Chen, H., Yin, J., & Shen, M. (2009). Storing fine detailed information in visual working memory—Evidence from event-related potentials. *Journal of Vision*, 9(7), 17.
- Glisky, E. L., Rubin, S. R., & Davidson, P. S. (2001). Source memory in older adults: an encoding or retrieval problem?. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(5), 1131.
- Gobet, F., & Clarkson, G. (2004). Chunks in expert memory: Evidence for the magical number four... or is it two?. *Memory*, 12(6), 732-747.
- Gorgoraptis, N., Catalao, R. F., Bays, P. M., & Husain, M. (2011). Dynamic updating of working memory resources for visual objects. *The Journal of Neuroscience*, 31(23), 8502-8511.

- Hedden, T., & Gabrieli, J. D. (2004). Insights into the ageing mind: a view from cognitive neuroscience. *Nature reviews neuroscience*, 5(2), 87-96.
- Huang, J., & Sekuler, R. (2010). Distortions in recall from visual memory: two classes of attractors at work. *Journal of Vision*. 10, 1-27
- Ikkai, A., McCollough, A. W., & Vogel, E. K. (2010). Contralateral delay activity provides a neural measure of the number of representations in visual working memory. *Journal of neurophysiology*, 103(4), 1963-1968.
- Jackendoff, R. (2002). Foundation of language-brain, meaning, grammar, evolution. NY: Oxford University Press.-2002.-247 p.
- Kahneman, D., Treisman, A., & Gibbs, B. J. (1992). The reviewing of object files: Object-specific integration of information. *Cognitive psychology*, 24(2), 175-219.
- Kang, M. S., & Woodman, G. F. (2014). The neurophysiological index of visual working memory maintenance is not due to load dependent eye movements. *Neuropsychologia*, 56, 63-72.
- Kanwisher, N. G. (1987). Repetition blindness: type recognition without token individuation. *Cognition*. 27, 117-143
- Kanwisher, N., Driver, J., & Machado, L. (1995). Spatial repetition blindness is modulated by selective attention to color or shape. *Cognitive psychology*, 29(3), 303-337.
- Karis, D., Fabiani, M., & Donchin, E. (1984). "P300" and memory: Individual differences in the von Restorff effect. *Cognitive Psychology*, 16(2), 177-216.
- Keshvari, S., van den Berg, R., & Ma, W. J. (2013). No evidence for an item limit in change detection. PLoS Computational Biology, 9(2). e1002927
- Kuo, B. C., Stokes, M. G., & Nobre, A. C. (2012). Attention modulates maintenance of representations in visual short-term memory. *Journal of cognitive neuroscience*, 24(1), 51-60.
- Lawrence, M. A. (2010). Estimating the probability and fidelity of memory. *Behavior research methods*, 42(4), 957-968.
- Lin, P., & Luck, S. J. (2009). The influence of similarity on visual working memory representations. *Visual Cognition*. 17(3). 1-15
- Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*. 390(6657). 279-281

- Luria, R., Balaban, H., Awh, E., & Vogel, E. K. (2016). The contralateral delay activity as a neural measure of visual working memory. *Neuroscience & Biobehavioral Reviews*.
- Luria, R., Sessa, P., Gotler, A., Jolicœur, P., & Dell'Acqua, R. (2010). Visual short-term memory capacity for simple and complex objects. *Journal of Cognitive Neuroscience*, 22(3), 496-512.
- Luria, R., & Vogel, E. K. (2011). Shape and color conjunction stimuli are represented as bound objects in visual working memory. *Neuropsychologia*, 49(6), 1632-1639.
- Ma, W. J., Husain, M., & Bays, P. M. (2014). Changing concepts of working memory. *Nature neuroscience*, 17(3), 347-356.
- Marr, D. (1976). Early processing of visual information. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 275(942), 483-519.
- Marshall, L., & Bays, P. M. (2013). Obligatory encoding of task-irrelevant features depletes working memory resources. *Journal of vision*, 13(2), 21-21.
- Martens, S., & Wyble, B. (2010). The attentional blink: Past, present, and future of a blind spot in perceptual awareness. *Neuroscience & Biobehavioral Reviews*, 34(6), 947-957.
- Matthey, L., Bays, P. M., & Dayan, P. (2015). A probabilistic palimpsest model of visual short-term memory. *PLoS Comput Biol*, 11(1), e1004003.
- McAdams, C. J., & Maunsell, J. H. (1999). Effects of attention on orientation-tuning functions of single neurons in macaque cortical area V4. *The Journal of Neuroscience*, 19(1), 431-441.
- McClelland, J. L., & Rumelhart, D. E. (1988). *Explorations in parallel distributed processing*. Cambridge: The MIT Press.
- McCollough, A. W., Machizawa, M. G., & Vogel, E. K. (2007). Electrophysiological measures of maintaining representations in visual working memory. *Cortex*, 43(1), 77-94.
- 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.
- Murdock, B. B. (1993). TODAM2: A model for the storage and retrieval of item, associative, and serial-order information. *Psychological Review*, 100(2), 183-203.
- Oberauer, K. (2002). Access to information in working memory: exploring the focus

of attention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(3), 411.

Oberauer, K., & Eichenberger, S. (2013). Visual working memory declines when more features must be remembered for each object. *Memory and Cognition*, May 2013

Oberauer, K., & Lin, H. Y. (2017). An interference model of visual working memory. *Psychological Review*, 124(1), 21.

Olson, I. R., & Jiang, Y. (2002). Is visual short-term memory object based? Rejection of the “strong-object” hypothesis. *Attention, Perception, & Psychophysics*, 64(7), 1055-1067.

Orhan, A. E., & Jacobs, R. A. (2014). Toward ecologically realistic theories in visual short-term memory research. *Attention, Perception, & Psychophysics*, 76(7), 2158-2170.

Pashler, H. (1988). Familiarity and visual change detection. *Perception & psychophysics*, 44(4), 369-378.

Peich, M. C., Husain, M., & Bays, P. M. (2013). Age-related decline of precision and binding in visual working memory. *Psychology and aging*, 28(3), 729.

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial vision*, 10(4), 437-442.

Phillips, W. A. (1974). On the distinction between sensory storage and short-term visual memory. *Perception & Psychophysics*, 16(2), 283-290.

Polich, J. (2007). Updating P300: an integrative theory of P3a and P3b. *Clinical neurophysiology*, 118(10), 2128-2148.

Prinzmetal, W., Amiri, H., Allen, K., & Edwards, T. (1998). Phenomenology of attention: 1. Color, location, orientation, and spatial frequency. *Journal of Experimental Psychology: Human Perception and Performance*. 24(1). 261-282

Rajcic, J., Swan, G., Wilson, D. E., & Pratt, J. (2017). Accessibility Limits Recall From Visual Working Memory. *Journal of experimental psychology. Learning, memory, and cognition*.

Remington, R. W., & Folk, C. L. (2001). A dissociation between attention and selection. *Psychological Science*, 12(6), 511-515.

Rouder, J. N., Morey, R. D., Cowan, N., Zwilling, C. E., Morey, C. C., & Pratte, M. S. (2008). An assessment of fixed-capacity models of visual working memory. *Proceedings of the National Academy of Sciences*, 105(16), 5975-5979.

- Salmela, V. R., Moisala, M., & Alho, K. (2014). Working memory resources are shared across sensory modalities. *Attention, Perception, & Psychophysics*, 76(7), 1962-1974.
- Schneegans, S., & Bays, P. M. (2016). No fixed item limit in visuospatial working memory. *cortex*, 83, 181-193.
- Schneegans, S., & Bays, P. M. (2017). Neural architecture for feature binding in visual working memory. *Journal of Neuroscience*, 37(14), 3913-3925.
- Shin, H., & Ma, W. J. (2016). Crowdsourced single-trial probes of visual working memory for irrelevant features. *Journal of vision*, 16(5), 10-10.
- Sims, C., Jacobs, R. A., & Knill, D. C. (2012). An ideal observer analysis of visual working memory. *Psychological Review*, 119(4), 807-30
- Störmer, V. S., Li, S. C., Heekeren, H. R., & Lindenberger, U. (2013). Normative shifts of cortical mechanisms of encoding contribute to adult age differences in visual-spatial working memory. *Neuroimage*, 73, 167-175.
- Suchow, J. W., Fougnie, D., Brady, T. F., & Alvarez, G. A. (2014). Terms of the debate on the format and structure of visual memory. *Attention, Perception, & Psychophysics*, 76(7), 2071-2079.
- Swan, G., Collins, J., & Wyble, B. (2016). Memory for a single object has differently variable precisions for relevant and irrelevant features. *Journal of Vision*, 16(3), 32-32.
- Swan, G., & Wyble, B. (2014). The binding pool: A model of shared neural resources for distinct items in visual working memory. *Attention, Perception, & Psychophysics*, 76(7), 2136-2157.
- Swan, G., Wyble, B., & Chen, H. (2017). Working memory representations persist in the face of unexpected task alterations. *Attention, Perception, & Psychophysics*, 1-7.
- Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, 12(1), 97-136.
- Van den Berg, R., & Ma, W. J. (2014). "Plateau"-related summary statistics are uninformative for comparing working memory models. *Attention, Perception, & Psychophysics*, 76(7), 2117-2135.
- van den Berg, R., Shin, H., Chou, W., George, R., & Ma, W. J. (2012). Variability in encoding precision accounts for visual short-term memory limitations. *Proceedings of the National Academy of Sciences of the United States of America*, 109(22), 8780-8785

- van den Berg, R., Awh, E., & Ma, W. J. (2014). Factorial comparison of working memory models. *Psychological review*, 121(1), 124.
- Vanrullen, R. (2009). Binding hardwired versus on-demand feature conjunctions. *Visual Cognition*, 17(1-2), 103-119.
- Vogel, E. K., Luck, S. J., & Shapiro, K. L. (1998). Electrophysiological evidence for a postperceptual locus of suppression during the attentional blink. *Journal of Experimental Psychology: Human Perception and Performance*, 24(6), 1656.
- Vogel, E. K., & Machizawa, M. G. (2004). Neural activity predicts individual differences in visual working memory capacity. *Nature*, 428(6984), 748-751.
- Vogel, E. K., McCollough, A. W., & Machizawa, M. G. (2005). Neural measures reveal individual differences in controlling access to working memory. *Nature*, 438(7067), 500-503.
- Vogel, E. K., Woodman, G. F., & Luck, S. J. (2001). Storage of features, conjunctions, and objects in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1), 92.
- Vogel, E. K., Woodman, G. F., & Luck, S. J. (2006). The time course of consolidation in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, 32(6), 1436.
- Vogel, E. K., Luck, S. J., & Shapiro, K. L. (1998). Electrophysiological evidence for a postperceptual locus of suppression during the attentional blink. *Journal of Experimental Psychology: Human Perception and Performance*, 24(6), 1656.
- Wei, W., Wang, X. J., & Wang, D. (2012). From Distributed resources to limited slots in multiple-item working memory: a spiking network model with normalization. *Journal of Neuroscience*, 32, 11228-11240.
- Wells, G. L., Memon, A., & Penrod, S. D. (2006). Eyewitness evidence: Improving its probative value. *Psychological Science in the Public Interest*, 7(2), 45-75.
- Wells, G. L., & Olson, E. A. (2003). Eyewitness testimony. *Annual Review of Psychology*, 54(1), 277-295.
- Wheeler, M. E., & Treisman, A. M. (2002). Binding in short-term visual memory. *Journal of Experimental Psychology: General*, 131(1), 48.
- Wilken, P., & Ma, W. J. (2004). A detection theory account of change detection. *Journal of Vision*. 4, 1120-1135

- Wilson, I. A., Gallagher, M., Eichenbaum, H., & Tanila, H. (2006). Neurocognitive aging: prior memories hinder new hippocampal encoding. *Trends in neurosciences*, 29(12), 662-670.
- Woodman, G. F., & Vogel, E. K. (2008). Selective storage and maintenance of an object's features in visual working memory. *Psychonomic bulletin & review*, 15(1), 223-229.
- Wyble, B., Bowman, H., & Nieuwenstein, M. (2009). The attentional blink provides episodic distinctiveness: sparing at a cost. *Journal of Experimental Psychology: Human Perception and Performance*. 35(3). 787-807
- Xu, Y. (2010). The neural fate of task-irrelevant features in object-based processing. *The Journal of Neuroscience*, 30(42), 14020-14028.
- Zhang, W., & Luck, S. J. (2008). Discrete fixed-resolution representations in visual working memory. *Nature*. 452(7192). 233-235

Appendix A

Binding Pool Model Matlab Code

Contents

- [parameters](#)
 - [input into model](#)
 - [connectivity](#)
 - [storage](#)
 - [retrieval](#)
-

```
function RunModel
```

```
% Code that runs Binding Pool model (Swan & Wyble, 2014)

% this path includes the code to run the mixture model (Bays et al., 2009)
% See Paul Bays's website for more details
addpath(genpath('jv10'))
```

parameters

```
type_size = 10; % number of type nodes
num_layers = 2; % number of type layers

bp_size = 800; % number of binding pool nodes
type_conn = [.45 .45]; % proportion of type weights that are active
token_conn = .45; % proportion of token weights that are active
token_size = 8; % number of token nodes
cue_layer = 2; % which type layer is the cue
response_layer = 1; % which type layer is the response layer
token_individuation = .016;
capacity = [2:7]; % number of available tokens on a given trial
type_overlap = [.3 .05]; % overlap between neighboring type nodes

act_peak = [1 1]; % column == num_layers

% task parameters
num_items = [1 2 4 6];
num_trials = 2000;

load_count = 0;
for which_load = num_items
    load_count = load_count + 1;

    reported = zeros(num_trials,which_load);
    presented = zeros(num_trials,num_layers,which_load);
    token = zeros(token_size,1);

    for trial = 1:num_trials
```

input into model

```

presented_items = ceil(rand(num_layers,which_load)*360);
presented(trial,:,:)= presented_items;

input = zeros(num_layers,type_size,which_load);
input = Type_activation(presented_items,type_size,act_peak,input);

```

connectivity

```

if mod(trial,ceil(num_trials*.01))==1
    model = Setupconn(type_conn,token_conn,bp_size,type_overlap,type_size,num_layers,t
oken_size);
    end

type_connectivity = model.type_weight;
token_connectivity = model.token_weight;

```

storage

```

bp_nodes = zeros(1,bp_size);

[null,cindex] = sort(rand(size(capacity)));

for which_item = 1:which_load
    if which_item <= capacity(cindex(1))

        token = zeros(1,token_size);
        token(1,which_item) = 1;

        a = 1;

        for layer = 1:num_layers
            a = a.* (input(layer,:,: ,which_item)*type_connectivity{layer}(:,:,1));
        end

        bp_nodes = bp_nodes + a.* (token * token_connectivity(:,:,1));

    end
end

bp_nodes = bp_nodes./sum(bp_nodes);

```

retrieval

```

for which_item = 1:which_load

    c = 1;
    for cue_type = cue_layer
        if cue_type ~= response_layer
            c = c.* (input(cue_type,:,: ,which_item)*type_connectivity{cue_type}(:,:,2));
        end
    end

```

```

which_tok = c.*bp_nodes;

retrieved_token_layer = token_connectivity(:,:,2)*transpose(which_tok);

sorted_list = sort(retrieved_token_layer, 'descend');

if abs(diff(sorted_list(1:2))) < token_individualuation
    % when the difference between the two most activated token
    % nodes is below the token individualuation parameter, the
    % model guesses

    retrieved_token = 0;
    reported(trial,which_item) = (rand*2*pi);
    retrieved_length(trial,which_item) = 0;

else % correct retrieval
    [act retrieved_token] = max(retrieved_token_layer);

    token = zeros(1,token_size);
    token(1,retrieved_token) = 1;

    e = 1;
    for cue_type = cue_layer
        if cue_type ~= response_layer
            e = e.*((input(cue_type,:,:which_item)*type_connectivity{cue_type}(:,:,2)));
        end
    end

    % retrieved type nodes
    which_typ = type_connectivity{response_layer}(:,:,2)*transpose((token*token_connectivity(:,:,2)).*(bp_nodes.*e));

    % Setting up polar values
    polarC = [1:type_size]';
    polarC = (polarC * 6.28/type_size) - 6.28/type_size;
    polarC(:,2) = which_typ;

    % Summing cartesian values
    [X Y] = pol2cart(polarC(:,1),polarC(:,2));
    x_value_sum = sum(X);
    y_value_sum = sum(Y);

    % Here are the two properties of the retrieved mean vector.
    [reported(trial,which_item) retrieved_length(trial,which_item)] = cart2pol(x_value_sum,y_value_sum);

    if reported(trial,which_item) < 0
        reported(trial,which_item) = reported(trial,which_item) + (2*pi);
    end
end
end

target(trial) = ceil(rand*which_load);

% reported value or the angle of the mean vector
r(trial) = reported(trial,target(trial));

```

```

% presented value of the target
p(trial) = deg2rad(presented(trial,response_layer,target(trial)));

% distractor color values
if which_load ~= 1
    possible_items = 1:which_load;
    d(trial,:) = deg2rad(presented(trial,response_layer,possible_items(possible_items-
=target(trial)))); 
end

end

% passing the data into the mixture model (Bays et al., 2009).
% for set size 1, there is no non-targets
if which_load == 1
    output = JV10_fit(r',p');

    sd(which_load) = rad2deg(sqrt(1/output(1))); % precision
    g(which_load) = output(4); % guess rate
    b(which_load) = output(3); % swap rate
    pm(which_load) = output(2); % percent in memory

else
    output = JV10_fit(r',p',d);

    sd(which_load) = rad2deg(sqrt(1/output(1))); % precision
    g(which_load) = output(4); % guess rate
    b(which_load) = output(3); % swap rate
    pm(which_load) = output(2); % percent in memory

    d = [];
end
end

function input = Type_activation(presented,type_size,act_peak,input)

% Code that converts the stimulus value into the activity in the type layer

type_resolution = ceil(360/type_size);

for layer = 1:size(presented,1)

    %Converting a stimulus value into type node activity
    for item = 1:size(presented,2)

        type_space_location = presented(layer,item) / type_resolution;

        type_node_1(item,layer) = floor(type_space_location)+1;
        type_node_2(item,layer) = ceil(type_space_location)+1;

        if presented(layer,item) == 360

            type_node_1(item,layer) = 1;

```

```

    type_node_2(item,layer) = 1;

    elseif presented(layer,item) > (360 - (360/type_size))

        type_node_1(item,layer) = type_size;
        type_node_2(item,layer) = 1;

    end

    type_node_1_activition(item,layer) = (1-(type_space_location - floor(type_space_location)))*act_peak(layer);
    type_node_2_activition(item,layer) = ((type_space_location - floor(type_space_location)))*act_peak(layer);

    input(layer,type_node_1(item,layer),item) = type_node_1_activition(item,layer);

    if type_node_1(item,layer) ~= type_node_2(item,layer)
        input(layer,type_node_2(item,layer),item) = type_node_2_activition(item,layer);
    end

end
end

function model = Setupconn(type_conn,token_conn,bp_size,type_overlap,type_size,num_layers,toke
n_size)

% This code set ups the connection weights for the token and type layers
% connected to the binding pool

for layer = 1:num_layers

    type_weight{layer} = zeros(type_size,bp_size);

    type_weight_prop(layer) = ceil(type_conn(layer)*bp_size);

    independent_type_weight_prop(layer) = ceil(type_conn(layer) * bp_size * (1-type_overlap(la
yer)));
    overlap_type_weight_prop(layer) = ceil(type_conn(layer) * bp_size * type_overlap(layer));

end

% For each type node, this sets up the initial connections
for layer = 1:num_layers
    for i = 1:type_size

        b = zeros(bp_size,1);
        a = randperm(bp_size);

        b(a(1:independent_type_weight_prop(layer))) = 1;

        type_weight{layer}(i,:) = b;
    end
end

% When similar connections is active, then this sets up the

```

```

for layer = 1:num_layers
    if type_overlap(layer)

        bps = [];
        bp = [];
        for(conn = 1:overlap_type_weight_prop(layer))

            % this goes through each type node
            for(t = randperm(type_size))

                % this will either 'hit' the left or right proximal type
                if(rand > .5)
                    whichbuddy = 1;
                else
                    whichbuddy = -1;
                end

                % looking at the neighbor will change if it is either the max
                % or min of the type space
                whichbuddy = t + whichbuddy;
                if(whichbuddy > type_size) %wrap around
                    whichbuddy = whichbuddy - type_size;
                end
                if(whichbuddy < 1)
                    whichbuddy = whichbuddy + type_size;
                end

                % Find all active connections
                bps = find(type_weight{layer}(whichbuddy,:)==1);
                % Randomly select one of these connections
                bp = randi(length(bps));
                while(type_weight{layer}(t,bps(bp)) == 1)
                    bp = randi(length(bps));
                end

                % Now, activate that connection in the buddy
                type_weight{layer}(t,bps(bp)) = 1;
            end
        end
    end

    for layer = 1:num_layers
        model.type_weight{layer}(:,:,1) = type_weight{layer};
        model.type_weight{layer}(:,:,2) = type_weight{layer};
    end

    % This is setting up connections between the Token layer and Binding pool.
    token_weight = zeros(1,bp_size);
    token_weight(1:bp_size) = 0;

    % Determining the proportion of overlap between token connections to the
    % binding pool
    token_weight_prop = ceil(token_conn * bp_size);

    for token = 1:token_size

```

```
% This code is used instead (which is the exact same as the setup
% for selecting type connections) if the token connections aren't
% fixed.
b = zeros(bp_size,1);
a = randperm(bp_size);

b(a(1:token_weight_prop)) = 1;

token_weight(token,:,1) = b;
token_weight(token,:,2) = b;

end

model.token_weight = token_weight;
```

Published with MATLAB® R2015a

Appendix B

Below is the Matlab code that is different between the old version of the model (Swan & Wyble, 2014) and the version of the model with the new encoding mechanism described in Chapter 4.

Contents

- [Parameters](#)
- [storage](#)

```
function RunModel_new
```

Parameters

```
num_items = [1 2 4 6];

type_size = 10;
num_layers = 2;

bp_size = 2500;
type_conn = [.2 .5];
token_conn = .12;
token_size = 8;
cue_layer = [2];
response_layer = 1;
token_individuation = .015;
capacity = [2:7];

type_overlap = [.3 .05];

act_peak = [1 1]; % column == num_layers

exp_bp_amount = 2;
```

storage

```
% The following lines of code would replace lines 50 to 73 (before retrieval)
% in the old Binding Pool model script

bp_nodes = zeros(1,bp_size);

[null,cindex] = sort(rand(size(capacity)));

for which_item = 1:which_load
    if which_item <= capacity(cindex(1))

        token = zeros(1,token_size);
        token(1,which_item) = 1;

        a = 1;

        for layer = 1:num_layers
            bp_nodes = bp_nodes + a.*((input(layer,:,:which_item)*type_connectivity{layer}(:,:,1)).*( token * token_connectivity(:,:,1)));
            end
    end
```

```
end  
bp_nodes = bp_nodes.^exp_bp_amount;  
bp_nodes = bp_nodes./sum(bp_nodes);
```

Published with MATLAB® R2015a

Vita
Garrett Swan

Education

2012 to 2017	Ph.D. in Psychology, The Pennsylvania State University
2011 to 2012	Ph.D. in Psychology, Syracuse University, Transferred with advisor to Pennsylvania State University
2007 to 2011	Bachelor of Arts, North Carolina State University

Selected Publications

- Swan, G., Wyble, B., Chen, H. (2017). On the robustness of memory representations when faced with an immediate surprise test. *Attention, Perception, & Psychophysics*
- Rajsic, J., Swan, G., Wilson, D. E., & Pratt, J. (2017). Accessibility Limits Recall from Visual Working Memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*
- Wyble, B., Swan, G., & Callahan-Flinton, C. (2016). Measuring Visual Memory in Its Native Format. *Trends in Cognitive Sciences*.
- Swan, G., Collins, J., & Wyble, B. (2016). Memory for a single object has differently variable precisions for relevant and irrelevant features. *Journal of vision*, 16(3), 32-32.
- Chen, H., Swan, G., & Wyble, B. (2016). Prolonged focal attention without binding: Tracking a ball for half a minute without remembering its color. *Cognition*, 147, 144-148.
- Wyble, B., & Swan, G. (2015). Mapping the spatiotemporal dynamics of interference between two visual targets. *Attention, Perception, & Psychophysics*, 77(7), 2331-2343
- Swan, G., Wyble, B. (2014). The Binding Pool: A Model of Shared Neural Resources for Distinct Items in Visual Working Memory. *Attention, Perception, & Psychophysics*, 76(7), 2136-2157

Invited talks

- "Relevant and irrelevant features of attended objects have differently variable precision" University of Groningen, Netherlands, June 9th 2016
- "Delayed estimation as a new tool for measuring the quality of visual memories in working memory" Centre for Functioning Health Research, Queensland, Australia, August 27th 2015
- "Mechanisms and simulations of the Binding Pool model of visual working memory" University of Sydney, New South Wales, Australia, July 21st 2015

Grants/Awards

- National Science Foundation: East Asia and Pacific Summer Institutes. Title: Investigating the mechanisms of visual working memory by testing predictions of a computational model. Funding Amount: \$5250.00
- Alumni Association Dissertation Award. Funding Amount: \$5000.00