THE ROLE OF HIERARCHICAL STRUCTURES IN COGNITION

by

MELISSA EMILY MOSS

A DISSERTATION

Presented to the Department of Psychology
and the Division of Graduate Studies of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

March 2023

DISSERTATION APPROVAL PAGE

Student: Melissa Emily Moss

Title: The Role of Hierarchical Structures in Cognition

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Psychology by:

Dr. Ulrich Mayr	Chairperson
Dr. Nash (David) Unsworth	Core Member
Dr. Elliot Berkman	Core Member
Dr. Melissa Redford	Institutional Representative

and

Krista Chronister	Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded March 2023

DISSERTATION ABSTRACT

Melissa Emily Moss

Doctor of Philosophy

Department of Psychology

March 2023

Title: The Role of Hierarchical Structures in Cognition

Individuals routinely execute complex tasks that involve multiple, dependent levels of information, such as driving a car or cooking dinner. It is amazing that our cognitive system is able to represent such complex, hierarchical tasks without becoming overwhelmed by the sheer amount of information needed to successfully complete the task. Hierarchical tasks require the integration of multiple levels of information. How the cognitive system organizes and uses this hierarchical information is a key question in cognitive psychology. Through disparate literatures in psychology, including serial-order control, task switching, and learning, this phenomenon has been studied from multiple angles. Many findings from these different areas point to the existence of hierarchical cognitive structures for representing complex tasks, though many questions remain. In this dissertation, I first address the question of how relationships between hierarchical components are defined and used by the cognitive system. Then I assess how the cognitive system allocates resources when executing hierarchical tasks. Finally, the question of related cognitive processes and of the application of hierarchical control to different types of complex tasks is addressed, using an individual differences approach.

This dissertation includes previously published and unpublished co-authored material.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Many skills and tasks require the cognitive system to handle multiple dependent levels of information. For instance, when dancing with a partner, it is necessary to know what step is being cued by the other person's preparation movement, based on information concerning the type of dance at the "top" level, and below that, which step within that dance type is being signaled, and further, what set of motor actions is required to produce that dance step. Thus, something as simple as a waltz is actually quite complex and hierarchical in nature. In another dance situation, performing a choreographed dance requires continued knowledge of where you are in the sequence of steps, to execute the moves in the correct order. Basic sets of motor actions are grouped, or "chunked," into individual dance steps, which are then chunked into eight-count series of steps. Based on what eight-count you are in, and which step in the eight-count you just executed, you can determine the next dance step. To be able to execute such a complex task, in the form of either partner-cued dance or serially ordered choreography, an individual must rely on their internal representation of the set of dance movement rules.

The information necessary for executing these, and other, complex tasks is organized hierarchically. By this definition, hierarchical tasks are those in which decisions must rely on multiple levels of information, such that decisions on one level depend on decisions made on a higher level, which in turn might depend on still higher-level decisions. Situations such as these, in which an individual must maintain and utilize multiple interdependent levels of information may also require hierarchical cognitive representations. It is important to differentiate here between hierarchical tasks on the one hand, and cognitive structures on the other. I focus on the question of how the cognitive system represents complex, or hierarchical, tasks. The task components or rules themselves are hierarchical, but that does not necessarily mean there are parallel structures in the cognitive representations of the tasks. If these types of tasks are similarly structured in the cognitive system, then those structures *in cognition* may be referred to as hierarchical cognitive structures or hierarchical representations. The general term for representing and using hierarchical concepts will be referred to here as hierarchical control. The chapters of this dissertation investigate different questions concerning the nature of these representations.

Relationships between Hierarchical Components

Foundational work in the field of hierarchical control has tried to identify how the structure of complex sequential information is represented by the cognitive system. In arguably the most important foundational paper in serial-order control, Lashley (1951) rejected the most prevalent theory at the time of associative chaining, in which each element in a sequence cues the next by direct association, as too simple a mechanism. Instead, the cognitive system organizes complex sequences of elements into subgroups, called chunks. Therefore, positional information concerning the location of element within chunk and chunk within sequence must be utilized in the execution of complex serial-order tasks.

Following Lashley's proposition that serial position information is necessary in hierarchical control, Restle (1970) found that people recognize and use abstract patterns when learning and executing serial-order tasks. Complex sequences are decomposed into subsequences represented in cognition as systems of rules to be applied serially. These rules, or codes, utilize information concerning sets of patterns with the same structural relations in a sequence, and it is these relationships between units that allows the cognitive system to handle complex sequences (see Figure 1.1).

Restle's application of structural trees from computer science to serial-order processes provided the basis for the tree traversal interpreter model (Collard & Povel, 1982). The tree traversal interpreter presents a possible mechanism for how hierarchical representations of sequential information, in the form of memory codes or formulas, are translated into the structural trees used in the execution of sequences. When executing a sequence, the tree traversal interpreter decodes the hierarchical representation, yielding a set of transformations that correspond to the differences between subsequent elements. In other words, each element in a sequence can be computed from the previous element by applying the appropriate operations to traverse the structural tree. This interpretation process occurs on-line, and each transformation is executed in a serial fashion, along the structural tree. Because each step in the structural tree (within or across level) requires an additional operation, longer paths between adjacent elements (e.g., between the last element in chunk 1 and the first element in chunk 2) should require more time to traverse. The tree traversal interpreter model provided a possible explanation for how subunits connect, and how an individual can get from one component to the next, by utilizing relationships between them to traverse the hierarchical structure.

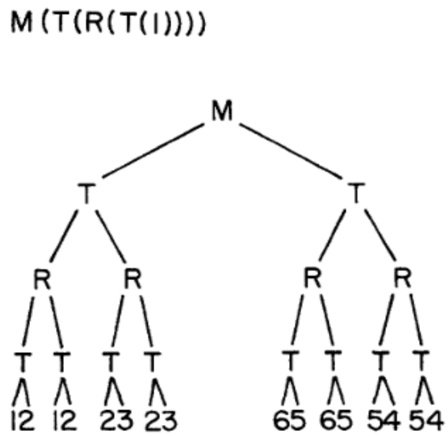


Figure 1.1. Example of the tree diagram for a long, regular binary pattern. M = mirror, T = transposition, R = repeat. Applying each operation of the formula to 1, from the furthest nested out, results in the sequence shown across the bottom of the structural tree. From Restle (1970).

Miller, Galanter, and Pribram (1960) proposed that the subcomponents within this possibly hierarchical cognitive structure operated as Test-Operate-Test-Exit (TOTE) units (see Figure 1.2). In this model, a recursive feedback loop, or set of loops, were proposed to be the basic units of behavior by which complex information, such as that required for hierarchical task execution, can be processed. Different elements can be placed into each TOTE unit in a sequence, and the units are completely modular, meaning they can be organized in any order. In contrast to the model put forth by Restle, in which relationships between chunks are utilized, the TOTE model proposes that all units are independent of and “blind to” each other. This work provided the basis for conceptualizing a decontextualized control model for learning and task execution.

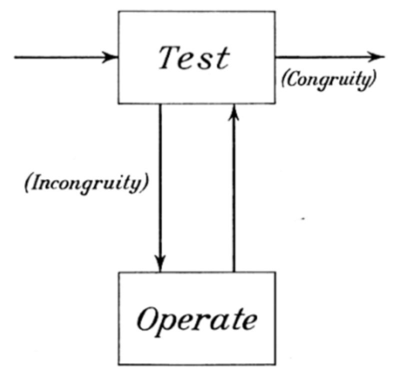


Figure 1.2. The TOTE unit. If the “test” criterion is not met (“incongruity”), an “operation” step would be performed on the stimulus, and then it would be re-tested, at which point it would either meet the test criterion (“congruity”) and exit the loop, or it would re-enter the loop. From Miller, Galanter, & Pribram (1960).

Based on this TOTE model, the efficiency of hierarchical cognitive control representations would seem to come from the modularity of different representational units, or chunks, within a hierarchical structure, which allows flexible rearranging and eliminates potential interference across units. In this idealized architecture, different units are accessed through symbols that carry no information about what is in a particular unit. Yet, such an architecture also comes with the drawback that the relationships between representational units cannot be utilized when encoding or using such structures. More importantly, there is empirical evidence that people are able to use exactly such relationships to arrive at parsimonious representations of complex structures. For instance, as previously mentioned, Restle (1970) showed that people can identify and use similarities across sequences, such that performance with transposed sequences was better than with random sequences. This finding indicates that the cognitive system may be able to exploit similarities across "independent" groupings. Whether and how the cognitive system identifies and uses these relationships between modules in hierarchical representations is unknown. How are relationships between modules discovered by the cognitive system? Do they only occur in domains that contain inherently ordinal structures, such as numerical patterns? Is this type of pattern detection an emergent quality with the acquisition of skilled or routine action, or is it utilized from the outset?

Dehaene and colleagues (2015) argue that chunks are represented by their abstract grammatical schemas, or "algebraic patterns defined by identity relationships," (p. 9) and then organized within nested (hierarchical) tree structures. For instance, the steps left-left-right follow an A-A-B pattern, which would more closely resemble another A-A-B chunk of steps (e.g., forward-forward-backward) than a different chunk pattern, such as forward-backward-forward (A-B-A). Research with spatial sequence learning has demonstrated that these algebraic patterns can be identified and used by the cognitive system in order to represent complex sequences most efficiently (Amalric et al., 2017). In this context, complexity is defined by how much of the to-be-represented hierarchical structure can be compressed based on patterns or regularities across chunks. Though these findings have not yet been generalized to non-spatial sequences or non-serial hierarchical tasks, they provide convincing evidence of one way in which the cognitive system might utilize positional information within chunks to identify relationships between the chunks of hierarchical information. Further, they give an early indication that the cognitive system may be built to look for relationships between chunks from the outset, although the ability to exploit those similarities must be strengthened through learning of the sequences.

How can these findings be reconciled with the theoretical assumptions of hierarchical structures as modular and content-independent? Going back to the early work concerning structural trees (Collard & Povel, 1982; Povel & Collard, 1982; Restle, 1970), it was theorized that an interpretation process happens in which sequential representations, encoded as memory codes of some sort, are translated into a set of transformations that compute each element from the one before, in order to establish the structural trees that must be traversed in the execution of complex sequences. This work highlights a mechanism by which relationships between sequential components can be utilized in order to navigate the hierarchical structure. Following this logic, chunk similarity (regularity) information may initially be encoded as part of the serial-order representation and then utilized when decoding the representation into executable sequential elements. This idea provides an alternative to the standard TOTE model of hierarchical representations.

Processing Constraints in Hierarchical Control

Though most of the work in hierarchical control focuses on defining or characterizing the types of representations that are involved, it is also important to understand why hierarchical control is difficult. In other words, what cognitive resources are necessary, and how do processing constraints arise when traversing the representational structures for hierarchical tasks?

The adaptive gating model identifies a biologically plausible mechanism for processing hierarchical information in the brain, by defining hierarchical control through recurrent connections between cortical and subcortical brain regions (Frank, Loughry, & O'Reilly, 2001; O’Reilly & Frank, 2006). In this model, there are working memory stripes, or slots, in the prefrontal cortex (PFC) that can encode different elements or sets of elements. This representation can be maintained through recurrent connections within PFC, or flexibly updated based on inputs from other areas. The basal ganglia enforce a gating mechanism whereby additional information can only flow to PFC when the gate is “open” (i.e., when the striatum disinhibits signals between frontal and motor regions). Each step of this integration/maintenance process requires cognitive resources. However, it is not clear exactly when the different steps occur, and therefore when different amounts of resources are used, within the execution of a hierarchical task.

Processing constraints could arise from either the integration of information across multiple levels at updating points or the maintenance of complex structures requiring the recruitment of more representational resources, or a combination of both. In order to address this question, it is important to determine what exactly causes performance to suffer with more (i.e., higher) hierarchical levels, referred to here as the number of levels, or “n-of-levels” effect.

If cognitive representations assume a hierarchical structure according to the standard TOTE model, then a complex task to be executed should be divided into its subcomponents, such that different hierarchical levels occupy distinct representational subspaces (Miller et al., 1960). This division into independent components across levels would protect lower-level decisions from the cognitive demands on higher levels of the hierarchy. In this way, starting at the top of the hierarchy, each level can “program” the next lower level in a ballistic manner (Povel & Collard, 1982), and performance costs at each level of the hierarchy would only occur when settings on that level needed to be updated, or reprogrammed. If this were the main processing constraint, n-of-levels effects would be confined to points of updating, during which information needs to be integrated across levels. Further, this n-of-levels effect would disappear in situations that maintain the same settings and therefore do not require updating.

Alternatively, maintenance within a global representational working memory space might be necessary for traversing multilevel control structures (Dehaene, Kerszberg, & Changeaux, 1998; Waltz et al., 2000). In this case, additional information (in the form of additional levels) would simply take up more representational resources: The more levels a complex task includes, the more neural resources are required to represent both the full hierarchical structure and the current state, or location, within that structure. If this type of maintenance is required in hierarchical control, performance costs should occur as a function of total amount of information being represented, and this effect should be present across all levels, regardless of level-specific decisions (Monsell, 2003). In this case, there would be global costs in the form of static n-of-levels effects based on the size of the hierarchical structure, regardless of the requirements of specific decision points. Interestingly, this behavioral indicator of maintenance can only be used for non-serial, or cue-based, paradigms in which it is possible to have a true no-updating condition (i.e., no movement from current position in the hierarchical structure). The interpretation becomes ambiguous in a serial-order paradigm, due to the fact that in sequential execution, position within the hierarchical structure must be updated with every action. Because of this, it is possible (and theoretically more plausible) that a maintenance-like effect on performance in serial order would actually be due to the updating process occurring at every decision point instead of only during the traditional decision tree branching points. There is not yet a solution for this ambiguity, because we have no way of creating a fully maintenance-based situation in sequencing. This marks an important difference between serial-order and cue-based hierarchical structures, in terms of research methods and interpretation. However, it does not provide direct evidence of whether or not serial-order and cue-based hierarchical tasks rely on the same resources or conform to the same underlying structures.

Defining Hierarchical Structures Across Individuals

To what degree is hierarchical control a unique phenomenon, above and beyond other cognitive control processes? This question actually has three parts. First, is there a "universal" structure of hierarchical control, identifiable across individuals, and if so, how are complex tasks broken down? Second, does performance vary more between individuals when they are dealing with higher levels of abstraction, versus lower levels? And third, if this is the case, can the unique higher-level variance be explained entirely by known constructs, such as working memory capacity, long-term memory, and fluid intelligence, as well as differences in motivation or learned strategies? In other words, can variability between hierarchical levels be completely explained by existing cognitive constructs, or is hierarchical control a unique process? This question is best addressed in terms of individual differences, to empirically determine hierarchical level and derive the unique sources of variance across individuals for each level, as well as for serial-order versus cue-based hierarchical control tasks.

Unsurprisingly, hierarchical control ability correlates to other cognitive control processes. Early work from Marshalek, Lohman, and Snow (1983), Just and Carpenter (1985), and Carpenter, Just, and Shell (1990) expanded the understanding of structures associated with control of higher cognitive processes and systems, such as intelligence and spatial ability, and provided support for the idea that cognitive control structures themselves can be hierarchical. Carpenter et al. (1990) used both experimental approaches and simulations of performance on the Raven Progressive Matrices Test ("Ravens") and Tower of Hanoi task to identify the hierarchical processes utilized in tasks that require fluid intelligence. From this work, they proposed a process by which individuals are able to represent multiple abstract relations and access them dynamically as necessary, to accomplish task goals. The Ravens and Tower of

Hanoi tasks both require hierarchical rules to be maintained. However, the rules are applied serially in the Tower of Hanoi task, whereas there is no seriality requirement for solving the Ravens task. Therefore, in addition to identifying a relationship between fluid intelligence and hierarchical control, this work also provides evidence of shared resources across the two types of hierarchical control processes, with very high correlations between performance on the two tasks (r=.77, p<.01). In other words, more hierarchically complex tasks, regardless of format, are more central to the inter-individual differences in fluid intelligence. Just and Carpenter (1985) identified similar hierarchical processes concerning spatial ability, whereby differences in type of cognitive coordinate systems led to individual differences in mental rotation and the ability to solve spatial tasks. This early work identified relationships between hierarchical control and other cognitive processes and established hierarchical control itself as an important function of cognitive control. Though these studies provide important evidence of the relationship between hierarchical control and other higher-level cognitive processes, it is important to note that fluid intelligence, for instance, is itself a complex activity. Therefore, the relationship of hierarchical control with fluid intelligence does not clarify its relationship with more basic cognitive resources, such as those involved in working memory and long-term memory.

Working memory capacity has been identified as another cognitive process that is strongly related to hierarchical control. Bo and Seidler (2009) tested whether visuospatial working memory capacity affected the learning and organization of explicitly acquired motor sequences, using a paradigm in which 12-element finger tapping sequences were instructed, with no fixed grouping patterns. The chunks formed by each subject were defined by identifying chunk boundaries in their data (i.e., elements with significantly longer RTs). Participants who used longer chunks and those who began using consistent chunking patterns early in learning performed better in the serial-order task. They also found a strong positive correlation between working memory capacity and both rate of early learning and chunk length. However, it should be noted that the correlation between working memory capacity and overall sequence RTs was not significant (although it was trending in the right direction), which introduces some ambiguity into the relationship between working memory and serial-order control. Additionally, this study found evidence of item-dependence in early but not later representations, indicating that chunking patterns become more abstractly represented with practice. Building upon this work, Kikumoto and Mayr (2018) found that individuals with lower working memory capacity showed worse performance at chunk transitions, compared to those with higher working memory. This indicates that working memory plays a role in identifying or retrieving the next chunk in a hierarchical structure. EEG decoding results further clarified the role of working memory capacity, showing that only high-capacity individuals were able to maintain a robust representation of sequential context in the form of current chunk identity, while executing within-chunk elements. Thus, working memory capacity selectively constrains an individual’s ability to represent both the current action and the higher-level contextual information necessary for traversing hierarchical structures.

In other research, multitasking, a situation that clearly requires hierarchical control, has been found to strongly correlate with working memory capacity and fluid intelligence (Redick et al., 2016), as well as long-term memory (Burgess et al., 2000). The relationships between these constructs are important to understand and account for when measuring hierarchical control, in order to determine how much of the effect is actually due to the act of managing and traversing hierarchical representations, above and beyond other important and strongly implicated cognitive control processes.

Though there is a well-documented relationship between working memory, long-term memory, and hierarchical control, there are remaining ambiguities. For instance, the construct of working memory comprises multiple processes, including the temporary storage process (maintenance) and storage limits (capacity), as well as retrieval and manipulation (processing) of representations (Unsworth et al., 2014; Unsworth & Engle, 2007). Different tasks require some of these processes more than others, indicating that the processes of working memory really do fill different roles. Further, the functions of working memory that deal with encoding and retrieval are clearly intertwined with long-term memory processes. To this end, it may be useful to split working memory into a storage component and a long-term memory encoding/retrieval component, in order to determine whether both facets of working memory are related to hierarchical control. It may also be the case that these different components of working memory are important for separate aspects of hierarchical control. For instance, working memory storage may be more important for determining the size of subunits or chunks in hierarchical representations, while retrieval processes may be necessary for determining which chunk to execute next at each decision point. It is also important to note that the cognitive processes implicated in hierarchical control (fluid intelligence, working memory, and long-term memory) have been shown to strongly correlate with each other (Unsworth et al., 2014). The complexities of these relationships must also be considered when modeling their unique contributions to hierarchical control processes. In principle, a comprehensive structural equation model should be able to account for these processes in order to assess whether hierarchical control is its own unique process. Further, this type of model would allow us to analyze the relationship between these related constructs with each hierarchical level separately, to determine if and to what extent they are differentially necessary resources across level of task complexity.

Overview of the Dissertation

The goal of this dissertation is to characterize the relationships between hierarchical components, and across hierarchical structures. In three empirical studies, I demonstrate how the cognitive system organizes hierarchical units in relation to one another (Chapter 2), the unique processing demands of executing cue-based and serial-order hierarchical tasks (Chapter 3), and whether individual differences in performance of different hierarchical tasks can be explained by unique cognitive processes (Chapter 4).

The extant body of work providing evidence of inter-chunk relationships in serial-order control includes only studies using material with some kind of inherent order, such as sequences of numbers or finger taps. However, it is unclear if the same conclusions could be drawn concerning relationships between chunks in which there are no easy mathematical transitions (e.g., when dealing with hierarchical rules or goals). We will address this gap in Chapter 2, using an explicit sequencing paradigm to determine whether the cognitive system is able to exploit similarities across chunks. In line with previous work (Amalric et al., 2017; Restle, 1970), chunk similarities are determined by the abstract patterns, or grammars, of the chunked elements. Importantly, the chunked elements are not mathematically related to one another, but are instead groups of rules to be applied to the stimulus on each trial. Thus, this set of experiments uses completely non-numerical sets of elements, in order to determine how abstract the cognitive representations of sequential information can be. Here, we will provide evidence across four experiments that serial-order performance benefits from chunks with matching abstract patterns, indicating that the cognitive system can indeed determine grammatical similarities from non-numeric patterns. Further, we will demonstrate that the cognitive system is able to identify such abstract relationships between chunks from the outset.

Chapter 3 will address the question of where processing constraints arise when performing serial-order and cue-contingent hierarchical tasks, using a model-based approach. To address the question of processing demands across hierarchical level and task structure, we use an experimental design that included three tasks, each split into four hierarchical levels, in both serial-order and cue-based formats. This symmetry between the serial-order and cue-based tasks allows us to compare analogous models of hierarchical control processing across the two formats. We will present two experiments: an in-lab version (n=23), for which various exploratory analyses were used to fine-tune our models, and an online version (n=53), in which we replicate our findings using the predefined models on a new sample. Here, we will determine the pattern of processing constraints for hierarchical control, as updating and adjacent-level integration costs in cue-based tasks, and global maintenance costs in serial-order tasks. This will provide evidence that serial-order and cue-contingent hierarchical structures have potentially different processing constraints, and therefore, though related, could possibly be considered distinct from one another.

Chapter 4 will explore the structure of hierarchical control across individuals and the relationship between hierarchical levels and other cognitive constructs. Previous neuroimaging work has provided evidence of distinct anatomical regions and functional pathways for processing different hierarchical levels (Badre & Nee, 2018; Frank & Badre, 2012; Koechlin, Ody, & Kouneiher, 2003; Ranti, Chatham, & Badre, 2015). However, given that these studies all used within-subjects methods, it is unclear how the different hierarchical levels affect inter-individual performance differences in hierarchical tasks. In order to address this, we will first determine the structure of hierarchical control by comparing structural equation models. We will then use the best models to ask whether performance varies more between individuals when they are dealing with higher levels of abstraction, versus lower levels. Further, we will determine whether serial-order and cue-based tasks utilize the same hierarchical structures. Finally, we will determine whether the different levels have unique relationships with other constructs, such as working memory capacity, long-term memory, and fluid intelligence.

Together, this set of empirical findings will help to characterize the how the cognitive system subdivides hierarchical task representations into independent subspaces, and in what ways information is or is not integrated across the distinct subspaces.

CHAPTER II

THE EFFECT OF ABSTRACT INTER-CHUNK RELATIONSHIPS ON SERIAL-ORDER CONTROL

From Moss, M.E., Zhang, M., & Mayr, U. (2022). The effect of abstract inter-chunk relationships on serial-order control. [Submitted for publication]. Department of Psychology, University of Oregon. Preprint available at SSRN: https://ssrn.com/abstract=4303221 or http://dx.doi.org/10.2139/ssrn.4303221

1. Introduction

Since the beginning of the cognitive revolution, hierarchical control has been viewed as a core element of human cognition that enables complex action patterns. Recently, this interest has been revived, mainly through neuroscience evidence suggesting a neuroanatomical reality to hierarchical levels (Badre, 2008; Koechlin et al., 2003). Theories of hierarchical control that have developed alongside the neural evidence usually suggest that higher levels of control represent the context or rules that shape the current, lower-level processes. On each level, selection of the currently relevant representation is shaped by the context provided by the representations in the next-higher level. The elements on a given level are typically treated as individual entities that are selected one at a time, often assuming a winner-take-all selection mechanism and therefore to the exclusion of competing entities on the same level (Duncan, 2010; Miller et al., 1960; Rosenbaum, Kenny, & Derr, 1983). This strategy provides a divide-and-conquer approach to complex task spaces: By carving a larger task into separate, manageable subspaces, interference can be eliminated and the lower-level subspaces can be flexibly recombined to solve new problems (Badre & Nee, 2018).

Notions of hierarchical control have been particularly important in the context of serial-order control of action sequences, such as playing the piano, typing, or processing language. People tend to proceed through complex sequences in "chunks" of 3-4 basic elements (e.g., notes in a piece of music), rather than in an element-by-element manner (Lashley, 1951). Chunks, in turn, can be organized into larger subsequence plans, and so forth (Dehaene et al., 2015; Krampe, Mayr, & Kliegl, 2005; Lashley, 1951). The borders of chunk representations within a larger sequence can be identified by higher RTs and lower accuracy rates at the chunk transition positions (Lien & Ruthruff, 2004; Schneider & Logan, 2006; Wu et al., 2017).

From the perspective of currently dominant models of hierarchical control, these types of behavioral patterns can be explained by assuming that on each level only one chunk or plan is active at a time, specifying the sequence of within-chunk elements on the level below. This comes with the implicit assumption that a chunk-level code "knows nothing" about its contents (the specific elements it contains), or of the serial-order or content in competing chunks (Fitch & Martins, 2014). Such knowledge transfer between chunks or levels would create exactly the kind of interference across regions of the task space that the hierarchical "divide and conquer" strategy is supposed to eliminate.

Interestingly however, much of the early literature on serial-order control indicates that abstract relationships between the content of chunks on the same level impact behavior. Such evidence comes from studies requiring participants to learn key-press sequences, which either followed abstract patterns or not, from feedback. Most notably, Restle (1970) showed that people use similarities across consecutive parts of a sequence, such that performance with transposed elements was better than with random sequences (Collard & Povel, 1982; Koch & Hoffmann, 2000; Povel & Collard, 1982). More recently, Dehaene and colleagues (2015) argued that chunks are encoded in terms of their abstract grammatical schemas, or “algebraic patterns defined by identity relationships,” (p. 9) and then organized within nested (hierarchical) tree structures. Further, Amalric et al. (2017) have provided evidence that people spontaneously apply geometric primitives to spatial sequences in order to recursively organize entire sequences.

The fact that the cognitive system appreciates abstract, inter-chunk relationships provides a theoretical challenge for models that assume only one chunk on each level is active at a time, and only placeholders that do not themselves contain chunk-relevant information are present on the level above (Fitch & Martins, 2014). However, the relationship between sequential performance and chunk similarity has only been demonstrated using learning paradigms in which participants were asked to “discover” a given sequence through hypothesis testing and feedback. Further, this previous work always used stimuli with an inherent order, such as sequences of numbers or spatial locations. With these limitations, it is not clear whether the appreciation of abstract inter-chunk patterns is a general feature of serial-order representations, or if it is simply limited to learning situations and ordered sets of stimuli for which inter-stimulus relationships can be derived from simple arithmetic operations.

Therefore, we aim to address here the following questions:

1) Are abstract patterns relevant, even in the absence of ordered sets of elements? If it is the case that chunks may be represented relative to one another based on shared content-independent patterns (Dehaene et al., 2015), executing a given chunk is not determined only by the chunk itself, but also by its abstract relationship to “neighboring” chunks, even when there is no intrinsic ordering relationship between elements. In this case, retrieval and execution of complex sequential information should benefit when neighboring chunks share abstract grammars. For instance, if our cognitive system uses

this coding of chunk patterns, the two-chunk sequence ABB-CDD, in which each letter stands for a basic element (such as a musical note or a stimulus-response rule, as in the current study), should be easier to execute than the sequence ABB-DCD, because the former uses a common chunk grammar across the sequence, whereas the latter does not.

2) By the "standard model", two competing chunks should not be active at the same time, which leads to the question of how exactly abstract, inter-chunk relationships are discovered and used. One possibility is that inter-chunk relationships are relevant when transitioning from one chunk to the next, possibly in a short time window during which both chunks are active. In this case, benefits from abstract relationships should be particularly strong during chunk transitions. Alternatively, sequences with abstract inter-chunk relationships may be overall represented in a more efficient manner that decreases inter-chunk interference, in which case there would be benefits across all chunk positions.

3) Does appreciation of abstract patterns emerge over time, or can they be exploited by the cognitive system from the outset? Given that previous work in this area has used learning paradigms in which relationships between chunks were established gradually through hypothesis testing, the question of gradual learning versus instantaneous use of abstract chunk relationships has yet to be addressed experimentally. Using explicitly instructed rule sequences, we aim to determine to what degree the cognitive system has an a priori expectation of relationships between chunks. If abstract relationships must be gradually discovered, we should see differential improvement across repetitions of sequences with more versus less similar chunk grammars. However, if our cognitive system generally expects to encode chunks relative to each other patterns, we should see benefits in performance of sequences containing similarly patterned chunks from the outset.

2. Experiment 1: Shared-Element Chunks

In order to address our hypothesis, we began with constructing sequences that followed relatively closely the types of sequences used in the work by Restle (1970) in that in our "matching" sequences, abstract relationships were used to translate one part of the sequence (i.e., chunk) into the next. However, given that we used no ordered set of stimuli, no arithmetic transformations could be applied here.

2.1. Methods

2.1.1. Participants

Data were collected from 56 undergraduate students at the University of Oregon. Existing research provides no strong guidance for effect-size estimates. Therefore, we chose a sample size that provided sufficient power (.8) for detecting small to moderate effect sizes (f=.15). Participation was voluntary, and students were offered extra credit for their participation.

2.1.2. Spatial Rules Task

We applied an explicit sequencing paradigm to a simple spatial rules task in order to test our question about the role of matching chunk patterns in sequential performance (Mayr, 2002, 2009). On every trial of the spatial rules task, a white dot would appear in one corner of a frame, and participants were instructed to apply a spatial rule to the dot (e.g., horizontal) to determine which corner the dot would move to. Responses were made using the 4, 5, 1, and 2 on the keyboard number pad, with each key representing the corresponding corner of the frame (top left, top right, bottom left, bottom right, respectively). In this task, all task features (stimulus, rule, etc.) are orthogonal to one another.

2.1.3. Sequencing Paradigm and Procedure

Sequences were constructed from the two different spatial rules as elements. These rule sequences allow more precise temporal control over participants' sequential performance (as opposed to using response sequences). Further, response sequences would utilize inherently ordered information, whereas different spatial rules have no obvious order and therefore allow us to establish truly abstract chunk grammars. Each rule sequence contained two 3-element chunks. For each new sequence, participants were presented with the sequence of rules listed vertically in an instruction screen (see Figure 2.1). Participants then cycled through each sequence, applying one rule per trial, several times. After an incorrect response, the sequence of rules would be

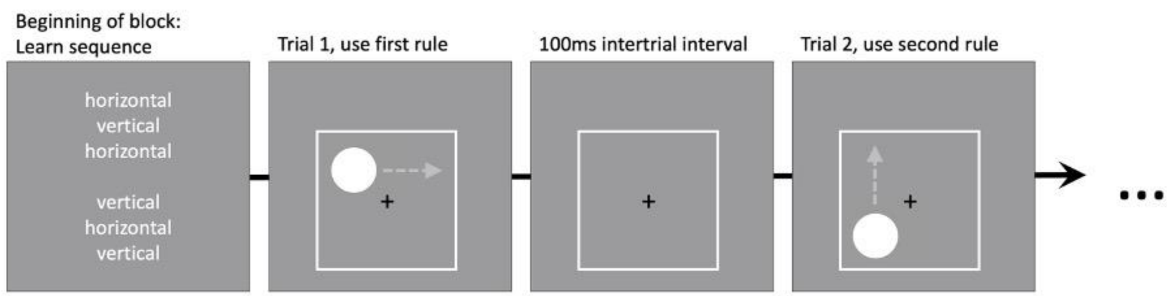


Figure 2.1. Sequence of events in the task-span procedure in Experiment 1, with explicitly instructed spatial rules. Each sequence of six rules was presented at the beginning of a block. After pressing any key, participants would begin cycling through the sequence, applying one rule per trial of the spatial rules task. Gray dashed arrows indicate correct responses and were not presented in the actual trials. In this example, the sequence of rules contains matching chunk grammars.

displayed above the frame, with the incorrect response colored red. Participants were then required to correct their mistake before moving on to the next trial.

Chunk patterns, or grammars, were defined by the order of their elements (rules), with matching chunks defined as those with the same abstract chunk grammar, regardless of what specific elements were in each position in the chunk. For this first, more conservative definition of grammars, the two chunks in a sequence contained the same pair of rule elements (horizontal and vertical), and matching grammars were identified by direct pattern inversion (e.g., ABB-

BAA). One-third of the sequences were "matching" and the other two-thirds were "non-matching." Participants completed 18 blocks of 48 trials (18 six-element sequences, each repeated 8 times) in a one-hour session.

2.2. Results

All data and analysis scripts relevant for this project are provided at https://osf.io/xuy6a/?view_only=6ecf656a428748d5a9c940e1f0473579.

The first cycle of each new sequence was removed as practice (i.e., the first six trials of each block), as well as trials with RTs below 100 ms or above the 99.5th percentile. Further, the data for subjects who did not complete the study or had accuracy rates below 70% were not used in analysis. Two subjects did not complete the study, and one subject did not meet the minimum accuracy inclusion criterion, and so the data from 53 subjects were used in analyses. Means and standard deviations can be found in Table 2.1.

Table 2.1. RTs and accuracy rates for matching and non-matching chunk grammars in each experiment.

	Reaction Times (ms)		Accuracy Rates (p)	
Experiment	Match	Nonmatch	Match	Nonmatch
1: Shared-Elements (N=53)	1073.23 (617.68)	1135.56 (683.00)	0.96 (0.19)	0.94 (0.22)
Chunk Position 1	1138.07 (659.86)	1300.90 (769.93)	0.95 (0.18)	0.94 (0.23)
Chunk Position 2	1054.46 (581.52)	1071.79 (618.95)	0.96 (0.19)	0.95 (0.21)
Chunk Position 3	1029.82 (552.13)	1037.75 (584.32)	0.96 (0.17)	0.95 (0.21)
2: Different-Elements (N=38)	1220.42 (810.78)	1227.03 (819.61)	0.96 (0.18)	0.96 (0.20)
Chunk Position 1	1606.69 (928.54)	1633.08 (944.09)	0.96 (0.18)	0.96 (0.20)
Chunk Position 2	1037.68 (670.72)	1036.66 (648.80)	0.97 (0.17)	0.95 (0.21)
Chunk Position 3	1024.33 (628.18)	1017.12 (652.97)	0.97 (0.16)	0.96 (0.18)
3: Different-Elements (N=41)	1226.33 (872.57)	1244.05 (903.99)	0.96 (0.19)	0.95 (0.21)
Chunk Position 1	1685.11 (1040.13)	1740.82 (1088.59)	0.95 (0.19)	0.94 (0.21)
Chunk Position 2	1022.44 (671.20)	1024.66 (684.70)	0.96 (0.18)	0.94 (0.22)
Chunk Position 3	986.54 (615.09)	978.73 (627.20)	0.96 (0.17)	0.95 (0.19)
4: Long Sequences (N=40)	1202.38 (865.93)	1217.64 (915.18)	0.96 (0.19)	0.94 (0.22)
Chunk Position 1	1680.71 (1074.84)	1786.66 (1148.28)	0.97 (0.13)	0.96 (0.15)
Chunk Position 2	1049.35 (703.62)	1090.35 (742.45)	0.95 (0.19)	0.92 (0.25)
Chunk Position 3	1073.47 (685.49)	1064.90 (750.49)	0.95 (0.18)	0.94 (0.21)
Chunk Position 4	1208.31 (838.21)	1231.4 (873.17)	0.96 (0.18)	0.94 (0.20)
Chunk Position 5	1148.82 (734.35)	1133.86 (762.91)	0.96 (0.18)	0.94 (0.21)
Chunk Position 6	1053.02 (712.42)	998.62 (679.73)	0.95 (0.19)	0.94 (0.21)

Figure 2.2 shows RTs and error rates as a function of sequence position and relationship between chunks (matching vs. non-matching grammars). Consistent with the notion that sequential performance is governed by hierarchically organized representations, the RT pattern shows strong and very typical sequence and chunk transition effects (see also, Mayr, 2009; Schneider & Logan, 2006). It is also apparent in the figure that RTs were faster and error rates

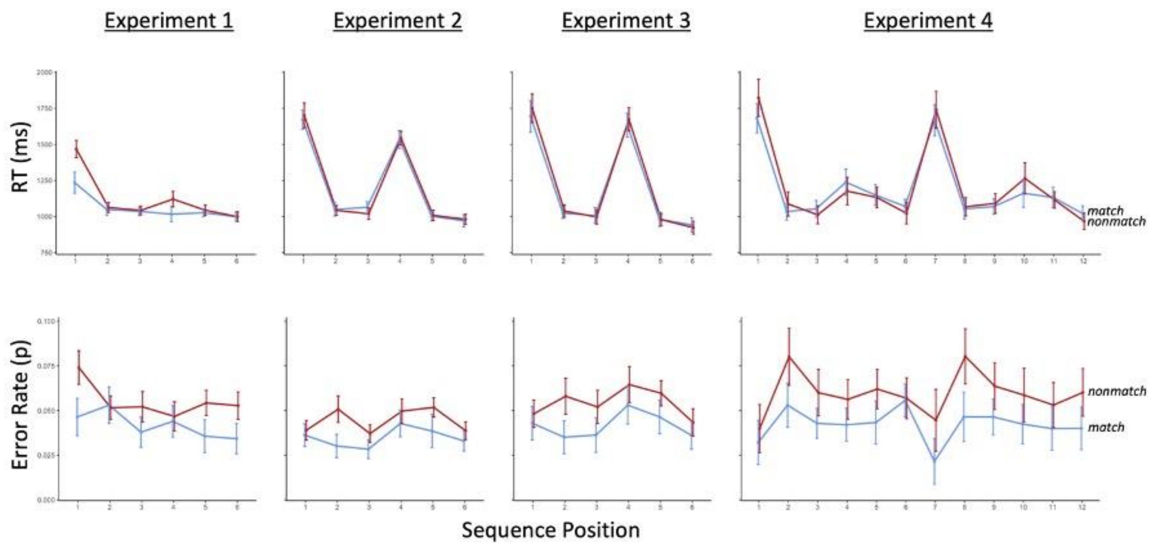


Figure 2.2. RTs (top panels) and error rates (bottom panels) for each experiment, at all positions in matching versus non-matching sequences. Error bars represent 95% within-subject confidence intervals.

smaller for sequences with matching versus non-matching chunks, and for RTs, that effect appears strongest at chunk transition points (first and fourth position in the sequence). We used linear mixed-effects models with subjects included as a random effect to control for between-subject variability to analyze these data (Bates et al., 2015; R Core Team, 2018). Correct trial RTs (log-transformed) and accuracy rates were the dependent variables, predicted by within-chunk element position, abstract grammar match, and their interaction as fixed effects. To capture potential transition effects on different levels of the hierarchical sequential representation, the following contrasts were applied across sequence positions: The chunk transition contrast (CT) compared positions 1 and 4 against positions 2, 3, 5, and 6. The within-chunk contrast (WC) tested positions 2 and 5 against positions 3 and 6). Finally, the sequence transition contrast (ST) compared the chunk transition of the first chunk (position 1) against the chunk transition of the second chunk (position 4) to test the effect of highest-level transition within these sequences. We are most interested here in the CT and the ST contrasts, which test for the effects of chunk-level and sequence-level transitions.

The results of these analyses are presented in Table 2.2. As is apparent, they confirm that matching chunk grammars indeed lead to performance benefits. For RTs matching effects were limited to transition points. For errors, there were specific effects for the highest-level sequence transition as well as an overall, position-unspecific matching effect.

Table 2.2. Fixed effects from linear mixed models for Experiment 1.

	RT				Error			
	B	SE	t	p	B	SE	z	p
Match	-0.061	0.005	-11.06	<0.001	-0.291	0.054	-5.42	<0.001
CT	0.092	0.003	29.07	<0.001	0.062	0.029	2.17	0.030
WC	0.011	0.004	2.99	0.003	0.006	0.034	0.19	0.850
ST	0.134	0.005	25.70	<0.001	0.249	0.046	5.41	<0.001
Match * CT	-0.065	0.005	-11.83	<0.001	0.000	0.054	0.01	0.994
Match * WC	-0.002	0.006	-0.29	0.771	0.101	0.064	1.58	0.114
Match * ST	-0.059	0.009	-6.58	<0.001	-0.221	0.086	-2.56	0.010

Note. Match = matching grammars within sequence. CT = chunk transition contrast: first and fourth elements vs. all others (second, third, fifth, and sixth). WC= within-chunk contrast: second and fifth elements vs. third and sixth. ST = sequence transition contrast: first vs. fourth element.

As a second step, we examined whether the grammar matching effect develops gradually through experience with a given sequence, or instead is present from the outset. For this purpose, we added within-block sequence repetitions as an additional predictor in the form of linear and quadratic polynomial contrasts that were allowed to interact with the chunk match variable. No statistically significant effects were obtained. In particular, the interactions between the repetition contrasts and the chunk match factor were reliable for neither RTs (linear t=-.57, p=.57) nor errors (linear z=.66, p=.51). As can be seen in Figure 2.3, the matching effects were

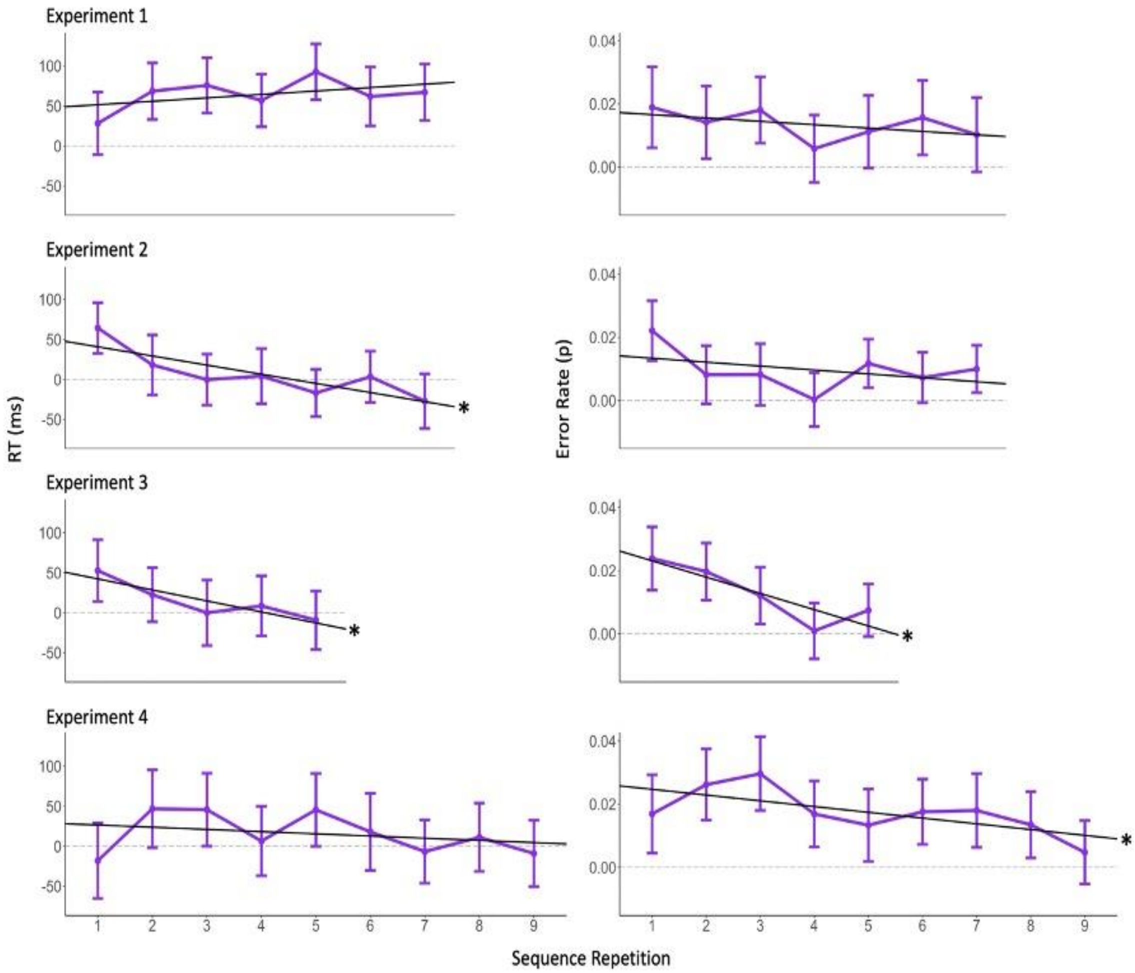


Figure 2.3. Difference scores in RTs (left panels) and error rates (right panels) for matching versus non-matching sequences across repetitions in all experiments (calculated as nonmatching RT – matching RT). The gray dashed line marks zero (no difference between matching and non-matching). The regression line is shown in black, with an asterisk to the right indicating significance (p<.05). Error bars represent 95% within-subject confidence intervals.

present from the beginning. This pattern of results is most compatible with the assumption that the cognitive system encodes sequences in a relational manner from the outset.

3. Experiment 2: Different-Element Chunks

Experiment 1 showed strong effects of abstract inter-chunk relationships. However, a limitation of this experiment is that the same elements occurred across all chunks. The use of shared elements may have facilitated detection of abstract relationships and it might have also produced particularly strong matching effects as relationships could be used to directly translate one part of the sequence into the next. To address the question of whether and to what degree abstract patterns could be utilized across chunks with non-shared elements, we conducted a second experiment in which the two chunks in each sequence contained different elements. In this experiment, each sequence contained one chunk of horizontal/vertical rules (like in Experiment 1) and one chunk of clockwise/counterclockwise rules. Thus, in this experiment, abstraction across the identity of sets of elements was required to in order to capitalize on the matching chunk patterns.

3.1. Methods

Participants (n=40) completed 36 blocks of 48 trials (36 six-element sequences, each repeated 8 times) in a 1.5-hour session. Two subjects did not complete the study and were therefore excluded, and the data from 38 subjects were used in analyses. Exclusion criteria and analysis plan were the same as in Experiment 1.

3.2. Results

Figure 2.2 (panel 2) shows the results of this experiment. While there was no effect of shared chunk pattern on RTs, there was a small but highly robust effect on error rates (Table 2.3). Though there was a slight, non-significant

Table 2.3. Fixed effects from linear mixed models for Experiment 2.

	RT				Error			
	B	*SE*	*t*	*p*	*B*	*SE*	*z*	*p*
Match	-0.004	0.005	-0.92	0.359	-0.219	0.049	-4.48	<0.001
CT	0.226	0.003	82.26	<0.001	-0.001	0.027	-0.02	0.981
WC	0.007	0.003	2.27	0.023	0.159	0.031	5.18	<0.001
ST	0.045	0.004	10.00	<0.001	-0.127	0.044	-2.92	0.004
Match * CT	-0.008	0.005	-1.72	0.085	0.103	0.049	2.10	0.036
Match * WC	-0.007	0.005	-1.28	0.202	-0.100	0.059	-1.71	0.088
Match * ST	0.002	0.008	0.31	0.754	0.041	0.078	0.53	0.599

Note. Match = matching grammars within sequence. CT = chunk transition contrast: first and fourth elements vs. all others (second, third, fifth, and sixth). WC= within-chunk contrast: second and fifth elements vs. third and sixth. ST = sequence transition contrast: first vs. fourth element.

tendency for matching benefits at chunk transitions for RTs, there was a counteracting effect specific to chunk transitions for errors — a pattern that may indicate a speed-accuracy tradeoff. No matching effects on the sequence transition level were observed (i.e., the ST contrast). Overall, the matching effects were much more subtle than in Experiment 1, consistent with the interpretation that in the first experiment, shared elements may have facilitated the detection and effective utilization of abstract relationships.

As in Experiment 1, we again examined how the matching effects developed across sequence repetitions. Interestingly, the pattern of repetition effects was, if anything, counter to a gradual learning hypothesis. For RTs, there was a statistically robust linear decrease of matching benefits across repetitions (t=3.51, p<.001); for errors there was no reliable linear trend (z=-.27, p=.79). We will discuss the unexpected pattern of diminishing matching effects in concert of the results in the remaining experiments in the General Discussion (Section 6). However, we can conclude that again, results are clearly not consistent with a gradual learning perspective.

4. Experiment 3: Different-Element Chunks with Auditory Instructions

Experiment 2 provided evidence for small, but robust matching benefits. However, a potential concern with the procedure of the first two experiments is that the visual presentation of each rule sequence prior to the block may have facilitated the detection of abstract patterns among the simultaneously visible, individual elements in the instruction screen. Therefore, in Experiment 3, we checked whether we could replicate the matching effect when sequence positions were presented in a serial, auditory manner.

4.1. Methods

Participants completed 36 blocks of 36 trials (36 six-element sequences, each repeated 8 times) in a 1.5-hour session. For this version, data were collected from 45 subjects. Four subjects did not meet the trial-wise inclusion criteria and were therefore excluded (41 subjects were kept for analysis). Exclusion criteria and analysis plan were the same as in Experiments 1-2.

The same sequence construction was used here as in Experiment 2. However, the instruction and error screens were different. Instead of displaying a sequence of rules on the screen at the beginning of each block, participants were shown a screen indicating that they should listen to the sequence of rules in their headphones. The six rules were presented one at a time for 1 second each, using a female text-to-speech voice. Participants could replay the sequence as many times as they wanted before pressing a key to begin the block.

Like in the previous experiments, after an incorrect response, the current rule would be displayed in red above the frame, in its position within the sequence. However, the rest of the rules in the sequence were represented by white dashes to indicate position without allowing multiple rules in the sequence to be presented visually at the same time. Participants were then required to correct their mistake before moving on to the next trial. Because the auditory presentation format took extra time, we reduced the number of sequence cycles per block from 8 to 6 repetitions in order to fit the entire procedure within the experimental session.

4.2. Results

As shown in Figure 2.2 (panel 3), as in Experiment 2, results of this experiment showed small and robust effects for error scores. In addition, we found robust RT matching benefits overall, as and additional befit for chunk transitions, which was somewhat offset by non-significant, counteracting error effect (Table 2.4). Again, no matching effects on the level of sequence transitions were observed.

Experiment 3 also replicated the unexpected interaction between cycle repetitions and matching benefits found in Experiment 2 (Figure 2.3). Here, for both RTs (t=4.43, p<.001) and error rates (z=2.88, p=.004), the effect of matching chunk patterns was strongest for earlier rounds of sequence execution and then dissipated linearly.

Table 2.4. Fixed effects from linear mixed models for Experiment 3.

	RT				Error			
	B	SE	t	p	B	SE	z	p
Match	-0.011	0.005	-2.02	0.043	-0.257	0.051	-5.01	<0.001
CT	0.255	0.003	80.46	<0.001	0.026	0.028	0.95	0.343
WC	0.014	0.004	3.79	<0.001	0.114	0.032	3.54	<0.001
ST	0.023	0.005	4.39	<0.001	-0.162	0.045	-3.60	<0.001
Match * CT	-0.015	0.005	-2.66	0.008	0.089	0.051	1.73	0.084
Match * WC	-0.005	0.006	-0.81	0.418	-0.051	0.062	-0.83	0.409
Match * ST	-0.003	0.009	-0.36	0.722	0.047	0.082	0.58	0.564

Note. Match = matching grammars within sequence. CT = chunk transition contrast: first and fourth elements vs. all others (second, third, fifth, and sixth). WC= within-chunk contrast: second and fifth elements vs. third and sixth. ST = sequence transition contrast: first vs. fourth element.

5. Experiment 4: Second-Order Matches

In this final experiment, we wanted to test the limits of the small, but robust abstract matching effect we had obtained in the previous two experiments with non-overlapping chunks. Specifically, we examined whether or not participants are able to benefit from second-order relationships. For this purpose, we created 12-element sequences constructed from two six-element subsequences containing two three-element chunks each. We will refer to these subsequences as “chunk plans”. The three-element chunks within each chunk plan used the same element pairs, as in Experiment 1. However, there were no shared elements across the two chunk plans. The critical match here was between the structures of the two chunk plans. Matching sequences had plans with similar chunk structures (e.g., A-B-A—B-B-A——C-D-C—D-D-C); non-matching sequences had plans with dissimilar chunk structures (e.g., A-B-A—B-B-A——D-D-C—C-C-D. Importantly, chunk plans were constructed from pairs of three-element chunks with no direct chunk pattern matches. Therefore, pattern matches on the cross-plan level were not confounded with any potential matches on the within-plan level (which were the focus of the preceding experiments).

5.1. Methods

Participants (n=44) completed 12 blocks of 120 trials (12 long sequences, each repeated 10 times) in a 1-hour session. Four subjects did not complete the study and were therefore excluded, and the data from 40 subjects were used in analyses. Exclusion criteria were the same as in Experiments 1-3, and analysis plan was similar, with a few necessary differences, described below.

5.2. Results

To reflect the more complex, hierarchical structure of the sequences used in this experiment, in addition to the CT contrast (positions 1, 4, 7, and 10, vs. rest), we used a plan transition contrast (PT) comparing positions 1 and 7 vs. 4 and 10, and a sequence-level transition contrast (ST) testing position 1 against position 7.

Again, Figure 2.2 (panel 4) and Table 2.5 show the results. In both RTs and errors, we found clear evidence for matching benefits. For RTs, these effects were slightly larger at plan and chunk transition points. On the level of sequence transitions, there seemed to be opposing tendencies between RTs and errors, again potentially reflecting a speed-accuracy tradeoff. Overall, these results indicate that even high-

Table 2.5. Fixed effects from linear mixed models for Experiment 4.

	RT				Error			
	B	SE	t	p	B	SE	z	p
Match	-0.030	0.007	-4.34	<0.001	-0.420	0.068	-6.16	<0.001
CT	0.088	0.002	46.18	<0.001	-0.091	0.016	-5.59	<0.001
PT	0.135	0.004	33.62	<0.001	-0.115	0.034	-3.37	0.001
ST	0.026	0.008	3.10	0.002	-0.055	0.078	-0.71	0.481
Match * CT	-0.012	0.003	-4.29	<0.001	-0.016	0.025	-0.63	0.528
Match * PT	-0.012	0.006	-2.18	0.029	-0.052	0.054	-0.96	0.336
Match * ST	-0.017	0.012	-1.43	0.153	0.262	0.124	2.12	0.034

Note. Match = matching grammars within sequence. CT = chunk transition contrast: first, fourth, seventh, and tenth elements vs. all others. PT = chunk plan transition contrast: first and seventh elements vs. fourth and tenth. ST = sequence transition contrast: first vs. seventh element.

level abstract relationships (i.e., relationships between relationships) are appreciated during sequence production.

Additionally, we again replicated the unexpected pattern of dissipating matching benefits as a linear function of sequence repetitions, but only for error scores (t=2.09, p=.036), and not for RTs (t=.61, p=.541).

6. General Discussion

Across four experiments, we tested whether abstract chunk relationships are registered and used by the cognitive system. Past research has provided evidence for the utilization of abstract patterns but was mostly limited to situations in which sequential elements had inherent ordering properties that might support the detection of patterns and where detection of patterns could be accomplished through gradual, feedback-based hypothesis testing and learning. We were therefore interested in determining the degree to which utilization of abstract sequential patterns is a general property of sequential representations. An affirmative answer to this question would have important theoretical implications for current models of hierarchical control. Such models typically assume that the hierarchical organization of sequential representations supports a divide-and-conquer strategy, in which different chunks of a sequence assume isolated representational subspaces in order to avoid between-chunk interference. Therefore, it is not clear how such models could at the same time account for the across-chunk integration that would be necessary to utilize abstract relationships.

6.1. Utilization of Abstract Relationships

Answering our primary research question, we found across four different experiments robust evidence that abstract relationships between different parts of a sequence do indeed lead to performance benefits. Importantly, in our paradigm the basic elements were response rules (e.g., horizontal vs. vertical), which implied that abstract grammars could only be based on patterns of repetition across elements (e.g., AAB vs. ABA vs. ABB) instead of sets of simple algebraic operations. Thus, these results confirm that the utilization of abstract relationships can be generalized beyond stimuli with inherent ordering properties, such as numbers or spatial locations. The pattern of element repetitions is an important indicator of potential ordering principles and therefore it makes sense that our cognitive system would be particularly attuned to detecting and utilizing such patterns (Dehaene et al., 2015). It is also noteworthy that participants were able to utilize even second-order abstract relationships, that is when the relationships between two chunks in the first half of the sequence matched the relationship between the two chunks in the second half of the sequence (despite different elements across the two sequences).

6.2. The Role of Sequence Positions

For our second research question, we wanted to clarify if abstract relationships affect all sequential elements, or instead are particularly expressed during chunk transitions (e.g., at the beginning of a chunk or chunk plan). One could argue that the latter result might indicate that abstract relationships are utilized only during the brief chunk transition period, when both chunks are activated. For all four experiments, we found robust error benefits that extended across all sequence elements. However, Figure 2.2 also clearly indicates that the results of Experiment 1 stand out from those of the remaining experiments. That experiment used shared elements across chunks and showed a very strong RT benefit for matching chunks at transition points. While interactions between the match factor and transitions contrasts were also found in the remaining experiments, these were much more subtle and in part offset by speed-accuracy effects.

What may explain the particularly large transition benefits in Experiment 1? We suggest that for matching chunks with shared elements, the abstract relationship can be used to directly translate the preceding chunk in the following chunk through an "alternation" operation, where every element in chunk 1 is turned into the alternative possible element in chunk 2. This eliminates any need for retrieving individual elements and placing them into their "slots," as is necessary when elements are not shared. In that sense, the result of Experiment 1 is very similar to the body of results from earlier work with abstract relationships in sequences with ordered elements (Restle, 1970; Restle & Burnside, 1972). Here too, the abstract relationships could be used to directly transform one part of a sequence into the next through the various possible transformation rules, leading to strongest benefits at transition points. However, beyond this direct translation effect during transitions, our results consistently show a second, more subtle matching benefit that affects all sequential elements equally. Importantly, this would imply that information seeps across chunks, not only at transition points, allowing a more efficient representation of the entire sequence.

6.3. The Role of Within-Sequence Experience

The third question we wanted to address is particularly important. There is previous evidence that people (and even infants) are able to capitalize on abstract sequential patterns (Aslin & Newport, 2012; Marcus et al., 1999). These results stem from learning paradigms that allowed the gradual discovery of "hidden" relationships. However, it is also possible that our cognitive system is geared towards encoding sequential material from the outset in terms of relationships between elements and sequence parts. If that is the case, we should see abstract pattern benefits emerging from the beginning, rather than gradually as a function of repeated exposure with the same sequence. Our results were very clear in this regard. We found that abstract benefits were present from the beginning in all of our experiments. Further, in each of Experiments 2-4, we found for at least one of the two dependent variables the opposite of what would be predicted by gradual learning, namely a decrease of the matching benefit (see Figure 2.3). Indeed, for Experiment 2 and 3 we found a reduction of an initial RT benefit—a result that also makes the finding of overall matching effects only in errors and not in RTs a bit less surprising.

While unexpected, the “reverse” learning effect is consistent with an “expectation of relationships” hypothesis. To see why, it is important to appreciate that even our non-matching sequences could allow inter-chunk relationships to be established. For example, a sequence such as ABB-DDC can be described as a mirroring of the initial abstract order. Thus, while our matched sequences allowed participants to pick up the relationships from the outset, the cognitive system might need more experience (i.e., a few cycles) with each new sequence to detect and utilize the more complex or subtle relationships present in our “non-matching” conditions.

This perspective also shines light on two other aspects of our results. In Experiments 2-4, our matching effects were subtle, and one might wonder how seriously one should take such small effects as a marker of an important principle of sequential representations. However, if the cognitive system is geared towards constructing relationships even when they are not obvious, it should be difficult or impossible to create an all-or-none contrast between sequences that contain abstract relationships and those that do not. The actual contrast we can establish will be much more graded and therefore subtle effects should not be a surprise.

The second and related observation is that, arguably, the matching effects from our experiments with non-shared elements were strongest in Experiment 4, which used the most difficult abstract relationships. It may appear odd that second-order abstract relationships produced larger effects than first-order relationships. However, again, the critical question concerns how difficult it is to establish relationships between matching plans compared to the non-matching control condition. It should be particularly hard to construct second-order, abstract relationships when they are not readily apparent, creating a particularly stark matching contrast in Experiment 4. We acknowledge that this aspect remains somewhat speculative, as there is no principled, a-priori way for determining which relationships are simple or hard for our cognitive system to detect and utilize. However, we believe it is plausible to assume that creating a first-order relationship between two "non-matching" chunks such as ABB and CCD is easier than constructing a second-order relationship between two "non-matching" chunk plans such as ABB-BBA and CCD-CDC.

6.4. Theoretical Implications

In this final section, we turn to the issue how our results fit with current models of hierarchical, serial-order control. One long-standing question in this context is to what degree serial order is established through some form of associative chaining between specific elements versus through abstract, element-independent position codes. There is evidence that through statistical learning of successive element relationships, abstract patterns can be extracted. However, such learning occurs gradually, across a number of repetitions (Botvinick & Plaut, 2004; Davachi & DuBrow, 2015). Therefore, the finding that abstract patterns for new sequences can be extracted and used from the outset clearly favors the abstract-position account (see also, Kikumoto & Mayr, 2018; Mayr, 2009).

The second, broader question is how our results that indicate integration of information across chunk boundaries can be reconciled with the idea that hierarchical control implements a divide and conquer approach towards resolving interference across different regions of a complex task space. Recent models of hierarchical control are informed by neuroimaging work indicating distinct neural substrates for different hierarchical levels (Badre, 2008; Badre & Nee, 2018; Koechlin et al., 2003; Koechlin & Jubault, 2006). Functionally, most models share the general idea that elements on each level of the hierarchy are competing with each other through a winner-take-all mechanism. This would ensure that only one chunk can be active at a time. The sequence of chunks should be represented (on the level above) through content-free pointers or symbols that themselves contain no information about the chunk content and therefore should not allow the detection of between-chunk relationships (Fitch & Martins, 2014).

Within this framework, there are two ways to allow for integration to occur. The first is to give up on the idea that insulation between different chunks on the same level is complete. In fact, there is substantial evidence of interference across competing chunks. Most notably in this regard is evidence of transposition errors, where an element occurring at a certain position within chunk 1 is more likely to be incorrectly inserted into the same position in chunk 2 than another position within the same chunk (Henson et al., 1996; Mayr, 2009). It is possible that such interference errors are a necessary sacrifice our cognitive system needs to accept in order to allow the passing of information across chunks that makes the detection of between-chunk relationships possible.

The second option for allowing between-chunk integration is to give up on the idea of strict between-level insulation. In the current case, this would mean that on the level above chunks, individual chunks are not just represented through content-free labels or pointers, but instead contain sufficient information about what happens on the level below to allow the utilization of relationships. Specifically, on the above-chunk level, the abstract grammar relevant for each chunk may be part of the represented information, allowing the detection of relationships, while avoiding element-specific interference (e.g., AAB could be represented as "repeat, then alternate").

These two options are not mutually exclusive. However, of the two, the second appears to be somewhat more plausible as an explanation for the type of immediate utilization of abstract patterns that we observed in our data. Arguably, the type of across-chunk information leakage apparent in transposition errors seems better suited for allowing a more gradual detection of structure and relationships.

Overall, our results are fully consistent with the view that the cognitive system comes with a built-in prior for expecting that sequences can be coded in terms of relationships between successive elements and chunks. This conclusion is very similar to those by Restle (1970) and others, but extends it to sequences without ordered elements and to a paradigm without feedback-driven learning and hypothesis testing. Recent models of hierarchical control have emphasized the question how our cognitive system is able to divide complex tasks into separable subspaces. The current results suggest that it is important to also account in these models for our ability to integrate useful information across such separate subspaces.

Acknowledgements

This research was supported by NSF grant 1734264 (PI: Mayr).

CHAPTER III

WHAT'S SO HARD ABOUT HIERARCHICAL CONTROL? PINPOINTING PROCESSING CONSTRAINTS WITHIN CUE-BASED AND SERIAL-ORDER CONTROL STRUCTURES

From Moss, M.E. & Mayr, U. (2022). What's so hard about hierarchical control? Pinpointing processing constraints within cue-based and serial-order control structures. [Submitted for publication]. Department of Psychology, University of Oregon.

1. Introduction

Everyday problem-solving or planning situations usually involve task spaces with a hierarchical structure in which each decision depends on an earlier-made decision on a higher level, which in turn might depend on still higher-level decisions (Collard & Povel, 1982; Cooper & Shallice, 2006; Fitch & Martins, 2014; Kikumoto & Mayr, 2018; Lashley, 1951; Logan & Crump, 2011; Miller et al., 1986; Restle, 1970; Rosenbaum et al., 1983). Operating within such spaces is difficult. Performance in terms of both speed and accuracy typically declines as a function of the number of hierarchical levels that need to be considered, which we refer to here as the *number-of-levels* effect. In fact, the ability to efficiently navigate hierarchically organized control structures or decision trees has been suggested as a key feature of human intelligence (e.g., Carpenter et al., 1990; Marshalek et al., 1983). Yet, the question of why hierarchical control is hard does not yet have a clear answer. To make progress on this question, we need to investigate how exactly the cognitive system deals with hierarchical control demands.

1.1. Costs of Ballistic Updating

In principle, when hierarchical control is applied to complex actions or decisions, different hierarchical levels should occupy distinct representational subspaces, offering protection of lower-level decisions from resource demands on higher levels of the hierarchy. In its pure version, this account assumes that each level "programs" the next lower level in a recursive, ballistic manner (Miller et al., 1986; Rosenbaum et al., 1983). Once programmed, no further performance costs associated with that level will be incurred until settings on that level need to be updated again. To provide a concrete example: In one version of our paradigm, participants needed to apply one of two possible S-R mappings to a given stimulus based on up to three additional hierarchically organized context rules (see Figure 3.1a). Each higher-level rule determines how to use information one level down, and all rules are signaled through distinct

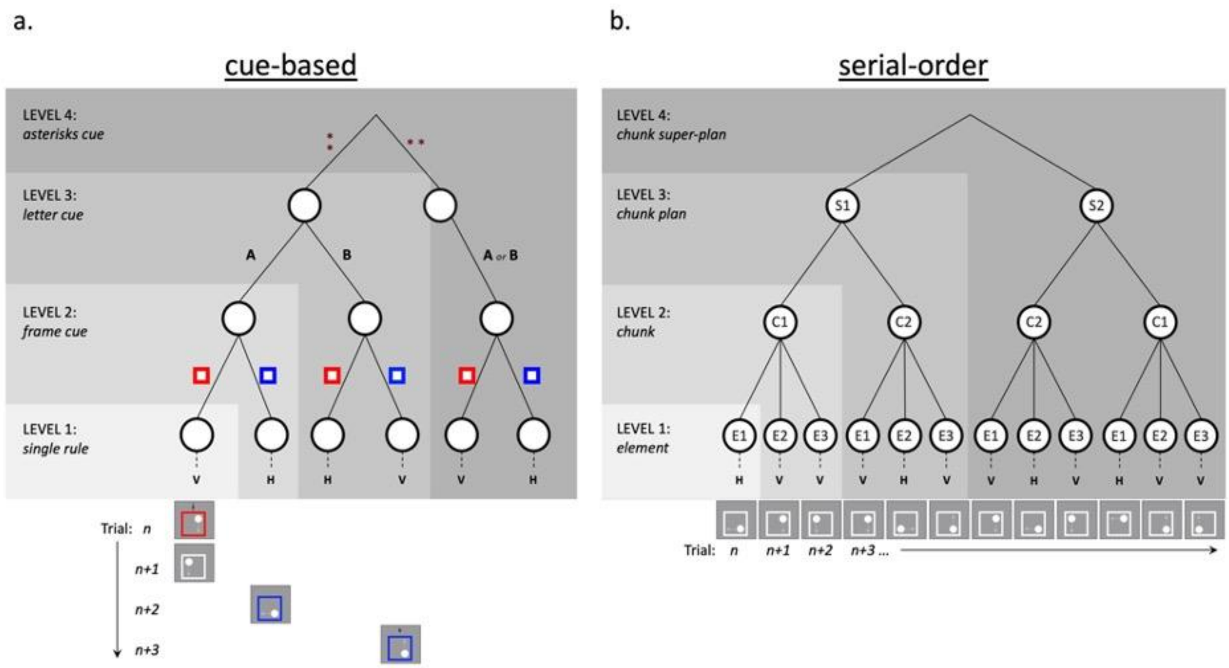


Figure 3.1. Diagrams of the hierarchical rule structure in (a) cue-based format and (b) serial-order format. Example set of trials for structural level 4 are shown below each diagram. The arrow within the stimulus frame indicates the correct response and was not presented in the actual trial.

visual cues. The ballistic updating account predicts that performance costs increase with the number of hierarchical levels, but only on trials in which relevant cues are actually presented. For trials during which no level-relevant cues are shown, the most recently used rule should simply carry over from previous trials and ballistically influence current processing, without additional costs.

Broadly consistent with such a conceptualization, recent neuroimaging evidence indicates that organization of frontal cortex supports distinct neural resources for different hierarchical levels of control (Badre, 2008; Badre & D'Esposito, 2009; Koechlin et al., 2003). However, this neuroanatomical information does not provide strong empirical constraints on how functional limitations might arise when operating in hierarchical control structures. In addition, more recent neuroimaging results have raised questions about a clean, neuroanatomical segregation along hierarchical levels (Badre & Nee, 2018; Crittenden & Duncan, 2014; Farooqui et al., 2012; Yokoi & Diedrichsen, 2019).

1.2. Global Costs of Maintaining a Hierarchical Structure

It is also possible that multi-level control structures induce performance costs because the current state on each level needs to be integrated and maintained within a global representational working memory space. In this case, operating with more levels would simply take up more representational resources than operating with fewer levels (Waltz et al., 1999). For example, Duncan (2010) has argued that prefrontal cortex provides a “multiple-demand network” that allows ad-hoc representations of current task requirements. Indeed, single-cell recording work has indicated that many prefrontal neurons code for task-relevant features in an ad-hoc manner, and in particular for nonlinear combinations of such features (Rigotti et al., 2013). The more hierarchical levels exist in a structure, the more neural resources are required to represent the current state on each level of the hierarchy, as well as the across-level combinations of states. As a result, fewer resources would be available to perform any operation within such a structure (Duncan et al., 2008). Importantly, having neural resources occupied should induce global costs that affect any operation within the control structure, no matter on which level it is occurring and whether or not that particular decision requires consideration of levels beyond the current decision level.

In the example of our cue-based paradigm (Figure 3.1a), such global costs would manifest as number-of-levels effects across all trials, regardless of when relevant cues are presented. Specifically, costs should increase with the number of hierarchically organized context rules that could be relevant in a given block, even in trials during which no cues are presented—and not just in updating trials, as predicted by the ballistic model.

One example of an empirical pattern consistent with such global integration costs comes from research using the task-switching paradigm (Monsell, 2003). Specifically, operating within task-switching blocks (i.e., two-level control structures) induces global costs that are apparent even if no change in task is required on a given trial. Interestingly, these costs are particularly large in groups or individuals thought to have reduced cognitive control resources, such as older adults and people with lower fluid intelligence (Kray & Lindenberger, 2000; Mayr, 2001; Mayr & Kliegl, 1993; Wasylyshyn, Verhaeghen, & Sliwinski, 2011). In principle, global costs in the task-switching situation could arise from fitting the two-level control structure within the same representational space that also needs to handle the next upcoming decision.

1.3. Dynamic Updating/Integration

The ballistic updating model and the global maintenance model describe two possible extremes. In the former, costs emerge in a strictly local manner, only for those aspects that actually need to be updated. In the latter, any aspect or dimension that *could* change in a given context (even if it needs no updating on a specific trial) will add to overall processing costs. However, there is an intermediary possibility that we refer to here as the dynamic integration model. Here, we assume that a lower-level decision (e.g., associated with a level 2 cue) needs to be integrated with previously established higher-level settings (e.g., associated with a level 3 cue in the case of a level 2 decision) in a costly manner. However, when several higher-level settings are involved (e.g., levels 3 and 4, in the case of a level 2 decision), in principle these could be encoded dynamically into a single combined rule. For the lower-level decision, only the combined rule needs to be consulted and integrated with the current decision, instead of retrieving the entire constellation of individual higher-level cues that make up the combined rule. This leads to the prediction that costs arise whenever a specific level needs to be updated (as in the pure ballistic model), with a further increase in costs if there is at least one additional level to be considered, but no more increase in costs for potential additional levels.

1.4. Cue-Based versus Serial-Order Control

In past research, hierarchical task spaces have been implemented either through signals in the environment (e.g., Badre et al., 2009; Koechlin et al., 2003) or in the form of memorized, serial-order plans (Kikumoto & Mayr, 2018; Schneider & Logan, 2006). However, these two modes of inducing hierarchical control demands have not yet been directly contrasted with each other in terms of their processing requirements.

In recent neuroimaging research, participants' trajectories through hierarchical control structures have been guided mostly through cues (Badre & D'Esposito, 2007; Koechlin et al., 2003). Specifically, participants were typically instructed to respond to stimuli in a manner that could depend on the state of one or more additional cues (depending on the number of levels). Thus, the environment triggers the need to update an internal state. As mentioned above, in this design, it is important to include a condition in which no state-relevant cues are presented. In such a no-cues condition, no updating is necessary, and therefore in principle, the system can continue to operate on the basis of the most recently established setting. This condition is particularly useful to distinguish between the ballistic model, the global integration model, and the dynamic integration model, which each make different predictions for number-of-levels effects when no cues are presented (see Figure 3.2).

In motor-control and executive-control research (Collard & Povel, 1982), there is a long history of inducing hierarchical control demands through ordered internal plans. For example, in the task-span procedure, participants might be instructed to memorize and then perform a short subsequence of tasks, which in turn might be combined into a higher-level sequential plan of subsequences (Kikumoto & Mayr, 2018; Mayr, 2009; Schneider & Logan, 2006) Here, no external signals are provided to indicate whether internal states need to be changed or not (e.g., when moving from

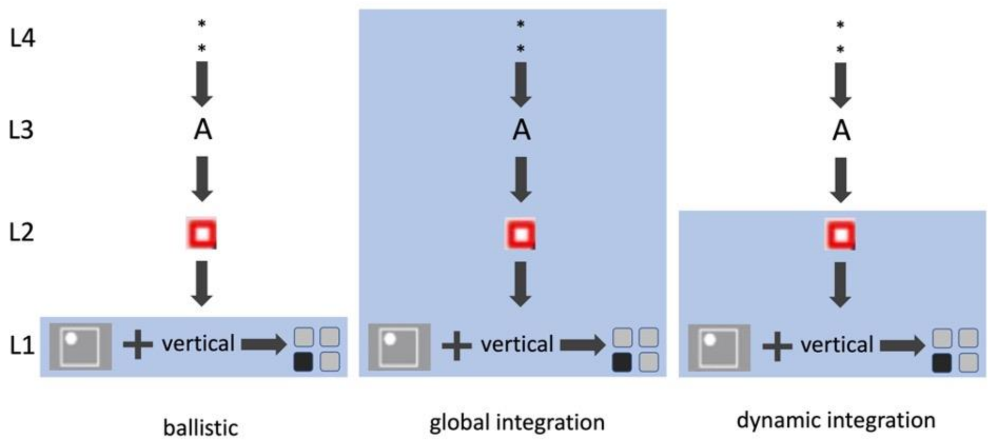


Figure 3.2. Three possible hierarchical control scenarios. The example shows a situation with four different levels that have been established through cues on previous trials. The current trial contains no cue except for the response-relevant stimulus (i.e., cue level 1 within a 4-level structure). In the ballistic account, the higher-level settings have set up a “protected” representational space, within which the low-level process can be executed no matter how many higher-level settings were initially involved. Thus, there should be performance costs at updating points, but no number-of-levels effects on no-cue trials (cue level 1). In the global integration account, no “protective” space exists, so information about all involved settings must be maintained and integrated on every trial, regardless of whether that information requires updating. Here, number-of-levels effects occur on all trials, including no-cue trials. In the dynamic integration account, information from all higher-level rules is collapsed into one setting that directly impacts the lowest-level decision. Here, number-of-levels effects on no-cue trials extend one level above the current decision level. In the example shown, this implies that the currently valid “vertical rule” is activated, but not the specific constellations of higher-level cues that has been put into place. In contrast, if this were a single-level structure with the vertical rule valid throughout, that rule would simply be executed without any additional reconsideration.

one subplan to the next). On the one hand, classic models of serial-order control, mainly in the motor domain, assume that longer sequences are broken into smaller chunks, each of which can be performed ballistically, in an autonomous manner. Therefore, updating costs should arise only when transitioning to a new chunk (i.e., when changing the current chunk). On the other hand, one might argue that the current position within a sequence is the sole internal signal for potential state changes. Thus, just to keep track of the sequence position, information across all levels/dimensions may have to be integrated at each new position, leading to the expectation of number-of-levels effects on every trial.

1.5. Paradigm and Predictions

How do these considerations translate into specific predictions within our paradigm? As mentioned in Section 1.1, we manipulated the number of hierarchical levels block-wise, separately within a cue-based and a serial-order format (see Figure 3.1a-b). For both formats, participants had to use one of two different task rules on each trial (e.g., applying either a vertical or horizontal spatial translation to a stimulus). The varying number of additional hierarchical levels specified which of the two rules was relevant on a given trial.

In order to separate the control-relevant effects of cues from their perceptual/attentional influences, we also manipulated the number of cues across the entire range for all cue-based conditions. Thus, on some trials, cues were presented that were not relevant in that block. More importantly, all cue-based blocks also contained trials without any cues, which therefore required no cue-based updating.

We formalized our predictions from the three models through sets of model variables that specified the presence of the hypothetical processing constraints for each trial type, separately for the two formats. Tables 3.1 and 3.2 contain the model matrices relevant for the cue-based and serial-order formats, respectively. In the remainder of Section 1, we describe each set of model variables in turn, for both cue-based (1.5.1-1.5.4) and serial-order (1.5.5-1.5.7) formats.

1.5.1. Cue-Based Ballistic

Here, costs are predicted on trials with cues that correspond to the currently relevant level. Note that as we describe the modeling of cue-based costs, it is important to distinguish between the hierarchical structure of a given block, which can be level 1, 2, 3, or 4 (structural level), and the number of cues presented on a given trial regardless of the hierarchical structure, which can also be level 1, 2, 3, or 4 (cue level). To be concrete, for a level 3 structure, the presence of a level 2 cue requires updating on level 2. When both level 2 and level 3 cues are shown, first level 3 and then level 2 need to be updated, leading to further costs. However, the addition of a level 4 cue should produce no additional updating costs because that level is not relevant, though it may lead to additional perceptual/attentional costs. Importantly, the notion that control settings are ballistic also indicates that the effects of lower-level updating are identical, regardless of whether these occur within a level 2 structure, or within a more complex structure (e.g., level 3 or 4). In other words, lower-level decisions should not be more demanding just because higher-level settings are also relevant within a given block. Finally, note that no matter how many structural levels are relevant in a given block, no-cue trials (i.e., cue level 1) are coded here as zeros (see Table 3.1), indicating that there should be no number-of-levels effect when there are no updating demands.

1.5.2. Cue-Based Global Integration

Here, the prediction is that each potentially relevant structural level in a block will add costs, regardless of whether updating is required on a given trial. The difference between this model and the ballistic model becomes particularly apparent in the no-cue trials (i.e., cue level

Table 3.1. Variables used in the three models of performance in the cue-based format, at each level and cue level.

		Ballistic Model			Global Integration Model						Dynamic Integration Model						
Level	Cue Level	L2 B	L3 B	L4 B	L2 B	L3 B	L4 B	L2 G	L3 G	L4 G	L2 B	L3 B	L4 B	L2 D	L3 D	L4 D	F
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**2**
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**
2	1	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0
	2	**1**	0	0	**1**	0	0	**1**	0	0	**1**	0	0	**1**	0	0	0
	3	**1**	0	0	**1**	0	0	**1**	0	0	**1**	0	0	**1**	0	0	**1**
	4	**1**	0	0	**1**	0	0	**1**	0	0	**1**	0	0	**1**	0	0	**2**
3	1	0	0	0	0	0	0	**1**	**1**	0	0	0	0	**1**	0	0	0
	2	**1**	0	0	**1**	0	0	**1**	**1**	0	**1**	0	0	**1**	**1**	0	0
	3	**1**	**1**	0	**1**	**1**	0	**1**	**1**	0	**1**	**1**	0	**1**	**1**	0	0
	4	**1**	**1**	0	**1**	**1**	0	**1**	**1**	0	**1**	**1**	0	**1**	**1**	0	**1**
4	1	0	0	0	0	0	0	**1**	**1**	**1**	0	0	0	**1**	0	0	0
	2	**1**	0	0	**1**	0	0	**1**	**1**	**1**	**1**	0	0	**1**	**1**	0	0
	3	**1**	**1**	0	**1**	**1**	0	**1**	**1**	**1**	**1**	**1**	0	**1**	**1**	**1**	0
	4	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0

Note. L = Level. B = ballistic variable. G = global integration variable. D = dynamic integration variable. F = filtering variable. The filtering variable was included as a predictor in each of the three models. The Ballistic model predictors are also included as the first three columns for the Global Integration and Dynamic Integration models.

1), where number-of-levels effects are expected, even in the absence of any updating demands. We conceptualized this as a potential additional source of costs in the global integration model, over and above the updating costs described in the previous section (1.5.1). This means the ballistic updating model is nested within the global integration model, and therefore, we can statistically test the degree to which the global integration model produces a better model fit than the ballistic updating model.

1.5.3. Cue-Based Dynamic Integration

This model assumes that costs over and above pure updating emerge whenever a "next higher" structural level above the level of the currently cued decision could be relevant, but any additional structural levels beyond that do not incur additional costs. Here again, the zero-cue (cue level 1) condition is particularly diagnostic. In contrast to both the ballistic updating and global integration models, this model predicts an increase in costs for structural level 2 (compared to structural level 1), but not for levels 3 and 4. Again, this is an additional source of

Table 3.2. Variables used in the three models of performance in the serial-order format, at each level and sequence position.

			Ballistic Model			Global Integration Model						Dynamic Integration Model					
Level	Position Level	Sequence Position	L2 B	L3 B	L4 B	L2 B	L3 B	L4 B	L2 G	L3 G	L4 G	L2 B	L3 B	L4 B	L2 D	L3 D	L4 D
1	1	1-12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
	2	2	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	3	3	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	1	4	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
	2	5	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	3	6	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	1	7	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
	2	8	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	3	9	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	1	10	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
	2	11	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	3	12	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
3	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
	2	2	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
	3	3	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
	4	4	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0
	5	5	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
	6	6	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
	1	7	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
	2	8	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
	3	9	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
	4	10	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0
	5	11	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
	6	12	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
	3	3	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
	4	4	1	0	0	1	0	0	1	1	1	1	0	0	1	1	0
	5	5	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
	6	6	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
	7	7	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1
	8	8	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
	9	9	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
	10	10	1	0	0	1	0	0	1	1	1	1	0	0	1	1	0
	11	11	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0
	12	12	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

Note. L = Level. B = ballistic variable. G = global integration variable. D = dynamic integration variable. The Ballistic model predictors are also included as the first three columns for the Global Integration and Dynamic Integration models. All sequence positions on Level 1 were coded as 0; they were collapsed into a single row here to simplify the table.

costs over and above the basic updating costs, which means the ballistic updating model is nested within the dynamic integration model. Because the dynamic integration model includes both updating and integrating variables, the two models can be compared statistically. However, the global integration and the dynamic integration models are not nested and have the same number of predictors. In the absence of a nested relationship, we used Akaike Information Criterion

(AIC) values, as well as the constellation of statistically reliable predictors to compare relative model fits. The dynamic integration model emerged during exploratory analyses of Experiment 1, and we then tested it on a new sample in Experiment 2. The analysis plan for Experiment 2 was pre-registered to reflect this process.

1.5.4. Cue Filtering

The design used for the cue-based format allowed us to test for the effects of cues presented in each trial, even when they were not relevant for that block (i.e., higher-level cues shown in blocks with lower-level structures). Therefore, we included a "Filter" variable, which represents the number of irrelevant cues in each trial (see Table 3.1).

1.5.5. Serial-Order Ballistic

Overall, for the serial-order format, all models operate in the same way as for the cue-based format, including the nesting relationships (see Table 3.2.). One obvious difference is that the cue-filtering predictor is not relevant here. The ballistic model again predicts costs only when updating is necessary (i.e., between chunks and subsequences). For the within-chunk trials 2 and 3, no costs are predicted in this model, as updating should happen only during chunk transitions.

1.5.6. Serial-Order Global Integration

Here we expect that costs are added with every level, even when no updating is required. Thus, number-of-levels effects are expected here, even for within-chunk trials 2 and 3. These costs are incurred in addition to the updating costs described in Section 1.5.5.

1.5.7. Serial-Order Dynamic Integration

Here, we predict that costs increase during updating whenever one additional structural level is relevant, with no further increase in costs for additional relevant structural levels beyond that. For example, we expect that within-chunk positions 2 and 3 show a higher cost for structure levels 2-4 versus level 1, but there should be no cost difference between structure levels 2, 3, and 4, at the within-chunk positions.

2. Experiment 1

In Experiment 1, we also wanted to ensure that any hierarchical control effects truly reflect structural phenomena, rather than processing constraints bound to specific domains or primary task types. Therefore, we replicated the same structural manipulations across three different task pairs. Aside from the spatial rules task pair, we used a pair of perceptual odd-one out tasks (Mayr & Keele, 2000), and a pair of number judgment tasks commonly used in a task-switching context. Additionally, in Experiment 1, we used only one order of presenting the two control formats (cue-based and serial-order). In Experiment 2, we conducted a close replication of our main results for one of the three task domains (i.e., the spatial rules task), but with the order of cue-based and serial-order conditions counterbalanced across participants. Predictions for Experiment 2 were pre-registered based on the results of Experiment 1.

2.1. Methods

2.1.1. Participants

Data were collected from 26 individuals between 18 and 35 years old. Participation was voluntary, and subjects received \$10 per hour (\$25 total), plus additional accuracy-based incentives, typically between \$6 and \$9 (see Section 2.1.3 for more information on the incentives structure). No direct evidence was available to estimate effect sizes. However, related effects (e.g., single-task versus mixed-task blocks, sequence structure effects, or updating costs in cue-based hierarchical control situations) usually produce robust effect sizes for which a sample size of around 20 participants seemed sufficient.

2.1.2. Stimuli and Design

Two hierarchical control formats, cue-based and serial-order, were used, each with four possible structural levels, and each with the same set of three "primary tasks." We will first describe the primary tasks and then the implementation of the control structures.

2.1.2.1. Tasks

Across both formats, participants worked with a spatial rules task, an odd-one-out task, and a number judgment task. For the spatial rules task (adapted from Mayr, 2002), in every trial, a white circle (60-pixel diameter) appeared randomly in one quadrant of a frame (60 pixels off-center), and participants were instructed to indicate which quadrant the circle would end up in if they applied one of two possible spatial rules: horizontal or vertical. Responses were made with the right-hand index finger, using the 4, 5, 1, and 2 keys on the keyboard number pad. These keys correspond to each quadrant of the frame (top left, top right, bottom left, and bottom right, respectively).

For the odd-one-out task, a rectangle (40x75 pixels) appeared in each quadrant of a frame (60 pixels off-center), with one rectangle different in color (blue or green, vs. black for all other rectangles), and one rectangle different in pattern (vertical stripes, diagonal zig-zags, or checkers, vs. solid for all other rectangles). Participants were instructed to use the same response mapping as in the spatial rules task to indicate which rectangle was the odd-one-out, based on either the color or the pattern rule.

For the number judgment task, participants were shown a number randomly chosen on each trial from 1, 2, 3, 4, 6, 7, 8, 9 (Arial font, size 88). Numbers were presented individually, centered inside the frame. Participants were instructed to judge whether the number was lower or higher than 5 (L/H), or whether it was odd or even (O/E). Responses were made with the right index finger using the left and right arrow keys, with the left arrow key representing the judgment to the left of the slash (L or O), and the right arrow key representing the judgment to the right of the slash (H or E).

2.1.2.2. Cue-Based Control Structure

In the cue-based format, participants were presented with intermittent in-trial visual cues to indicate which task rule to use. At the lowest level (structure level 1), a single rule was prompted at the beginning of the block, and participants were instructed to apply this rule in every trial. For structure level 2, the frame color indicated which rule to use. The specific mapping between frame color and rule for each task pair was instructed at the beginning of the block. For structure level 3, the conjunction of frame color and above-frame letter cue indicated which rule to use. For structure level 4, the orientation of stars around the letter cue indicated whether to use the same rule combination as indicated in structure level 3, including the conjunction of frame color and letter cue, or to ignore the letter cue and instead, determine which rule to use based on an instructed "default" rule structure (see Figure 3.1a). We chose this arrangement on structural level 4 after pilot work revealed that a complete reversal of level 3 rules through level 4 rules was too difficult for participants.

Cue presentation was determined the same way across all structure levels, irrespective of which structural levels were relevant on a given block. When higher-level cues were displayed, all lower-level cues were displayed as well. The trial-wise probability of displaying a level 2 cue was .333. In trials with level 2 cues, the probability of also displaying a level 3 cue was .5 (.167, overall). In trials with level 3 cues, the probability of also displaying a level 4 cue was .5 (.083 overall). For trials without cues (or with lower cue level than hierarchical structure level), participants were told to use the rule indicated by the most recently displayed cue(s). After an incorrect response, a diagram of the current block's cue meanings would appear in the top right corner of the screen (similar to the instruction diagram), and participants would need to make a correct response before continuing to the next trial.

2.1.2.3. Serial-Order Control Structure

In the serial-order format, no trial-by-trial cues were provided. Instead, the relevant rule on a given trial was specified through sequences of varying hierarchical complexity. Sequences were explicitly instructed at the beginning of each block and participants "cycled through" repeatedly until the end of the block (as in (Mayr, 2009). For structure level 1, participants simply repeated the same rule across trials. Note that except for the omission of visual cues, this condition is identical to structure level 1 in the cue-based format. For level 2, three-trial sequences of rules ("chunks") were instructed, which were repeated until the end of the block (see Figure 3.1b). These chunks used one of the following possible formats: A-B-B, A-A-B, A-B-A (and the inverse of each). For level 3, two chunks were grouped into a six-element plan, while avoiding chunk repetitions (e.g., A-B-B–B-A-B). Level 4 chunk super-plans used the same basic, two-chunk plans as on level 3, but added a chunk-level reversal of that plan to create a 12-element sequence (e.g., A-B-B–B-A-B—B-A-B–A-B-B). The instruction screen for level 4 contained six rules (like on level 3), but also included a down arrow on the left side and an up arrow on the right side, indicating that participants were to execute a fourth-level sequence. After an incorrect response, the sequence of rules would be displayed above the frame, with the incorrect response colored red. For structure level 4, one of the arrows would also be red after an error, to indicate location within the 12-element sequence. Participants needed to make a correct response in order to continue on to the next trial.

2.1.3. Procedure

The total duration of the experiment was 2.5-hours. All components of the experiment were completed on the computer (24-inch display), with participants completing tasks first in the cue-based format, and then in the serial-order format. The session was bookended with a cued switching task[1], such that participants performed the first half of the switching task, then the two contexts, and then the second half of the switching task. A trained experimenter used images and examples to instruct participants before each section of the experiment. The script for the explanations of each component can be found in Appendix A. Within each of the four components, participants completed the spatial rules task, followed by the odd one out task, and then the number judgement task. In the cue-based and serial-order formats, participants completed all structure levels in a "mountain structure" (level order: 1-2-3-4-3-2-1) for each task.

Within the two formats, the number of trials varied across hierarchical level, so that participants completed more trials for each increase in complexity. For the cue-based context, participants completed 1440 total trials across the four levels: 144 trials (48 per task) on evel 1, 288 trials (96 per task) on Level 2, 432 trials (144 per task) on Level 3, and 576 trials (192 per task) on Level 4. For the serial-order context, participants completed 1296 total trials across the four levels: 108 trials (36 per task) on Level 1, 216 trials on Level 2 (18 different 3-element chunks (6 per task), repeated 4 times each), 324 trials on Level 3 (18 different two-chunk sequences (6 per task), repeated 3 times each), and 648 trials on Level 4 (18 different four-chunk

[1] We also included a standard cued switching task, for which participants were instructed to perform the same three tasks as in the cue-based and serial-order structures, but instead of receiving block-wise rule sequences or in-trial symbolic cues, one of the two rules was presented at the center of the screen for the whole trial, starting 100ms prior to stimulus onset. Rules were randomly chosen and followed no pattern or hierarchical structure. Because these results are not relevant to our main question, we are not using these data in the current paper.

super-plans (6 per task), repeated 3 times each). The switching task included 108 trials (36 per task) at the beginning of the session and another 108 trials at the end. The intertrial interval was 50ms in the cue-based format, 10ms in the serial-order format, and 300ms for the switching tasks. For cued trials in the cue-based format, cues were displayed with the onset of task stimuli and remained on the screen for 1000ms. Participants could respond any time after stimulus onset (regardless of whether the cues were on the screen). If no response was made after the 1000ms cue display period, the cues would disappear, and only the frame and stimulus would remain until a response was made. For the switching tasks, rule cues were displayed at the center of the screen for 500ms before the onset of task stimuli and remained on the screen until a correct response was made.

Monetary incentives were earned based on block-wise accuracy rates. For the level 1 hierarchical structures in both formats, participants earned $0.03 for every block they completed with at least 95% accuracy. For level 2, they earned $0.03 for every block they completed with at least 85% accuracy. For levels 3-4, they earned $0.03 for every block they completed with at least 75% accuracy. Based on this, participants could earn up to $9.36, in addition to the $10 hourly base rate.

2.1.4. Availability of Data, Codes, and Preregistration

Data, main analysis codes, as well as the preregistration for Experiment 2 are available at the OSF site https://osf.io/gzsk8/.

2.2. Results and Discussion

After excluding subjects with accuracy rates below 70%, we used the data from 23 subjects in our analyses. On the trial level, trials with extreme response times (below 200ms and above the 99.5th percentile for response times across all subjects and conditions) were excluded from analysis.

With the large range of complexity across the four hierarchical levels, neither RTs nor error scores may be sufficient to adequately capture performance. Therefore, we created for our primary analyses a standardized performance cost variable by first z-scoring error rates and RTs across subjects and conditions, and then averaging the two z-scores into a composite performance cost measure (see Liesefeld & Janczyk, 2019). For this performance cost variable, higher numbers represent worse performance. We also present RTs and errors in Appendix B (B1 for RTs and B2 for errors in Experiment 1; B3 for RTs and B4 for errors in Experiment 2), which overall show qualitatively similar patterns as the combined scores.

For the cue-based and serial-order formats separately, we entered our three sets of model-based variables into linear mixed regression analyses, to predict performance within subject and task (see Table 3.3 for condition means). The coding schemes applied to the predictor variables have been introduced in *Paradigm and Predictions* (Section 1.5) and can be found in Tables 3.1 and 3.2 for cue-based and serial-order formats, respectively. As previously mentioned, because the ballistic model can be nested within either the dynamic or global integration models, we tested the ballistic model against the other two individually (i.e., ballistic vs. dynamic, and ballistic vs. global integration). To compare the dynamic and global integration models, we used the Akaike Information Criterion (AIC), with lower numbers indicating better model fit.

2.2.1. Cue-Based Models

Table 3.3. Experiment 1 and 2 mean and SD of accuracy and RT by hierarchical level and task, in both formats.

		Experiment 1 (N=23)		Experiment 2 (N=53)	
Cue-Based Format					
Level	Task	Accuracy	RT (ms)	Accuracy	RT (ms)
1	Spatial Rules	0.99 (0.07)	696.63 (380.80)	0.96 (0.16)	637.72 (347.83)
	Number Judgment	0.98 (0.09)	659.63 (286.90)	-	-
	Odd One Out	1.00 (0.01)	541.13 (157.04)	-	-
2	Spatial Rules	0.97 (0.15)	819.12 (469.99)	0.93 (0.24)	874.93 (592.43)
	Number Judgment	0.95 (0.19)	1110.63 (743.47)	-	-
	Odd One Out	0.96 (0.19)	832.05 (442.09)	-	-
3	Spatial Rules	0.95 (0.20)	1073.37 (911.61)	0.91 (0.27)	1050.42 (848.73)
	Number Judgment	0.95 (0.21)	1386.25 (1262.00)	-	-
	Odd One Out	0.95 (0.21)	1010.75 (812.85)	-	-
4	Spatial Rules	0.94 (0.23)	1157.53 (1057.92)	0.89 (0.30)	1151.60 (1048.70)
	Number Judgment	0.94 (0.22)	1465.79 (1422.82)	-	-
	Odd One Out	0.94 (0.22)	1099.79 (964.21)	-	-
Serial-Order Format					
Level	Task	Accuracy	RT (ms)	Accuracy	RT (ms)
1	Spatial Rules	0.99 (0.04)	549.61 (159.82)	0.96 (0.15)	697.56 (301.08)
	Number Judgment	0.98 (0.09)	664.43 (337.36)	-	-
	Odd One Out	0.99 (0.05)	524.22 (159.20)	-	-
2	Spatial Rules	0.97 (0.13)	838.97 (392.20)	0.96 (0.18)	1077.70 (584.96)
	Number Judgment	0.96 (0.17)	1218.05 (672.41)	-	-
	Odd One Out	0.98 (0.12)	726.90 (320.11)	-	-
3	Spatial Rules	0.96 (0.18)	1007.79 (582.66)	0.94 (0.22)	1282.26 (801.88)
	Number Judgment	0.94 (0.22)	1398.96 (902.30)	-	-
	Odd One Out	0.93 (0.24)	956.54 (607.75)	-	-
4	Spatial Rules	0.93 (0.24)	1308.36 (896.49)	0.90 (0.27)	1531.12 (1120.83)
	Number Judgment	0.92 (0.25)	1585.86 (1122.19)	-	-
	Odd One Out	0.93 (0.25)	1145.30 (813.61)	-	-

Observed and model-predicted performance across all three tasks is shown in Figure 3.3 (top panels), and results of each of the models can be found in Table 3.4. Generally, the results show worse performance as a function of hierarchical level and number of cues. The model tests provide more detailed information about the precise origin of processing constraints. All predictors were highly reliable for the ballistic updating model (Table 3.4). As is clear from the model predictions, this model captured the substantial performance costs that arose whenever a cue was presented that “fit” the level of control relevant for that block (e.g., the cost increase for a level 2 cue in a level 2 structure, or for a level 3 cue in a level 3 structure).

Both the global integration ($X^2(3)$=31.65, p<.001) and the dynamic integration ($X^2(3)$=76.73, p<.001) models provided significantly better fit than the ballistic model. Clearly, the associated X^2 value was much larger for the dynamic integration model, and the AIC values

Table 3.4. Fixed effects from all cue-based and serial-order models in Experiment 1, with performance cost as the outcome variable.

Cue-Based Format

	Ballistic Model (AIC = 440.49)				Global Integration Model (AIC = 414.85)				Dynamic Integration Model (AIC = 369.76)			
Predictors	b	SE	t	p	b	SE	t	p	b	SE	t	p
L2 B	0.47	0.020	23.42	<0.001	0.46	0.026	17.63	<0.001	0.33	0.030	11.03	<0.001
L3 B	0.70	0.025	27.80	<0.001	0.65	0.029	22.21	<0.001	0.50	0.034	14.97	<0.001
L4 B	0.41	0.040	10.19	<0.001	0.37	0.043	8.69	<0.001	0.27	0.047	5.72	<0.001
L2 G	-	-	-	-	0.02	0.033	0.49	0.622	-	-	-	-
L3 G	-	-	-	-	0.09	0.029	3.05	0.002	-	-	-	-
L4 G	-	-	-	-	0.07	0.026	2.50	0.013	-	-	-	-
L2 D	-	-	-	-	-	-	-	-	0.13	0.031	4.07	<0.001
L3 D	-	-	-	-	-	-	-	-	0.22	0.033	6.82	<0.001
L4 D	-	-	-	-	-	-	-	-	0.23	0.041	5.65	<0.001
Filter	0.02	0.01	2.12	0.034	0.05	0.012	4.44	<0.001	0.07	0.012	6.04	<0.001

Serial-Order Format

	Ballistic Model (AIC = -109.23)				Global Integration Model (AIC = -1395.81)				Dynamic Integration Model (AIC = -1256.85)			
Predictors	b	SE	t	p	b	SE	t	p	b	SE	t	p
L2 B	0.37	0.011	33.54	<0.001	0.32	0.010	33.45	<0.001	0.11	0.013	8.78	<0.001
L3 B	0.27	0.019	14.01	<0.001	0.20	0.016	12.00	<0.001	0.001	0.021	0.04	0.971
L4 B	0.22	0.033	6.72	<0.001	0.13	0.028	4.57	<0.001	0.04	0.033	1.13	0.260
L2 G	-	-	-	-	0.12	0.010	11.79	<0.001	-	-	-	-
L3 G	-	-	-	-	0.12	0.010	12.16	<0.001	-	-	-	-
L4 G	-	-	-	-	0.14	0.010	14.35	<0.001	-	-	-	-
L2 D	-	-	-	-	-	-	-	-	0.24	0.008	29.16	<0.001
L3 D	-	-	-	-	-	-	-	-	0.35	0.017	21.06	<0.001
L4 D	-	-	-	-	-	-	-	-	0.27	0.029	9.38	<0.001

Note. The outcome for each of the three models was standardized performance cost. Fixed effects were nested within task and participant. L = level. B = ballistic variable. G = global integration variable. D = dynamic integration variable. N = 23. All predictors are defined in Table 2 for cue-based and Table 3 for serial-order.

indicate a better fit for the latter than the former (369.76 vs. 414.85). Further, all coefficients for the dynamic integration model were robust, whereas the coefficients for the global integration model were small and partly non-significant. Note that the global integration model tries to enforce an across-the-board number-of-levels effect that is not apparent in the data (see Figure 3.3, first and third top panels). In contrast, the dynamic integration model captured the fact that performance costs arose, both at level-specific decision points, and whenever at least one additional structural level was relevant beyond the cue level for the current decision. Additional relevant structural levels beyond that did not add additional costs. This pattern is particularly clear across structure levels, with no presented cues (i.e., cue level 1 in Figure 3.3, first top panel). In structure level 2, there were small but highly robust costs over and above those incurred at the same cue level in a level 1 structure, but there were no additional costs in structure levels 3 or 4. The same general pattern repeats itself for structure level 3 versus 2 at cue level 2, and for structure level 4 versus 3 at cue level 3.

Across all models, the coefficient for the filtering variable was highly significant, indicating that even irrelevant cues had a small, but robust effect on performance. The preregistration for Experiment 2 was submitted before we had conducted analyses including the filtering variable in Experiment 1. Therefore, we also present modeling results without that variable in Appendix B (B5), which produced overall qualitatively similar conclusions as the analyses with the complete model.[2]

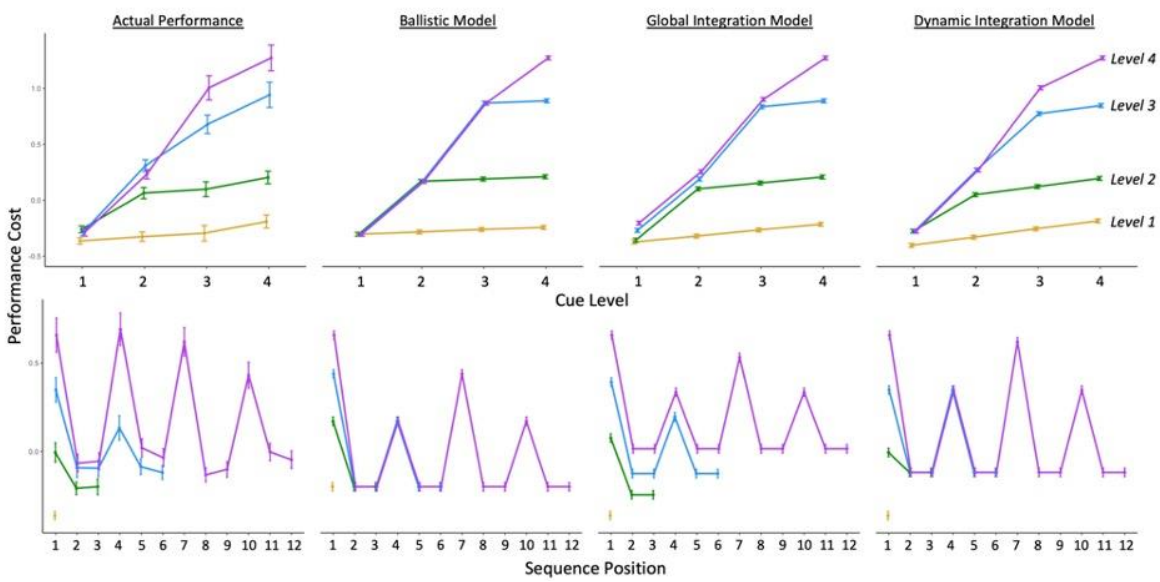


Figure 3.3. Performance results and model fits for Experiment 1 in cue-based (top panels) and serial-order (bottom panels) formats. Error bars indicate 95% within-subject confidence intervals.

---

[2] The reliable filtering effect also raises an interesting substantive question: Does it reflect perceptual/attentional interference that arises because of additional objects on the screen, or is it better understood as a kind of proactive interference caused by the fact that the irrelevant cues were relevant in other encountered conditions? To address this question, we ran a short control experiment in which a new group of participants (*N*=39) performed only the trials from structural level 1 in the cue-based format from Experiment 1, with all four cue levels (which in this case were never relevant, and therefore meaningless). Results (presented in Appendix B6) showed a small, but robust perceptual/attentional filtering cost (i.e., in the control experiment), but a significantly larger cost in the hierarchical control context. We interpret this increased filtering cost as reflecting proactive interference from the instructed and/or in previous blocks experienced, hierarchical control structure.

2.2.2. Serial-Order Models

For the serial-order format, again the ballistic updating predictors were highly reliable, accounting for the fact that there were robust performance costs at chunk transition points, and these costs increased with additional levels (see Figure 3.3 bottom panels, and Table 3.4 for full results from the three models). As in the cue-based format, both the global integration and dynamic integration models led to a robust increase in fit over the ballistic model. However, here the X^2 value was substantially higher for the global integration model ($X^2(3)=1292.58$, $p<.001$) than for the dynamic integration model ($X^2(3)=1153.61$, $p<.001$), and AIC values indicated a better fit for the former than the latter (-1395.81 vs. -1256.85). The global integration model clearly captured the presence of number-of-levels effects across all sequence positions, including within-chunk positions 2 and 3 (which required no chunk transitions). Further, contrary to the ballistic updating model, we found that the same level-specific transition cost increased with relevant structural level. For example, the level 2 chunk transition cost in a level 3 structure (at the start of the second chunk in a two-chunk sequence, i.e., sequence position 4) was higher than the level 2 chunk transition cost in level 2 structure (sequence position 1 in the one-chunk condition), and the cost of switching to the second or fourth chunk within a 12-element sequence (at positions 4 and 10) was even higher than that.

2.2.3. Model Limitations

Overall, the modeling provided a reasonable account of the pattern of empirical results. Nevertheless, there were a few points where predicted and empirical results diverged. In the cue-based format, the most significant point of misfit was for cue levels 3 and 4 at structural levels 3 and 4, where all three models predicted a steeper increase from cue level 3 to 4 for structural level 4 than for structural level 3. In contrast, the empirical results showed a parallel increase across structural levels. One possible explanation for this pattern is that the two cues relevant for levels 3 and 4 were perceptually integrated (i.e., vertically or horizontally aligned stars surrounding one of the two possible context letters, e.g., “A” or “B”). This may have made it more difficult to interpret each cue in isolation, and for that reason made the effect of cue level 4 more similar across structural levels 3 and 4.

Another pattern of model misfit occurred in the serial-order format on structural level 4 (four-chunk super-plans). Specifically, the overall best-fitting, global integration model predicted greatest costs at transitions between the two-chunk plans (positions 1 and 7). Yet, empirically, this predicted pattern of costs did not appear across positions 1 and 7, versus 4 and 10 (i.e., at chunk transitions within the super-chunks). This source of misfit also affected the prediction for chunk transitions on level 3. Here, empirical costs were much smaller for position 4 than 1 (as predicted). However, in trying to account for the opposite pattern on level 4, the models could not recover this difference. A likely reason for this misfit arises from the way in which we implemented level 4, namely as a chunk-level serial-order reversal between the two, two-chunk plans (e.g., chunk 1–chunk 2 — chunk 2–chunk 1). This means that there was always a complete repetition of chunks across the boundary of the two-chunk plans, which may have reduced costs at those boundaries. As shown in Appendix B7, adding a post-hoc chunk repetition predictor to the model eliminates the fit issue on level 3 and leads to a substantial increase in model fit.

Finally, the task-specific results in each format can be found in Appendix B (B8-9). For the cue-based format, these results show another source of misfit. We found that for one of the three task domains, the spatial rules task, there was no clear differentiation between structural level 1 versus levels 2-4 in the no-cue condition (cue level 1), a difference that is particularly critical for ruling out the ballistic model. We believe that the most likely reason for this difference in the overall pattern is that the spatial rules task was presented first in the experimental session, when participants may not have been fully familiarized with the complex overall procedure. In Experiment 2, we were able to replicate the characteristic cost pattern for the spatial rules task.

3. Experiment 2

Results from Experiment 1 revealed consistent evidence in favor of the dynamic integration model across all three tasks when the hierarchical structure was implemented through environmental cues. In contrast, when hierarchical control was based on sequential plans, we obtained robust evidence in favor of the global integration model. Given the complex pattern of results and the exploratory nature of our modeling procedure, we wanted to provide an independent replication before considering further theoretical implications. Also, in the Experiment 1 procedure, participants always completed the cue-based format before the serial-order format. Therefore, in Experiment 2 we repeated the design from Experiment 1 for one of the three tasks (spatial rules) and counter-balanced the order of the two formats, in order to check that the format order was not responsible for our findings. We preregistered our predictions and analyses prior to analysis of Experiment 2.

3.1. Methods

3.1.1. Participants

Data were collected from 84 University of Oregon undergraduate psychology students. Participation was voluntary, and subjects received class credits for completing the study. Because this was an online study conducted during the COVID pandemic, we oversampled participants in order to arrive at a sufficient number of participants after applying our a-priori cut-off criteria (see Section 3.2.). Subjects signed up to participate through the university's

SONA system for human subjects research. After signing up, they were given a link to the experiment on Pavlovia. The study was conducted in a single online session and lasted about 1.5 hours.

3.1.2. Stimuli, Design, and Procedure

Stimuli and design were the same as in Experiment 1. Participants were instructed to complete all the levels in both hierarchical formats, bookended with the switching task, for the spatial rules task only. Because of the online format, instructions were presented in the form of self-paced slides, and participants had to click through the full set of instructions for a given section before continuing onto the task. The instruction slides are presented in Appendix C. We kept the instructions as consistent as possible to those in Experiment 1.

For the cue-based context, participants completed 624 total trials across the four levels: 72 trials on Level 1, 120 trials on Level 2, 168 trials on Level 3, and 264 trials on Level 4. For the serial-order context, participants completed 864 total trials across the four levels: 72 trials on Level 1, 144 trials on Level 2 (6 different three-element chunks, repeated 8 times each), 216 trials on Level 3 (6 different two-chunk sequences, repeated 6 times each), and 432 trials on Level 4 (6 different four-chunk super-plans, repeated 6 times each). The intertrial interval was 500ms in the cue-based context and 50ms in the serial-order context. The switching task included 50 trials at the beginning of the session and another 50 trials at the end. Importantly, in this version of the experiment, format order (cue-based vs. serial-order) was counterbalanced across participants.

3.2. Results and Discussion

The same exclusion criteria as Experiment 1 were used for Experiment 2. However, in order to avoid skewing the upper RT cutoff from trials in which the participant had clearly turned their focus from the experiment, we removed all trials with RTs above 20 seconds before calculating the 99.5th percentile for each level (0.15% of total serial-order trials, and 0.11% of total cue-based trials). We excluded trials with RTs below 200ms or above the upper RT cutoff.

After excluding 8 subjects who did not complete all components of the experiment and 23 subjects with accuracy rates below 70%, we used the data from 53 subjects in our analyses. Though this exclusion rate seems high, it is not surprising given the complexity of the experiment and the fact that motivation and performance tend to vary more for online studies than for in-person studies, which can be more carefully controlled and monitored.

Other than excluding the within-task nesting parameter (because there was only one task pair in Experiment 2), all analyses were the same as in Experiment 1. Means and standard deviations can be found in Table 3.

3.2.1. Cue-Based Models

Observed and model-predicted performance is shown in Figure 3.4 (top panels). Most of the model-based variables were significant predictors of performance in the three models (for complete results of each of the models, see Table 3.5). The same pattern as in Experiment 1 was found, with both dynamic integration ($X^2(3)$=60.61, p<.001) and global integration ($X^2(3)$=37.83, p<.001) models fitting the data significantly better than the ballistic model. Further, as was the case in Experiment 1, the dynamic integration model provided the best fit, with an AIC of -234.02 (vs. AIC=-211.24 for global integration).

As in Experiment 1, the cost increase from cue level 3 to 4 was more parallel across structural levels 3 and 4 than any of the models predicted. Possible reasons for this misfit were discussed in the context of Experiment 1 (Section 2.2.3).

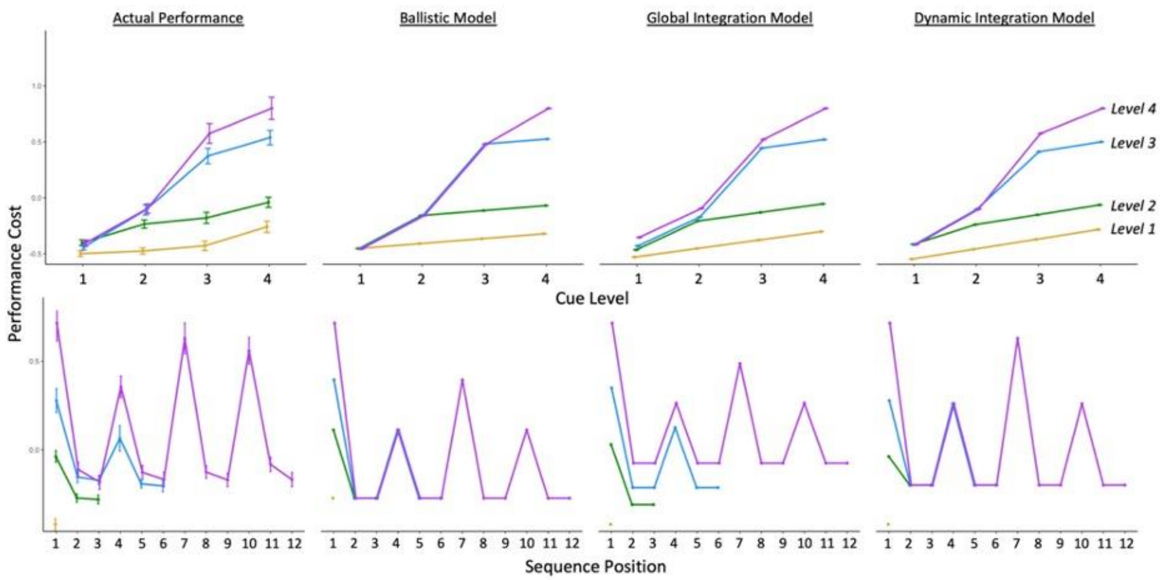


Figure 3.4. Performance results and model fits for Experiment 2 in cue-based (top panels) and serial-order (bottom panels) formats. Error bars indicate 95% within-subject confidence intervals.

3.2.2. Serial-Order Models

Observed and model-predicted performance is shown in Figure 3.4 (bottom panels), and results from the three linear mixed models are shown in Table 3.5. Again, the pattern of results matches that found in Experiment 1, with significantly better model fit for both dynamic integration ($X^2(3)$=966.98, p<.001) and global integration ($X^2(3)$=1056.15, p<.001) models, relative to the ballistic model. The global integration model yielded the best fit, with a lower AIC (-1647.98) than that of the dynamic integration model (AIC=-1558.82). Further, all predictors were statistically robust for the global integration model but not for the dynamic integration model. This finding is again in line with Experiment 1, indicating that for serial order control, the cognitive system requires integration across all levels, at each point of sequential execution. Finally, while level 4 costs had deviated from the predicted pattern in Experiment 1 (which we attributed to the presence of chunk repetitions across super-chunk transitions), in the current experiment, the profile of costs was closer to predictions, at least for the first super-chunk. However, as in Experiment 1, including a variable that accounts for chunk repetitions led to an increase in fit (see S7; note specifically, the size of the level 4 costs for sequence position 4). This suggests that, again, repetition benefits muted the profile of costs that would otherwise have been observed on level 4.

Table 3.5. Fixed effects from all cue-based and serial-order models in Experiment 2, with performance cost as the outcome variable.

	Ballistic Model (AIC = -179.41)				Global Integration Model (AIC = -211.24)				Dynamic Integration Model (AIC = -234.02)			
Cue-Based Format												
Predictors	*b*	SE	*t*	*p*	*b*	SE	*t*	*p*	*b*	SE	*t*	*p*
L2 B	0.30	0.017	17.92	<0.001	0.26	0.021	12.28	<0.001	0.18	0.024	7.35	<0.001
L3 B	0.64	0.021	31.11	<0.001	0.62	0.024	26.17	<0.001	0.51	0.027	18.76	<0.001
L4 B	0.32	0.032	9.83	<0.001	0.28	0.035	8.02	<0.001	0.22	0.038	5.89	<0.001
L2 G	-	-	-	-	0.07	0.027	2.42	0.016	-	-	-	-
L3 G	-	-	-	-	0.03	0.023	1.48	0.140	-	-	-	-
L4 G	-	-	-	-	0.08	0.021	3.62	<0.001	-	-	-	-
L2 D	-	-	-	-	-	-	-	-	0.13	0.025	5.19	<0.001
L3 D	-	-	-	-	-	-	-	-	0.14	0.026	5.20	<0.001
L4 D	-	-	-	-	-	-	-	-	0.16	0.033	4.90	<0.001
Filter	0.04	0.008	5.64	<0.001	0.08	0.010	7.88	<0.001	0.09	0.010	8.95	<0.001
Serial-Order Format												
	Ballistic Model (AIC = -597.83)				Global Integration Model (AIC = -1647.98)				Dynamic Integration Model (AIC = -1558.82)			
Predictors	*b*	SE	*t*	*p*	*b*	SE	*t*	*p*	*b*	SE	*t*	*p*
L2 B	0.39	0.011	34.53	<0.001	0.34	0.010	35.56	<0.001	0.16	0.013	12.78	<0.001
L3 B	0.28	0.019	14.66	<0.001	0.22	0.016	13.73	<0.001	0.02	0.020	0.83	0.409
L4 B	0.32	0.033	9.73	<0.001	0.23	0.027	8.33	<0.001	0.09	0.033	2.59	0.010
L2 G	-	-	-	-	0.11	0.010	11.17	<0.001	-	-	-	-
L3 G	-	-	-	-	0.10	0.010	9.82	<0.001	-	-	-	-
L4 G	-	-	-	-	0.14	0.010	14.25	<0.001	-	-	-	-
L2 D	-	-	-	-	-	-	-	-	0.22	0.008	26.64	<0.001
L3 D	-	-	-	-	-	-	-	-	0.30	0.017	17.95	<0.001
L4 D	-	-	-	-	-	-	-	-	0.35	0.029	12.22	<0.001

Note. The outcome for each of the three models was standardized performance cost. Fixed effects were nested within participant. L = level. B = ballistic variable. G = global integration variable. D = dynamic integration variable. N = 53. All predictors are defined in Table 2 for cue-based and Table 3 for serial-order.

4. General Discussion

Our goal was to determine the source of processing constraints that result from increasing the number of levels within a hierarchical control structure. For that purpose, we manipulated the number of relevant levels (between one and four) within separate blocks of trials. We also established control structures either through environmental signals (cues) or in form of memorized, sequential plans. Hierarchical control structures are decision trees, in which a higher-level (and therefore earlier) decision establishes a contextual setting that governs lower-level decisions. As each additional level adds a decision point, it is not surprising that level-dependent decision costs arose whenever a potential change on a particular level needed to be considered. This is consistent with essentially any conceivable model of hierarchical control. Indeed, for the cue-based format, costs increased with every additional cue presented on a given trials, as long as that cue was relevant in that particular structure level. For the serial-order format, costs increased whenever a transition occurred between a subunit of the sequence, and the costs were larger with more nested units to be considered (i.e., elements, three-element chunks, two-chunk sequential plans, and four-chunk super-plans).

The theoretically more important question concerns what happens *after* the decision about a higher-level setting has been made. If the cognitive system is optimized for processing efficiency, it should retain the most recently reached point in the decision tree for subsequent lower-level decisions. Any such lower-level decision would then start from that point, instead of requiring routing through the previous decisions again. Based on our results, we can clearly reject such a pure ballistic model of hierarchical control.

In the cue-based format, this was reflected in the finding that lower-level decisions were negatively affected whenever higher structure levels were relevant, even on trials in which no higher-level cues were shown. For example, in the zero-cue condition (cue level 1), there was a small but highly robust difference between structure level 1 performance and performance in the remaining three structure levels. Equally, at cue level 2, performance was better in the level 2 structure than in either of the remaining structure levels (3 or 4). For the serial-order format, the ballistic model predicts that within-chunk positions 2 and 3 should not require any updating or consideration of higher-level settings. Nevertheless, these positions showed a very clear ordinal scaling of performance cost as a function of number of levels.

While results for cue-based and serial-order hierarchical control are both inconsistent with the pure ballistic model, the similarity in cost profiles between formats ends here. The just-mentioned finding that costs increased with the number of structural levels across all positions in the serial-order format conforms with the assumption of the global integration model that all potentially relevant aspects (i.e., levels) need to be integrated for each individual decision during sequence execution. In contrast, for the cue-based format, data were consistent with what we have referred to as the dynamic integration model. The critical data pattern here is that in the absence of higher-level cues, lower-level decisions are negatively affected to the same degree by all relevant higher levels– additional costs are only incurred when there is a next-highest structural level to be considered; levels beyond that do not influence performance. In the remainder of this discussion, we will address the theoretical implications of these results in turn.

4.1. The Limitations on Ballistic Control and the "Next-Level-Up" Effect

One possible reason for peoples' limited ability to benefit from previous higher-level decisions in a ballistic manner is that having to maintain the higher-level setting in working memory absorbs resources that would otherwise be available for lower-level decisions. Evidence from dual-task studies suggests that working memory maintenance demands may impose unspecific costs on simultaneous RT tasks, but these results also provide little evidence for interference between working memory load and parallel executed decisions (e.g., Moss, Kikumoto, & Mayr, 2020). The present data pattern is also not consistent with a static, maintenance-cost view. If maintenance were the critical factor, we would expect that costs generally scale with the increase in maintenance demands from increasing the number of structural levels. Yet for the cue-based format, that clearly was not the case, as additional costs were incurred only for the next structural level up from the highest-level cue presented on a given trial, but not for additional structural levels beyond that.

What then is behind this next-level-up cost? We believe that the most likely explanation involves stimulus-driven retrieval in order to dynamically integrate the stimulus with the current task rule. Specifically, we suggest that a stimulus which requires a decision triggers an "upward" retrieval process to activate the *necessary and sufficient* context relevant for that decision. For example, when in a 4-level structural context, if a level 2 cue appears (frame color), the relevant context is the level 3 rule, which is required to interpret that level 2 cue (e.g., "red=vertical/green=horizontal"). The exact constellation of higher-level cues that have put this rule into place is not relevant for the lower-level decision and therefore does not need to be retrieved. Consequently, performance is not affected by whether the level 2 cue appears within a level 3 or level 4 structural context. This "upward-retrieval" account necessarily assumes that during encoding of cues, people do not keep a perceptual record of the input information (i.e., the cues). If that were all they retained, interpretation of the next-level cue would also require the perceptual record from one further level up (and so forth). Instead, we suggest that the cue information is translated during encoding into the functionally relevant instructions (e.g., "red=vertical/green=horizontal"), and it is these instructions that are retrieved. This account is consistent with recent evidence from Ehrlich & Murray (2021), who used stimuli in a working memory task that could be encoded either veridically or in terms of functionally relevant rules for post-delay responding. Results indicated that participants immediately encoded information into response rules rather than retaining stimulus information.

One important assumption behind our explanation of the next-level-up effect in the cue-based format is that it is prompted by experienced, stimulus/cue-driven ambiguity or conflict, rather than by the mere presence of a higher-level structure. Incidentally, our task design allowed for an additional, and originally unintended test of this explanation of costs. In order to make structural level 4 manageable for participants, we included this level in a "degenerate" manner, namely by having the level 4 cue indicate whether level 3 cues were necessary or could be ignored in favor of applying the default rule set associated with one of the level 3 cues (e.g., cue letter "A;" see Figure 3.1a). Thus, when the A context matched the default rule set, participants only experienced ambiguity about what to do when the letter B appeared as a cue. The upward-checking assumption would predict that only in this case is it important to reaffirm whether the current level 4 context requires use of the level 3 cues or not. To test this prediction, we split the structural level 4 data into the default-matching level 3 cue ("A") and the non-default cue ("B") groups. Figure 3.5 shows the results for both experiments, with structural level 4 split into default and non-default trials, along with the results from the remaining conditions. As apparent, the prediction that costs arise mainly for the non-default letter was very clearly confirmed for Experiment 1, and at least partially confirmed for Experiment 2, namely for cue level 4. Overall, this pattern of results provides an additional

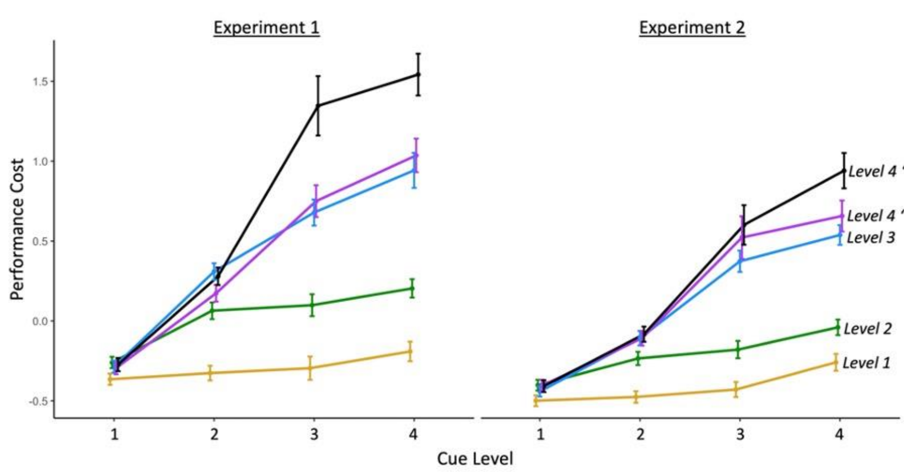


Figure 3.5. Cued performance, with structure level 4 split out into two groups. Level 4 "A" represents the level 3 cue that signaled the same rule set as the "default context," and Level 4 "B" represents the level 3 cue that signaled the other possible rule set. Error bars indicate 95% within-subject confidence intervals.

confirmation that much of the costs in the cue-based format arise dynamically, in order to resolve stimulus/cue-driven ambiguity.

4.2. Why Does Serial-Order Control Require Integration across All Levels?

From the qualitatively different patterns of hierarchical control costs across formats, one might conclude that fundamentally different control architectures are involved in the cue-based and serial-order formats. For the latter, we found number-of-levels effects for every single trial, even when no context transition was necessary (i.e., within-chunk positions 2-3). So why would overall contextual load matter here but not in the cue-based format? Does a serial-order control structure impose complexity-dependent maintenance demands that are not required for an environmental signals-based control structure? While we currently cannot rule out such a global maintenance-cost account, we suggest instead that this cost pattern reflects the specific needs of serial-order control, which requires keeping track of one's current position within task space. A sequential position is one point within a space with as many dimensions as hierarchical levels. For example, the current location within a level 4 sequence is jointly defined by the chunk super-plan, the chunk plan, the chunk, and the within-chunk position. Given that position within the sequence changes on every trial, this step of cross-level integration also needs to occur on every trial.

A substantial literature addresses how sequential representations are reflected in the within-sequence pattern of RTs or accuracy rates (Mayr, 2009; Restle, 1970; Rosenbaum et al., 1983; Schneider & Logan, 2006; Verwey, Shea, & Wright, 2015). One of the most pervasive findings in literature is that large costs arise at transitions between sequence subunits (i.e., between chunks), suggesting that our cognitive system deals with the problem of serial-order mainly at these transition points. To our knowledge, the current study provides the first systematic assessment of how the overall control structure complexity of a sequence affects performance. The finding of level effects across *all* sequence positions is novel and important, as it indicates that the serial-order representation needs to be referenced continuously, at every position of a sequence. These results are also consistent with recent work using EEG decoding of three-level serial-order control structures and (Kikumoto & Mayr, 2018), which demonstrated the continuous identity representation of individual sequential elements, within-chunk position codes, and individual chunks. These are exactly the aspects that need to be represented and integrated on every trial in order to keep track of one's position in the overall sequence.

Though our results demonstrate a difference in cost patterns between the cue-based and serial-order formats, this does not necessarily imply that the two formats use fundamentally different processing architectures. A more parsimonious assumption would be that our cognitive system *always* applies dynamic stimulus-driven integration of necessary and sufficient context information through "upward retrieval." The difference comes from the fact that what is necessary and sufficient varies across format. The serial-order format requires the integration of all relevant levels in order to keep track of current sequence position. In contrast, in the cue-based format there is no danger in "getting lost in task space," because higher-level transitions are reliably indicated through external cues. Therefore, here checking the next-level-up context is sufficient for interpreting the current stimulus.

4.3. Maintenance and Updating in Working Memory

Our results are broadly consistent with models of working memory that assume a distinction between maintenance and updating modes (e.g., Badre, 2012; Kessler & Meiran, 2008; Kessler & Oberauer, 2014; Mayr, Kuhns, & Hubbard, 2014; O'Reilly, 2006). In these models, maintenance is typically regarded to be the default mode that is geared towards shielding current information from new input. Therefore, it is plausible to assume that in the cue-based format with no cues presented, participants remain in the efficient maintenance mode. In contrast, presentation of potentially relevant cues triggers an updating process, for which costs track with the amount of information (i.e., number of levels) to be considered on a given trial. In this regard, the number-of-levels effect might be considered an instance of the set-size effect on updating costs that is typically found when the amount of information that needs updating is experimentally manipulated (Kessler & Oberauer, 2014).

So how then do our results from the serial-order and cue-based formats fit with this maintenance/updating distinction? Interestingly, research by Kessler and Meiran (2008) indicates that while people can partially update working memory content, the updating cost for a given amount of updated information is larger when there is additional to-be-maintained information in working memory. The authors attribute these extra costs to a "global" updating process that "is responsible for stabilizing the representations in working memory after the relevant modification takes place" (p. 1346). This is not unlike our interpretation of the pervasive number-of-levels effects in the serial-order format: When moving through a sequence, each position change requires reintegration of position information with all remaining relevant aspects in order to keep track of the current location within the sequence. Similarly, for the next-level-up effect in the cue-based format, the current stimulus or cue needs to be integrated with the relevant higher-level working memory content in order to be accurately interpreted.

It is, however, important to acknowledge that there are critical differences between the standard working memory updating paradigm and hierarchical control situations. In the former, participants need to handle unrelated pieces of information. In the latter, control-relevant information is structured and interrelated: Information on different hierarchical levels is essentially connected through if-then rules. It is not clear how these types of relationships affect the updating/maintenance dynamics or whether they warrant an expansion of standard working memory models. While the standard model assumes that all information is alike and hierarchical structures emerge from the information represented (Crittenden & Duncan, 2014), neurologically informed conceptualizations assume that working memory is segregated into hierarchically organized, functionally separate cortical/subcortical loops (Badre & Nee, 2018). A detailed account of where processing costs emerge during hierarchical control is critical for continuing to develop our understanding of these working memory processes.

4.4. Conclusion and Broader Implications

In principle, hierarchical control should allow a complex problem to be parsed into subproblems that can be solved independently without needing to reconsider the earlier, higher-level decisions upon which the lower-level decisions are contingent. As our results indicate, the cognitive system does not comply with this ideal. Rather, it appears that each lower-level decision requires a context check to determine the currently valid rules. While the pure ballistic model can be refuted, it is also important to acknowledge that the cost of checking an established context remains lower than the cost incurred when that original contextual setting was put into place (i.e., costs at cued updating points or transitions between sequential subunits are larger than at points where no obvious updating/transition is necessary). Thus, there are considerable savings from previous higher-level decisions, likely because the last-established setting remains highly accessible for retrieval. Further, our results show that the context check during lower-level decisions comes in different degrees, depending on how hierarchical control is established. For the cue-driven format, it is only necessary to retrieve the “next-level-up” rule from the current decision. In contrast, for the serial-order format, all relevant levels need to be integrated on every trial in order to establish location within the current sequential context.

The fact that lower-level decisions require checking across levels implies that the supposedly independent representational spaces that should arise from hierarchical control structures are not truly insulated from each other. On the one hand, produces interference and therefore a loss of efficiency. On the other hand, this may allow for information exchange, generalization, or the induction of abstract patterns across representational units. For example, contingencies that affect lower-level decisions in one subspace can partially generalize to alternate subspaces (Bryck & Mayr, 2005). In the case of serial-order control, people can extract abstract relationships between successive chunks to efficiently code an overall sequence (Restle, 1970). Future research should address this question of the degree to which the less efficient, non-ballistic features of control are the very aspects that allow useful information to be integrated across different regions of a task space. The way hierarchical control operates may be a compromise between the opposing goals of efficient performance on the one hand and efficient representations through learning and structure building on the other.

Acknowledgements

This research was supported by NSF grant 1734264 (PI: Mayr).

CHAPTER IV

INDIVIDUAL DIFFERENCES IMPLICATIONS OF THE HIERARCHICAL CONTROL ARCHITECTURE

1. Introduction

With the cognitive revolution came the discovery of hierarchical structure as an essential feature of complex action and thinking. Hierarchical structures dissect large problems into small, independently manageable subproblems, and provide the basis for generalization and generativity by allowing recombination of existing processes to achieve new goals or cognitive products. Hierarchical structures are key to executing complex sequences such as in music, dance, or language, by organizing individual elements recursively into short subsequences (chunks). However, they can also be implemented through environmentally prompted contexts that modulate how specific signals in the environment need to be interpreted.

The hierarchical organization of task spaces also implies that differences in cognitive functioning between people may result not (or not only) from the specific types of processes or content domains of a particular task. Rather, the characteristics of the of control structure itself may be what drives differences between people in task performance. Concretely, the key question here is whether, and how exactly, higher levels of control produce unique individual differences variance. To address this question, we examined individuals' performance across three different task domains in which we applied hierarchical control structures that varied from one to four different levels (see Figure 4.1). We implemented these structures in a serial-order format, that is in terms of memorized sequences, and in a cue-driven format, with control contexts prompted through external signals. In addition, we assessed each participant's fluid

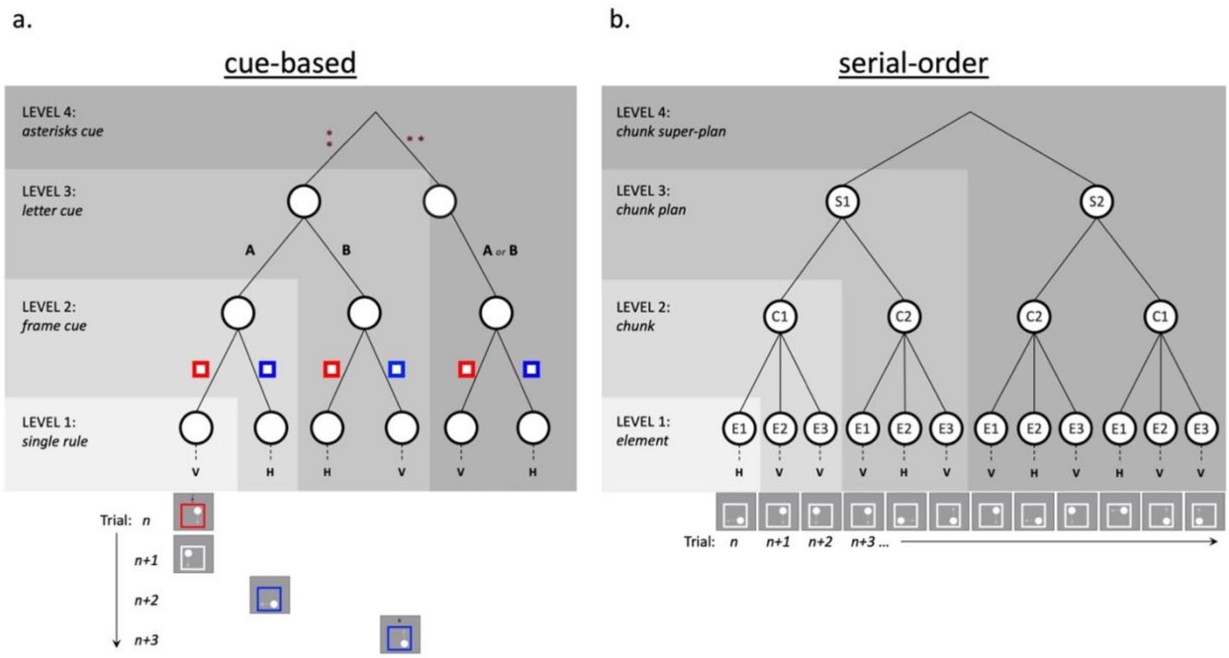


Figure 4.1. Diagram of the hierarchical rule structure in (a) cue-based format and (b) serial-order format. Example trials in the highest task level (level 4) are shown below each diagram. The arrow within the stimulus frame indicates the correct response and was not presented in the actual trial. From Moss & Mayr (2022).

intelligence, working memory capacity, and long-term memory ability in order to examine how level-specific variance may relate to existing, theoretically important constructs. To test key hypotheses about the nature of individual differences variance within this set of task demands, we used structural equation modeling. We begin here by situating our specific hypotheses and questions within the existing literature on hierarchical control.

1.1. Costs of Hierarchical Control

Considerable research documents how hierarchical structures are expressed in terms of performance costs. More complex hierarchical structures produce longer RTs and greater error rates. Evidence from research with the task-switching paradigm suggests that when more than one task can occur within the same context (presumably requiring at least a two-level control structure), RTs and error rates increase, in what is known as global or mixing costs (Mayr, 2001;

Rubin & Meiran, 2005). Moreover, additional costs are observed whenever higher-level representations need to be updated or changed. For example, there is consistent evidence that when executing memorized action sequences, chunk transitions produce large increases of RTs and error rates, as do trial-to-trial transitions between tasks in the task-switching paradigm, known as local switch costs (Kray & Lindenberger, 2000). Yet, performance costs associated with higher levels of hierarchical control do not necessarily imply level-specific processing constraints. In principle, it is possible that basic subcomponents are used recursively to build up complex structures. In this case, there is no need to assume that worse performance with higher-level structures indicates the use of distinct sets of cognitive resources. Such a single-factor hypothesis presents the baseline model against which any more complex model of individual differences within hierarchical control structures needs to be tested.

1.2. Evidence in Favor of Unique Higher-Level Variance

Based on available empirical evidence, there is reason for skepticism about such a single-factor model. For example, it has been demonstrated that higher levels of control are more dependent on working memory capacity and fluid intelligence than lower-level structures (e.g., Bo & Seidler, 2009; Kikumoto & Mayr, 2018). Further, these cognitive abilities have been shown to strongly correlate with each other (Unsworth et al., 2014). Work with the task-switching paradigm has indicated that unique individual differences variance is established when two tasks are performed within the same block (global switch costs), and additional variance emerges when tasks need to be changed (local switch costs). Maybe the most direct exploration of this issue has been carried out by Carpenter et al. (1990), who found that error patterns in performing the Tower of Hanoi task pointed to specific problems in coordinating a complex hierarchical control structure and were relatively strongly related to performance in the Raven

Progressive Matrices task. As these authors argue, the demands in terms of operating within hierarchical control structures are responsible for shared variance between these two—otherwise very different—types of tasks. A similar, general argument had been made earlier by Marshalek et al. (1983) based on the finding that independent of content, complex tasks share more individual differences variance than simple tasks, again supposedly because of the shared demand on hierarchical coordination.

These general observations however, leave a number of possibilities and open questions about how exactly unique, level-specific variance emerges within hierarchical control structures. For example, the existing evidence is largely consistent with just two different sources of variance—one when no hierarchical control is necessary, and the other whenever any kind of higher-level control is required, no matter the number of levels. Importantly however, neuroimaging research documents a dorsal-frontal continuum from lower to higher levels of control, apparently distinguishing at least three levels of control (Badre & D'Esposito, 2009; Koechlin et al., 2003; but see also Badre & Nee, 2018; Duncan, 2010; Riddle et al., 2020). Such a neuroanatomical dissociation between multiple levels does not require the assumption of unique sources of individual differences at each level of control, but raises the possibility that such a differentiation could also exist.

1.3. Temporal Dynamics: Updating versus "Mere-Structure" Effects and Serial-Order versus Cue-Based Control

Another important question regards the temporal dynamics of hierarchical processing constraints that are responsible for individual differences. Such constraints could be very local in nature and arise only when higher-level representations need to be updated. However, it is also possible that merely operating within a complex hierarchical structure taps critical resources, such as a limited working memory space, even in the absence of an obvious, local updating operation. Moss and Mayr (2022) recently showed that for the question of whether control costs are local or global in nature, it may matter whether hierarchical control is established through environmental cues or through memorized action sequences. In the former, behavioral costs arose specifically at points at which the currently relevant, highest level of control needs to be updated and/or integrated with the next lower level of control. This was demonstrated clearly on trials when no cues were present and the last-established setting could simply be reused. Here, no performance costs arose. In contrast, when operating within a memorized, serial-order control structure, complexity (i.e., the number of levels of that structure) appeared to matter for every single trial, regardless of whether an obvious transition between higher-level representations was required. Therefore, one might expect that level-specific individual differences variance would also become apparent on each trial of a hierarchically organized sequence, but only on trials for which higher-level representations need to be updated when operating within a cue-based hierarchical structure. Within the structural equation modeling framework this requires testing for separate sources of variance from trials on which updating might occur, such as between higher-level transitions in serial-order control situations, or when relevant cues are presented for cue-based situations.

1.4. The Architecture of Hierarchical Control

If unique individual differences variance is indeed generated at higher levels, then one particularly interesting question concerns the "degree of modularity" of these sources of variance. At one end of the spectrum are traditional models of hierarchical control that have implicitly or explicitly assumed an additive and ballistic architecture. Here, higher levels set parameters for the lower levels, which then implement these parameters autonomously. Such an architecture can be translated into a latent variable structure in which one latent factor representing the lowest level explains some variance for all observed variables (common variance), no matter how many levels of control were required. The next higher level latent factor then explains residual, or unique, variance for all measures that were performed within at least a two-level structure, and so forth. For a purely additive model, one would test as an additional constraint that the loadings of each lower-level latent factor remain constant across measures reflecting the same basic tasks, irrespective of the number of additional levels relevant for that task. This reflects the idea that the total variance in each measure is simply the sum of independent, level-specific sources of variance. Here, the idealized correlation matrix generated by this model essentially consists of "level-specific bands" of increasing intercorrelations.

We tested this purely additive architecture against an alternative way in which unique higher-level variance could arise, with relaxed additivity assumptions. We refer to this model as the "graded hierarchical control model." It reflects the idea that there is a common resource, such as a global workspace or working memory that handles hierarchical control and that is more relevant as more levels are added (Carpenter et al., 1990; Marshalek et al., 1983). This model can be captured by a variant of the strict "additive" model, in which there are only two factors: a lower-level general factor and one higher-level factor whose loadings increase as a function of task levels. A critical prediction of this model is that correlations will increase for higher levels as a function of distance from the diagonal.

1.5. Structural Effects on Decision Efficiency versus Threshold

Prominent models of how our cognitive system makes action decisions distinguish two important parameters: The efficiency with which decision-relevant information accumulates over time, called drift rate, and how much information is required to make a decision, called decision threshold (Ratcliff & McKoon, 2008). Plausibly, requiring additional levels of control could either reduce information processing efficiency or increase cautiousness. Therefore, failing to distinguish these two aspects may make it difficult to clearly characterize the underlying latent structure, or may lead to incorrect inferences about the nature of processing constraints. Indeed, Lerche and colleagues (2020) recently showed that separating out drift rate and threshold separation is critical in order to detect the psychometric structure underlying simple and complex versions of the same tasks. In the context of hierarchical control, the examination of processing efficiency and decision threshold as separate processes is particularly important given that existing evidence with the task-switching paradigm already suggests that people increase their decision boundaries when operating with multiple tasks (i.e., level 2) compared to a single-task condition (i.e., level 1).

2. Methods

2.1. Participants

Data were collected from a total of 209 subjects for session 1. Of those subjects, 201 also completed session 2. Because of various problems that arose during session 2, the data from eight subjects were excluded, leaving a total of 193 subjects for analysis. Participation was voluntary and was monetarily compensated through both an hourly base rate and additional accuracy-based incentives that could be earned throughout the experiment.

2.2. Session 1

2.2.1. Stimuli and Design

In session 1, two hierarchical control formats, cue-based and serial-order, were used, each with four different possible levels, and each with the same set of three "primary tasks." We will first describe the primary tasks and then the implementation of the control structures.

2.2.1.1. Tasks. Across both formats, participants worked with a spatial rules task, an odd-one-out task, and a number judgment task. For the spatial rules task (adapted from Mayr, 2002), in every trial, a white circle (60-pixel diameter) appeared randomly in one quadrant of a frame (60 pixels off-center), and participants were instructed to indicate which quadrant the circle would end up in if they applied one of two possible spatial rules: horizontal or vertical. Responses were made with the right-hand index finger, using the 4, 5, 1, and 2 keys on the keyboard number pad. These keys correspond to each quadrant of the frame (top left, top right, bottom left, and bottom right, respectively).

For the odd-one-out task, a rectangle (40x75 pixels) appeared in each quadrant of a frame (60 pixels off-center), with one rectangle different in color (blue or green, versus black for all other rectangles), and one rectangle different in pattern (vertical stripes, diagonal zig-zags, or checkers, versus solid for all other rectangles). Participants were instructed to use the same response mapping as in the spatial rules task to indicate which rectangle was the odd-one-out, based on either the color or the pattern rule.

For the number judgment task, participants were shown numbers randomly chosen on each trial from 1, 2, 3, 4, 6, 7, 8, 9 (Arial font, size 88). Numbers were presented individually, centered inside the frame. Participants were instructed to judge whether the number was lower or higher than 5 (L/H), or whether it was odd or even (O/E). Responses were made with the right index finger using the left and right arrow keys, with the left arrow key representing the judgment to the left of the slash (L or O), and the right arrow key representing the judgment to the right of the slash (H or E).

2.2.1.2. Cue-Based Control Structure. In the cue-based format, participants were presented with intermittent in-trial visual cues to indicate which task rule to use. At the lowest level (structure level 1), a single rule was prompted at the beginning of the block, and participants were instructed to apply this rule in every trial. For structure level 2, the frame color indicated which rule to use. The specific mapping between frame color and rule for each task pair was instructed at the beginning of the block. For structure level 3, the conjunction of frame color and above-frame letter cue indicated which rule to use. For structure level 4, the orientation of stars around the letter cue indicated whether to use the same rule combination as indicated in structure level 3, including the conjunction of frame color and letter cue, or to ignore the letter cue and instead, determine which rule to use based on an instructed "default" rule structure (see Figure 1a). We chose this arrangement on structural level 4 after pilot work revealed that a complete reversal of level 3 rules through level 4 rules was too difficult for participants.

Cue presentation was determined the same way across all structure levels, irrespective of which structural levels were relevant on a given block. When higher-level cues were displayed, all lower-level cues were displayed as well. The trial-wise probability of displaying a level 2 cue was .333. In trials with level 2 cues, the probability of also displaying a level 3 cue was .5 (.167, overall). In trials with level 3 cues, the probability of also displaying a level 4 cue was .5 (.083 overall). For trials without cues (or with lower cue level than hierarchical structure level), participants were told to use the rule indicated by the most recently displayed cue(s). After an incorrect response, a diagram of the current block's cue meanings would appear in the top right corner of the screen (similar to the instruction diagram), and participants would need to make a correct response before continuing to the next trial.

2.2.1.3. Serial-Order Control Structure. In the serial-order format, no trial-by-trial cues were provided. Instead, the relevant rule on a given trial was specified through sequences of varying hierarchical complexity. Sequences were explicitly instructed at the beginning of each block and participants “cycled through” repeatedly until the end of the block (as in Mayr, 2009). For structure level 1, participants simply repeated the same rule across trials. Note that except for the omission of visual cues, this condition is identical to structure level 1 in the cue-based format. For level 2, three-trial sequences of rules (“chunks”) were instructed, which were repeated until the end of the block (see Figure 1b). These chunks used one of the following possible formats: A-B-B, A-A-B, A-B-A (and the inverse of each). For level 3, two chunks were grouped into a six-element plan, while avoiding chunk repetitions (e.g., A-B-B–B-A-B). Level 4 chunk super-plans used the same basic, two-chunk plans as on level 3, but added a chunk-level reversal of that plan to create a 12-element sequence (e.g., A-B-B–B-A-B—B-A-B–A-B-B). The instruction screen for level 4 contained six rules (like on level 3), but also included a down arrow on the left side and an up arrow on the right side, indicating that participants were to execute a fourth-level sequence. After an incorrect response, the sequence of rules would be displayed above the frame, with the incorrect response colored red. For structure level 4, one of the arrows would also be red after an error, to indicate location within the 12-element sequence. Participants needed to make a correct response in order to continue on to the next trial.

2.2.1.4. Cued Switching Task. Finally, we also included a standard cued switching task, for which participants were instructed to perform the same three tasks as in the cue-based and serial-order structures, but instead of receiving block-wise rule sequences or in-trial symbolic cues, one of the two rules was presented at the center of the screen for the whole trial, starting 100ms prior to stimulus onset. Rules were randomly chosen and followed no pattern or hierarchical structure. Because these results are not relevant to our main question, we are not using these data in the current paper.

2.2.2. Procedure

This was a 2.5-hour session. All components were completed on the computer (24-inch display), with participants completing tasks first in the cue-based format, and then in the serial-order format. The session was bookended with the cued switching task, such that participants performed the first half of the switching task, then the two contexts, and then the second half of the switching task. A trained experimenter used images and examples to instruct participants before each section of the experiment. Within each of the four components, participants completed the spatial rules task, followed by the odd one out task, and then the number judgement task. In the cue-based and serial-order formats, participants completed all structure levels in a "mountain structure" (level order: 1-2-3-4-3-2-1) for each task.

Within the two formats, the number of trials varied across hierarchical level, so that participants completed more trials for each increase in complexity. For the cue-based context, participants completed 1440 total trials across the four levels: 144 trials (48 per task) on level 1, 288 trials (96 per task) on level 2, 432 trials (144 per task) on level 3, and 576 trials (192 per task) on level 4. For the serial-order context, participants completed 1296 total trials across the four levels: 108 trials (36 per task) on level 1, 216 trials on level 2 (18 different 3-element chunks (6 per task), repeated 4 times each), 324 trials on level 3 (18 different two-chunk sequences (6 per task), repeated 3 times each), and 648 trials on level 4 (18 different four-chunk super-plans (6 per task), repeated 3 times each). The switching task included 108 trials (36 per task) at the beginning of the session and another 108 trials at the end. The intertrial interval was 50ms in the cue-based format, 10ms in the serial-order format, and 300ms for the switching tasks. For cued trials in the cue-based format, cues were displayed with the onset of task stimuli and remained on the screen for 1000ms. Participants could respond any time after stimulus onset (regardless of whether the cues were on the screen). If no response was made after the 1000ms cue display period, the cues would disappear, and only the frame and stimulus would remain until a response was made. For the switching tasks, rule cues were displayed at the center of the screen for 500ms before the onset of task stimuli and remained on the screen until a correct response was made.

Monetary incentives were earned based on block-wise accuracy rates. For the level 1 hierarchical structures in both formats, participants earned $0.03 for every block they completed with at least 95% accuracy. For level 2, they earned $0.03 for every block they completed with at least 85% accuracy. For levels 3-4, they earned $0.03 for every block they completed with at least 75% accuracy. Based on this, participants could earn up to $9.36, in addition to the $10 hourly base rate.

2.3. Session 2

2.3.1. Stimuli and Design

Participants completed nine tasks in the second session, in order to get performance scores from three different tasks for each of the three related latent factors: fluid intelligence, working memory capacity, and long-term memory. The tasks were modeled after those used by Brewer and Unsworth (2012), and Unsworth et al. (2014).

2.3.1.1. Fluid Intelligence Measures. For fluid intelligence (gF), participants completed the Raven Advanced Progressive Matrices task, the Number Series task, and the Letter Sets task.

The Ravens is a measure of abstract reasoning in which different 3x3 matrices of geometric patterns are displayed, with the bottom right of the pattern missing, for each trial. Participants must select the correct image to complete the matrix, from a set of eight options. The trials are presented in order of increasing difficulty, and participants are instructed to try each problem in turn, as they become more difficult to solve. For this task, our participants had 10 minutes to complete the 18 odd-numbered problems (from the original total of 36 items). Scores were calculated as the total number of correct responses. Because of technical issues with the stimulus display, unrelated to any other parts of session 2, we did not get Ravens scores for the first 49 participants. Therefore, we applied multiple imputation, using the scores from all other session 2 tasks, to estimate Ravens scores for those participants with missing data.

For the number series task, participants needed to determine what unstated rule each series of numbers followed, in order to determine the next element in the series. For each trial, the number series was presented visually on the left side of the screen, and five answer choices were provided on the right side. Each answer choice corresponded to a number on the keyboard number line. Participants had 4.5 minutes to complete the 15 problems, and their scores were calculated as the total number of correct responses.

The letter sets task required participants to identify letter patterns. For each trial, five sets of letters containing four letters each were presented on the screen. Four of the sets of letters followed the same pattern rule, and participants were instructed to determine which letter set did not follow that pattern. Each answer choice corresponded to a number on the keyboard number line. Participants were given 5 minutes to complete 20 items, and their scores were calculated as the total number of correct responses.

2.3.1.2. Working Memory Capacity Measures. For working memory capacity (WMC), participants completed a color change detection task, an orientation change detection task, and a space task.

For the color change detection task, four or six colored squares were presented at random non-overlapping positions on the screen. All squares were different colors, and they could be any of the following colors: red, green, blue, yellow, orange, pink, purple, white, or black. The squares were presented on the screen for 200ms, followed by a blank-screen 600ms retention interval. After the retention interval, one of the squares would reappear in its original position, in either its original color, or a different color. Participants were instructed to respond to whether the square was the same color as before using the left and right arrow keys, with the left arrow indicating a change in stimulus color and the right arrow indicating no change in stimulus color. The probe square remained on the screen until a response was made. Square color changed on 50% of trials, and participants completed 11 blocks of 18 trials.

For the orientation change detection task, three or five black circles were presented at random non-overlapping positions on the screen. Each circle was rotated randomly and contained a drawn radius in order to indicate circle orientation. The circles were presented on the screen for 250ms, followed by a blank-screen 650ms retention interval. After the retention interval, one of the circles would reappear in its original position, at either the same orientation as it was originally, or rotated at least 90 degrees. Participants were instructed to respond to whether the circle was at the same orientation as before using the left and right arrow keys, with the left arrow indicating a change in stimulus orientation and the right arrow indicating no change in stimulus orientation. The probe circle remained on the screen until a response was made. Circle orientation changed on 50% of trials, and participants completed 11 blocks of 18 trials.

For the space task, six letter stimuli (A, B, C, D, E, and F) were simultaneously presented on an imaginary circle on the computer screen for 200ms. Participants were instructed to remember as many locations as possible over a 600ms retention interval. After the retention interval, one of the letter stimuli would appear in the center of the screen as a probe, and participants reported the original location of the probe letter by pressing a corresponding key on the keyboard number pad. The response keys mapped onto the on-screen positions, starting at the top left position and moving clockwise: 8, 9, +, 3, 2, 4. The probe stayed on the screen until a response was made. Participants completed 11 blocks of 18 trials.

For each of the WMC tasks, performance for each set size condition was converted to a standard capacity estimate (K) by Cowan's formula: $K = N*(H-FA)$, where N is set size, H is hit rate, and FA is false alarm rate (2001). Scores were calculated by averaging K across set size for the color and orientation tasks separately. For the space task, there was only one set size (six), and so K was calculated for that set size alone.

2.3.1.3. Long-Term Memory Measures. For long-term memory (LTM), participants completed a picture source-recognition task, a paired-associates cued recall task, and a delayed free recall task.

For the picture source-recognition task, participants were presented with a total of 40 pictures. Pictures were presented randomly in one of four quadrants of a white frame, one at a time for one second each. Participants were instructed to pay attention to both the picture and the quadrant in which it appeared. At test, participants were presented again with each of the 40 pictures in random order, in the center of the screen, and for each picture, they had five seconds to indicate which quadrant it had originally appeared in. Responses were made using the 4, 5, 1, and 2 on the keyboard number pad, corresponding to the four quadrants of the frame (top left, top right, bottom left, and bottom right, respectively). Scores were calculated as the total number of correct responses.

For the paired-associates task, participants were given three lists of 10 word pairs each. All words were common nouns, and the 10 pairs were presented vertically for two seconds each, with the cue word at the top and the target word on the bottom. At test, participants were presented with each of the 10 cue words (in random order) with '???' below, where the target word was originally. For each cue word, participants were given five seconds to type in the associated target word and then press ENTER. The same procedure was used for all three lists, and scores were calculated as the total number of correct responses.

For the delayed free recall, participants were given six lists of 10 words each. All words were common nouns, and they were presented at the center of the screen for one second each. After presentation of all ten words, participants completed a digit-sorting distractor task for 16 seconds, in which eight three-digit numbers appeared for two seconds each, and participants were required to write the digits for each number in ascending order on a piece of paper. After the distractor task, participants were given 45 seconds to type as many words as they could remember from the current list of 10, in any order. The same procedure was used for all six lists, and scores were calculated as the total number of correct responses.

2.3.2. Procedure

This was a 1.5-hour session. All components were completed on the computer (24-inch display), with participants completing all tasks within each domain, starting with the gF tasks, followed by the WMC tasks, and then the LTM tasks. The tasks were completed in the following order: Ravens, number series, letter sets, color change detection, orientation change detection, space task, picture source-recognition, paired-associates, delayed free recall. As in session 1, a trained experimenter used images and examples to instruct participants before each section of the experiment.

3. Results and Discussion

We first inspected average performance profiles across all relevant conditions. As depicted in Figure 4.2, we fully replicated our previous within-individual results with the same set of tasks. In particular, we found that with each additional level came additional performance costs. As in our previous work, number of levels costs were present even for non-transition trials in the serial-order format. In contrast, in the cue-based context costs arose only when the highest-relevant level had to be updated or integrated with the next level down. Performance cost was used here in order to directly compare our results to those of the previous study. However, using drift rate as the outcome yielded qualitatively similar results.

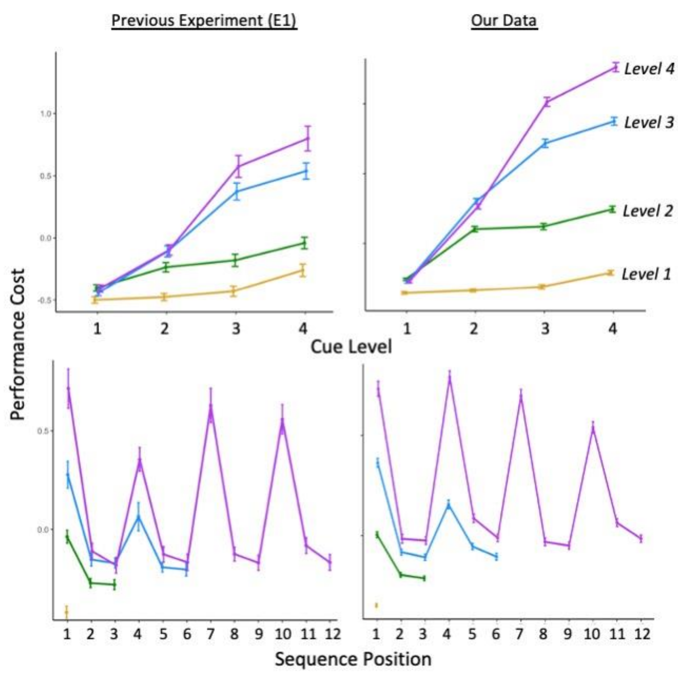


Figure 4.2. Side by side comparison of performance by condition in Experiment 1 of the previous study and data from the current study.

3.1. Drift Diffusion Parameter Estimation

We used task and level to derive all possible conditions. Level here is broken into six steps, based on the questions posed in Section 1.3 and our previous results concerning integration: levels 1 and 2 included all trials completed in their respective task level, while levels 3 and 4 were both split out into integration and non-integration sub-levels. For level 3, non-integration trials were those in which no real updating was required (cue level 1 in cue-based, and within-chunk in serial-order), and all other level 3 trials were labelled as integration. For level 4, non-integration trials were those in which no higher-level updating was required (cue levels 1-2 in cue-based, and within-sequence in serial-order), and all other level 4 trials were labelled as integration. For cue-based and serial-order formats separately, this six-level variable, along with task type (three levels) were used to estimate diffusion model parameters using the maximum likelihood optimization criterion implemented in *fast-dm-30* (Voss, Voss, & Lerche, 2015). Parameters were estimated separately for each participant, each task, and each of the six redefined levels. In line with Lerche et al. (2020), thresholds were associated with correct (1) and incorrect (0) responses, the starting point was centered between thresholds, and the intertrial variabilities of drift rate and starting point were fixed to zero.

From this procedure, we estimated drift rate and threshold separation in all conditions, for both cue-based and serial-order formats. These estimated parameters, in each task level condition, were used as our main observed variables. Before entering them into structural equation models, we standardized all observed variables in order to avoid estimation problems that could arise from the differences in variance of drift rate, threshold separation, and all secondary, psychometric variables.

3.2. Structural Equation Modeling

Before modeling, we again examined performance across all relevant conditions, for both drift rate and threshold separation parameters (see Table 4.1 for measures from the cue-based hierarchical format, Table 4.2 for measures from the serial-order hierarchical format, and Table 4.3 for fluid intelligence, working memory capacity, and long-term memory measures). Most of our measures had generally acceptable skewness and kurtosis values[3], indicating that they are approximately normally distributed, and all measures had generally acceptable reliability or communality scores.[4]

Correlations for the secondary measures (from fluid intelligence, working memory capacity, and long-term memory tasks) were generally stronger between measures for the same construct, and weaker between measures for different constructs.[5] Correlations among the task level variables for both drift rate and threshold separation within each format generally seemed to go up with hierarchical task level, with level 1 variables showing the weakest correlations. For drift rate in both formats, the task level variables showed relatively weak relationships with secondary measures, with some increase in correlations with the higher-level variables. For threshold separation in both formats, the task level variables showed negative correlations with secondary measures, with the strongest negative correlations with level 1 variables, and the relationship lessening with higher-level variables (for correlation matrices, see Tables D1 and D2 in Appendix D).

[3] Drift rate for Dot Rule, Level 1 in the serial-order format showed high kurtosis (5.85). A square root transformation was applied in order to reduce this, and then the difference between old and new means was subtracted from the transformed variable. After transformation, kurtosis=2.17.

[4] Communality reflects the degree of relationship between each variable and all other variables. Although not an ideal measure, it can be used as a crude estimate of reliability (for example, see Engle et al., 1999).

[5] This was not the case for the picture source-recognition task, which may have been due to the fact that we did not include new images in the test phase, as is typically done.

Table 4.1. Descriptives for drift rate (v) and threshold separation (a) parameters estimated for all level and cue levels of each task in the cue-based format.

Parameter	Level, cue level	Task	Mean	SD	Skew	Kurtosis	Communality
Drift Rate (v)	1, 1-4	DR	0.082	0.028	0.773	0.223	.355
		OO	0.095	0.029	0.680	0.096	.363
		NJ	0.062	0.018	1.072	1.500	.295
	2, 1-4	DR	0.051	0.011	0.826	0.764	.488
		OO	0.048	0.009	0.372	-0.377	.480
		NJ	0.037	0.007	0.208	-0.332	.457
	3, 1	DR	0.068	0.018	0.681	0.239	.401
		OO	0.074	0.017	0.644	0.493	.333
		NJ	0.048	0.013	0.468	0.252	.392
	3, 2-4 (3i)	DR	0.027	0.007	0.103	0.414	.502
		OO	0.026	0.008	-0.099	-0.460	.618
		NJ	0.024	0.006	-0.034	0.023	.480
	4, 1-2	DR	0.050	0.011	0.332	-0.075	.541
		OO	0.049	0.009	0.356	-0.251	.450
		NJ	0.037	0.007	0.519	0.594	.506
	4, 3-4 (4i)	DR	0.018	0.008	-0.105	0.042	.527
		OO	0.021	0.008	-0.577	0.209	.599
		NJ	0.022	0.006	-0.296	-0.407	.469
Threshold Separation (a)	1, 1-4	DR	51.035	15.985	0.372	-0.707	.354
		OO	44.152	12.858	0.335	-0.900	.381
		NJ	62.881	15.576	0.323	-0.409	.234
	2, 1-4	DR	77.833	15.984	-0.048	-0.530	.643
		OO	72.166	12.383	0.090	-0.139	.616
		NJ	94.200	17.421	0.081	0.016	.634
	3, 1	DR	65.860	17.276	0.692	0.301	.404
		OO	61.587	12.820	0.570	0.178	.317
		NJ	76.079	18.487	0.532	0.182	.399
	3, 2-4 (3i)	DR	100.767	21.820	-0.091	-0.496	.675
		OO	92.357	17.279	0.060	-0.401	.739
		NJ	114.553	23.130	-0.313	0.026	.677
	4, 1-2	DR	84.542	22.109	0.464	0.039	.672
		OO	77.265	18.619	0.510	0.612	.721
		NJ	92.977	22.110	0.077	-0.488	.696
	4, 3-4 (4i)	DR	99.893	20.837	-0.017	0.186	.673
		OO	97.897	18.221	-0.044	0.106	.632
		NJ	114.398	22.200	-0.670	0.477	.688

Note. DR = Direction Rule; OO = Odd One Out; NJ = Number Judgment. Communality calculated as a crude indicator of reliability separately for each standardized parameter, across level and cue level.

Table 4.2. Descriptives for drift rate (v) and threshold separation (a) parameters estimated for all level and element positions of each task in the serial-order format.

Parameter	Level, element	Task	Mean	SD	Skew	Kurtosis	Communality
Drift Rate (v)	1, all	DR	0.081	0.022	1.625	5.698†	.439
		OO	0.093	0.027	0.842	0.523	.355
		NJ	0.060	0.017	1.332	3.122	.315
	2, all	DR	0.053	0.012	0.705	0.363	.577
		OO	0.060	0.012	0.976	1.799	.374
		NJ	0.041	0.008	0.424	0.567	.379
	3, 1/4/7/10 (3i)	DR	0.050	0.015	0.723	1.504	.590
		OO	0.051	0.013	0.775	2.823	.443
		NJ	0.038	0.011	0.356	0.676	.517
	3, else	DR	0.035	0.011	0.712	1.558	.487
		OO	0.033	0.010	0.119	0.202	.427
		NJ	0.031	0.008	0.168	0.295	.621
	4, 1/7 (4i)	DR	0.032	0.009	0.425	1.820	.682
		OO	0.034	0.010	0.118	1.805	.658
		NJ	0.028	0.008	-0.283	1.764	.527
	4, else	DR	0.019	0.012	-0.350	0.517	.740
		OO	0.024	0.013	-0.174	1.890	.716
		NJ	0.025	0.009	-0.008	0.655	.692
Threshold Separation (a)	1, all	DR	45.945	10.900	0.128	-0.576	.407
		OO	41.188	10.931	0.228	-0.809	.326
		NJ	60.347	14.231	0.186	-0.091	.206
	2, all	DR	74.910	14.268	0.036	-0.282	.521
		OO	66.133	11.012	0.170	0.063	.467
		NJ	91.726	16.592	-0.043	0.003	.508
	3, 1/4/7/10 (3i)	DR	72.766	15.085	0.121	-0.560	.416
		OO	66.552	14.307	0.538	-0.037	.445
		NJ	84.784	18.562	0.280	-0.481	.610
	3, else	DR	83.706	14.437	-0.147	-0.315	.425
		OO	80.636	15.542	0.249	-0.296	.437
		NJ	98.721	19.854	-0.112	-0.352	.470
	4, 1/7 (4i)	DR	93.763	18.546	0.252	0.122	.522
		OO	87.168	18.590	0.259	0.175	.549
		NJ	103.599	24.397	-0.116	0.084	.541
	4, else	DR	92.134	17.338	0.265	0.307	.688
		OO	85.677	18.034	0.135	-0.445	.604
		NJ	100.574	22.060	-0.143	-0.282	.717

Note. DR = Direction Rule; OO = Odd One Out; NJ = Number Judgment. Communality calculated as a crude indicator of reliability separately for each standardized parameter, across level and element.

†A square root transformation was applied in order to reduce kurtosis, and then the difference between old and new means was subtracted from the transformed variable. After transformation, kurtosis=2.17.

Table 4.3. Descriptives for all secondary measured variables.

Factor	Measure	Mean	SD	Skew	Kurtosis	Reliability[b]
Fluid Intelligence	Letter Sets	10.49	2.981	0.188	-0.403	.87
	Number Series	9.41	2.517	0.004	-0.708	.78
	Ravens Matrices[a]	10.46	2.121	-0.594	0.962	.63
Working Memory Capacity	Color K	2.92	0.707	-0.398	-0.244	.81
	Orientation K	1.58	0.698	-0.609	1.531	.87
	Space K	-2.07	0.796	-0.405	-0.109	.92
Long-Term Memory	Pic Source Recognition	29.43	6.052	-0.687	-0.131	.77
	Paired Associates	15.29	7.312	0.135	-0.990	.90
	Delayed Free Recall	26.36	9.621	0.690	0.258	.90

a. Values calculated after imputing values for 49 individuals, because of technical issues. Descriptives were not significantly different from those calculated without the imputed values.

b. Split-half reliability calculation: Spearman-Brown Coefficient. For Ravens, this was calculated only for cases with complete (non-imputed) data.

We used confirmatory factor analysis to determine the latent individual difference structure present in our data. Specifically, we applied a sequence of bifactor measurement models in which standardized task level variables loaded onto level-based latent factors with increasing specificity (depicted in Figure 4.3). These models allowed us to examine whether and which higher hierarchical task levels predict unique variance, by first including a latent variable that is composed of variance common to all task levels, and then progressively including additional latent variables that are composed of only the residual (or unique) variance common to hierarchical tasks with at least two levels, then three, then four, and all their combinations. This sequence of measurement models was applied separately for serial-order and the cue-based formats and for the drift rate and the threshold separation parameters. In all models, method variance was accounted for by allowing covariance of all adjacent-level variables within task.

After running all possible models, it was determined that each of the four hierarchical levels as defined in our paradigm do not contribute unique variance from the levels below, when included in the same model (see Tables 4.4 and 4.5 for fit indices of drift rate and threshold separation models, respectively). However, we did find evidence of two such levels. First, the

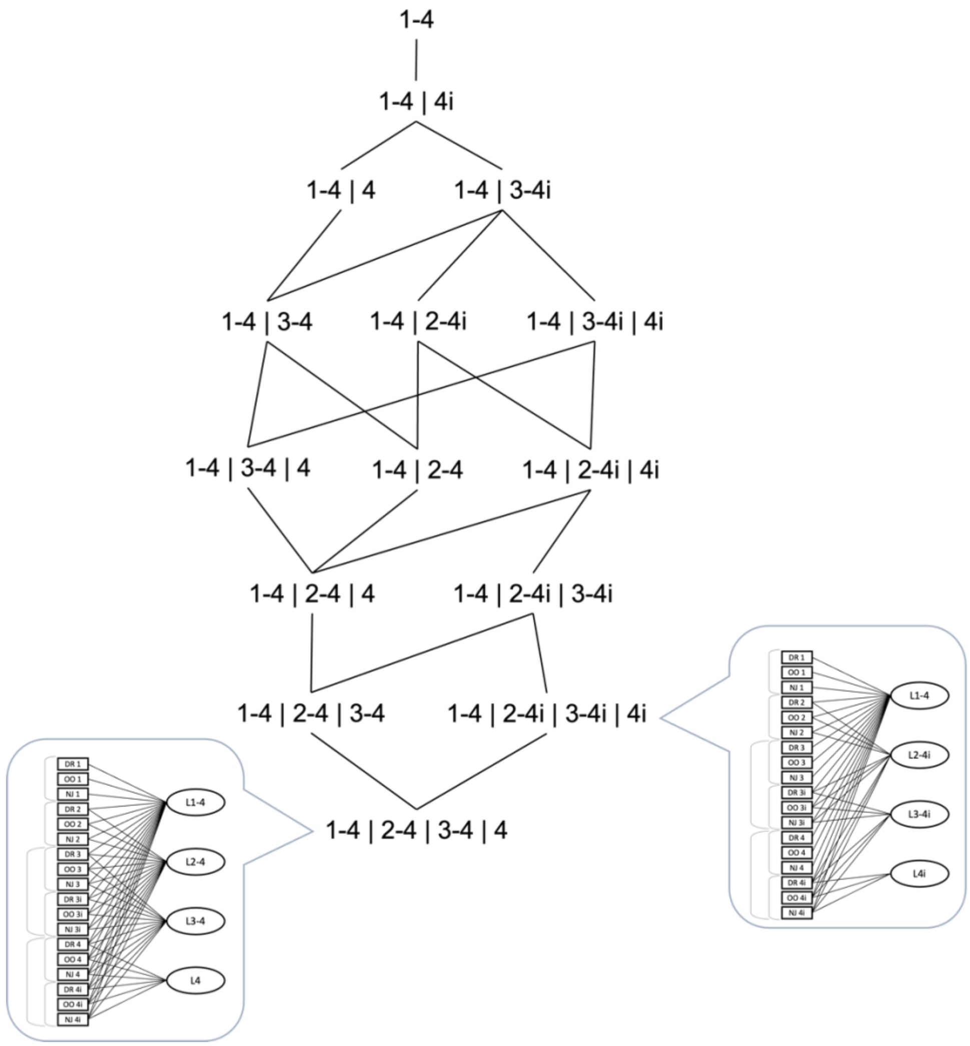


Figure 4.3. Flow chart of SEM testing, with example structures. Each number or range indicates a latent factor for hierarchical level, and 'i' indicates that only integration trials were included for those levels. Latent factors in the same model are separated by a bar. For example, '1-4 | 3-4i' represents a model with two factors: one factor with loadings from all observed variables (common variance), and another factor with loadings from only the integration variables in task levels 3 and 4 (residual variance for those conditions).

general factor accounted for significant shared variance across all levels. This factor will be referred to here as the "lower-level" factor, as it is the only one that includes variance in performance on task level 1. Beyond this, variance specific to the higher task levels (observed variables in levels 2-4) yielded a "higher-level" factor (see Figure 4.4). This means that after accounting for the variance shared across all levels, there was still significant unique variance associated with this higher level. This structure provided the best model fit across format and parameter, indicating that there may be a general representational structure that the cognitive system can flexibly apply to different task spaces. Further, this two-level bifactor model provided a better fit than any other combination, including models with latent factors for only integration variables. This means that there may not be unique individual differences variance for integration processes, even though we found evidence of integration-based processing constraints within individuals.

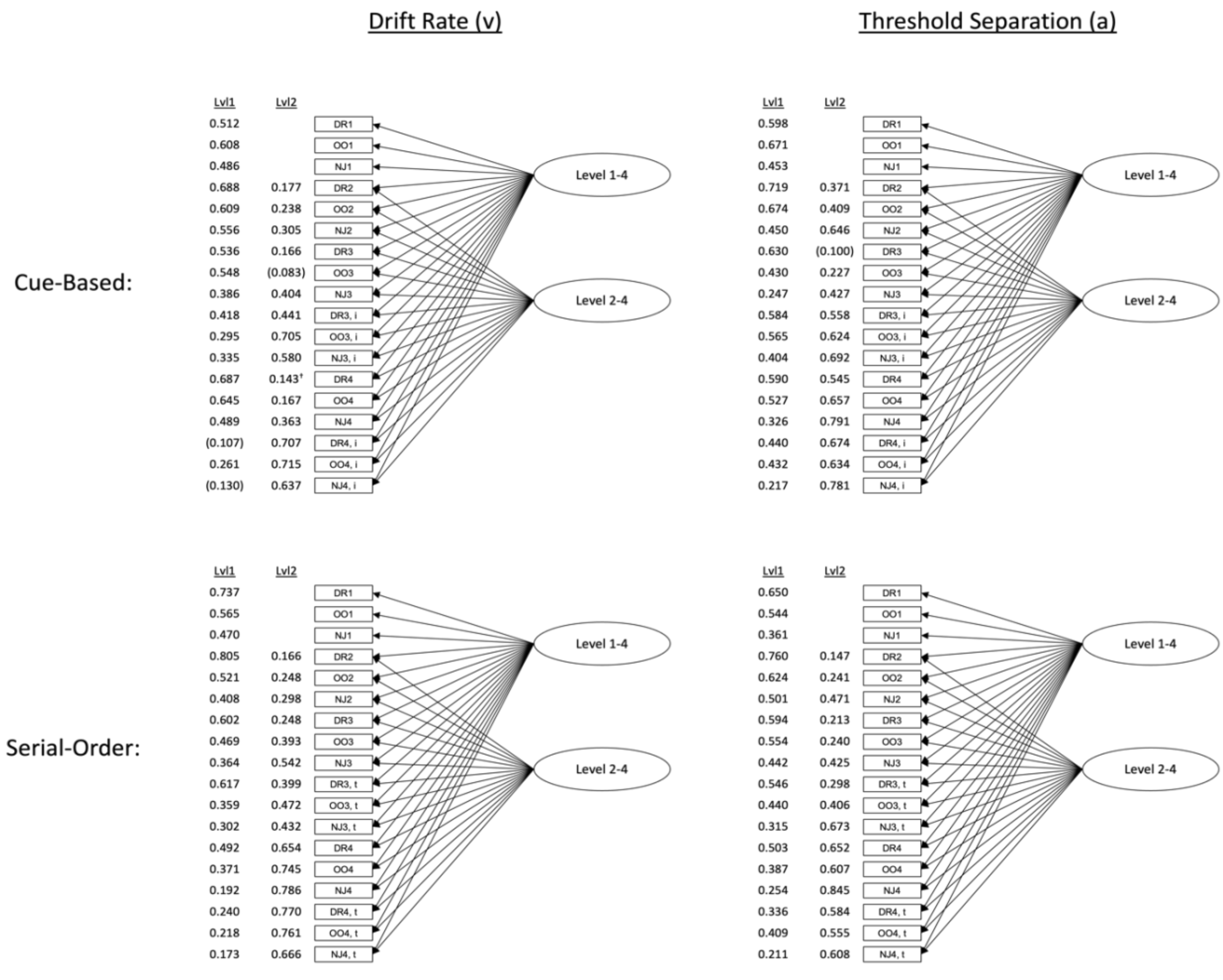


Figure 4.4. Best bifactor model of hierarchical level in each format and parameter. Paths indicate which observed variables (squares) load onto the different latent variables (circles). The numbers to the left of the observed variables represent their standardized loadings onto each of the latent factors (Lvl1 = Level1-4; Lvl2 = Level2-4). Factor loadings in parentheses were not significant, and those with a † were marginal ($p<.10$). All other loadings were significant.

With the “best” model in each format and parameter, we then tested whether the strictest additive model of hierarchical control would fit the data, by applying within-task equality constraints to the loadings on the lower-level factor. In all four models, these results produced significantly worse fit (drift rate: cue-based $X^2(15)=60.02$, $p<.001$, serial-order $X^2(15)=45.18$, $p<.001$; threshold separation: cue-based $X^2(15)=44.98$, $p<.001$, serial-order $X^2(15)=34.23$, $p=.003$). Therefore, we can reject the purely additive model of hierarchical control.

Instead, the two factors showed a level-congruent tradeoff in loadings. The loadings for the lower-level factor decreased as the hierarchical level of the observed variables increased, and the opposite pattern occurred for the higher-level factor. Though this pattern in the lower-level factor was not predicted, the increasing amount of variance captured by the higher-level factor

Table 4.4. Drift rate model fit comparison table.

<table>
<tr><td colspan="9">Cue-Based Format</td></tr>
<tr><td></td><td>Model</td><td>X^2</td><td>df</td><td>RMSEA</td><td>CFI</td><td>NNFI</td><td>AIC</td><td>BIC</td></tr>
<tr><td>1</td><td>1-4</td><td>360.30</td><td>117</td><td>0.10</td><td>0.81</td><td>0.76</td><td>8879.75</td><td>9055.94</td></tr>
<tr><td>2</td><td>1-4 | 4</td><td>308.75</td><td>111</td><td>0.10</td><td>0.85</td><td>0.79</td><td>8840.21</td><td>9035.97</td></tr>
<tr><td>3</td><td>1-4 | 3-4‡</td><td>207.81</td><td>105</td><td>0.07</td><td>0.92</td><td>0.89</td><td>8751.26</td><td>8966.60</td></tr>
<tr><td>4</td><td>1-4 | 2-4</td><td>192.06</td><td>102</td><td>0.07</td><td>0.93</td><td>0.90</td><td>8741.51</td><td>8966.64</td></tr>
<tr><td>5</td><td>1-4 | 4i</td><td>318.99</td><td>114</td><td>0.10</td><td>0.84</td><td>0.79</td><td>8844.45</td><td>9030.42</td></tr>
<tr><td>6</td><td>1-4 | 3-4i</td><td>227.65</td><td>111</td><td>0.07</td><td>0.91</td><td>0.88</td><td>8759.10</td><td>8954.86</td></tr>
<tr><td>7</td><td>1-4 | 2-4i</td><td>225.02</td><td>108</td><td>0.07</td><td>0.91</td><td>0.87</td><td>8762.47</td><td>8968.02</td></tr>
<tr><td>8</td><td>1-4 | 3-4 | 4†</td><td>196.09</td><td>99</td><td>0.07</td><td>0.93</td><td>0.89</td><td>8751.54</td><td>8986.46</td></tr>
<tr><td>9</td><td>1-4 | 2-4 | 4†</td><td>182.12</td><td>96</td><td>0.07</td><td>0.93</td><td>0.90</td><td>8743.57</td><td>8988.27</td></tr>
<tr><td>10</td><td>1-4 | 2-4 | 3-4†</td><td>131.95</td><td>90</td><td>0.05</td><td>0.97</td><td>0.95</td><td>8705.40</td><td>8969.68</td></tr>
<tr><td>11</td><td>1-4 | 3-4i | 4i</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>12</td><td>1-4 | 2-4i | 4i†</td><td>219.42</td><td>105</td><td>0.08</td><td>0.91</td><td>0.87</td><td>8762.88</td><td>8978.21</td></tr>
<tr><td>13</td><td>1-4 | 2-4i | 3-4i†</td><td>186.71</td><td>102</td><td>0.07</td><td>0.94</td><td>0.90</td><td>8736.16</td><td>8961.29</td></tr>
<tr><td>14</td><td>1-4 | 2-4i | 3-4i | 4i</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>15</td><td>1-4 | 2-4 | 3-4 | 4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="9">Serial-Order Format</td></tr>
<tr><td></td><td>Model</td><td>X^2</td><td>df</td><td>RMSEA</td><td>CFI</td><td>NNFI</td><td>AIC</td><td>BIC</td></tr>
<tr><td>1</td><td>1-4</td><td>389.04</td><td>117</td><td>0.11</td><td>0.85</td><td>0.80</td><td>8420.11</td><td>8596.29</td></tr>
<tr><td>2</td><td>1-4 | 4</td><td>284.27</td><td>111</td><td>0.09</td><td>0.90</td><td>0.87</td><td>8327.34</td><td>8523.10</td></tr>
<tr><td>3</td><td>1-4 | 3-4</td><td>241.09</td><td>105</td><td>0.08</td><td>0.92</td><td>0.89</td><td>8296.16</td><td>8511.50</td></tr>
<tr><td>4</td><td>1-4 | 2-4</td><td>217.68</td><td>102</td><td>0.08</td><td>0.93</td><td>0.90</td><td>8278.75</td><td>8503.87</td></tr>
<tr><td>5</td><td>1-4 | 4i</td><td>318.41</td><td>114</td><td>0.10</td><td>0.88</td><td>0.85</td><td>8355.48</td><td>8541.45</td></tr>
<tr><td>6</td><td>1-4 | 3-4i‡</td><td>318.17</td><td>111</td><td>0.10</td><td>0.88</td><td>0.84</td><td>8361.24</td><td>8557.00</td></tr>
<tr><td>7</td><td>1-4 | 2-4i‡</td><td>311.46</td><td>108</td><td>0.10</td><td>0.89</td><td>0.84</td><td>8360.53</td><td>8566.08</td></tr>
<tr><td>8</td><td>1-4 | 3-4 | 4‡</td><td>192.06</td><td>99</td><td>0.07</td><td>0.95</td><td>0.92</td><td>8259.13</td><td>8494.04</td></tr>
<tr><td>9</td><td>1-4 | 2-4 | 4‡</td><td>164.51</td><td>96</td><td>0.06</td><td>0.96</td><td>0.94</td><td>8237.58</td><td>8482.28</td></tr>
<tr><td>10</td><td>1-4 | 2-4 | 3-4‡</td><td>161.96</td><td>90</td><td>0.06</td><td>0.96</td><td>0.93</td><td>8247.03</td><td>8511.31</td></tr>
<tr><td>11</td><td>1-4 | 3-4i | 4i†</td><td>313.35</td><td>108</td><td>0.10</td><td>0.88</td><td>0.84</td><td>8362.42</td><td>8567.97</td></tr>
<tr><td>12</td><td>1-4 | 2-4i | 4i</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>13</td><td>1-4 | 2-4i | 3-4i</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>14</td><td>1-4 | 2-4i | 3-4i | 4i</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>15</td><td>1-4 | 2-4 | 3-4 | 4†</td><td>128.82</td><td>84</td><td>0.05</td><td>0.97</td><td>0.95</td><td>8225.89</td><td>8509.74</td></tr>
</table>

Note. RMSEA = root mean square error of approximation; CFI = comparative fit index; NNFI = non-normed fit index; AIC = Akaike information criterion; BIC = Bayesian information criterion. Fit statistics only reported for models that converged (versus dashes for invalid models).
†At least one of the Level factors in the model did not have significant variance.
‡At least three loadings on an upper-level factor were nonsignificant and not marginal (*p*>.10).

with task level increases is consistent with the graded hierarchical control model.

After determining that both cue-based and serial-order variables fit the same type of individual differences structure, we integrated serial-order and cue-based variables into the same model, to determine whether there is shared variance within hierarchical level, across format. Thus far, we have only managed to identify parallel structures across format. In this next step, we were able to investigate directly how they relate to one another. In this large integrated model, we allowed both cue-based level factors to covary with both serial-order level factors. As shown in Figure 4.5, the analogous levels shared a significant amount of variance, for both drift rate and threshold

Table 4.5. Threshold separation model fit comparison table.

	Model	X^2	df	RMSEA	CFI	NNFI	AIC	BIC
	Cue-Based Format							
1	1-4	292.69	117	0.09	0.92	0.89	7919.84	8096.03
2	1-4 | 4†	267.98	111	0.09	0.93	0.90	7907.13	8102.89
3	1-4 | 3-4‡	218.66	105	0.07	0.95	0.92	7869.81	8085.15
4	1-4 | 2-4	162.55	102	0.06	0.97	0.96	7819.70	8044.82
5	1-4 | 4i†	274.82	114	0.09	0.93	0.90	7907.97	8093.94
6	1-4 | 3-4i	247.48	111	0.08	0.94	0.91	7886.63	8082.40
7	1-4 | 2-4i‡	242.55	108	0.08	0.94	0.91	7887.71	8093.25
8	1-4 | 3-4 | 4	-	-	-	-	-	-	-
9	1-4 | 2-4 | 4	-	-	-	-	-	-	-
10	1-4 | 2-4 | 3-4†	113.49	90	0.04	0.99	0.98	7794.64	8058.92
11	1-4 | 3-4i | 4i†	246.49	108	0.08	0.94	0.91	7891.64	8097.19
12	1-4 | 2-4i | 4i†	239.69	105	0.08	0.94	0.91	7890.84	8106.18
13	1-4 | 2-4i | 3-4i	-	-	-	-	-	-	-
14	1-4 | 2-4i | 3-4i | 4i†	225.35	99	0.08	0.94	0.91	7888.50	8123.42
15	1-4 | 2-4 | 3-4 | 4	-	-	-	-	-	-	-
	Serial-Order Format							
	Model	X^2	df	RMSEA	CFI	NNFI	AIC	BIC
1	1-4	317.14	117	0.09	0.87	0.83	8537.47	8713.66
2	1-4 | 4	284.14	111	0.09	0.89	0.85	8516.48	8712.24
3	1-4 | 3-4	227.63	105	0.08	0.92	0.89	8471.97	8687.31
4	1-4 | 2-4	189.54	102	0.07	0.94	0.92	8439.88	8665.00
5	1-4 | 4i†	310.30	114	0.09	0.88	0.83	8536.63	8722.60
6	1-4 | 3-4i†	293.54	111	0.09	0.88	0.84	8525.87	8721.63
7	1-4 | 2-4i†	287.13	108	0.09	0.89	0.84	8525.47	8731.02
8	1-4 | 3-4 | 4	-	-	-	-	-	-	-
9	1-4 | 2-4 | 4†	177.08	96	0.07	0.95	0.92	8439.41	8684.11
10	1-4 | 2-4 | 3-4	-	-	-	-	-	-	-
11	1-4 | 3-4i | 4i	-	-	-	-	-	-	-
12	1-4 | 2-4i | 4i	-	-	-	-	-	-	-
13	1-4 | 2-4i | 3-4i	-	-	-	-	-	-	-
14	1-4 | 2-4i | 3-4i | 4i	-	-	-	-	-	-	-
15	1-4 | 2-4 | 3-4 | 4†	122.43	84	0.05	0.98	0.96	8408.77	8692.62

Note. RMSEA = root mean square error of approximation; CFI = comparative fit index; NNFI = non-normed fit index; AIC = Akaike information criterion; BIC = Bayesian information criterion. Fit statistics only reported for models that converged (versus dashes for invalid models).
†At least one of the Level factors in the model did not have significant variance.
‡At least three loadings on an upper-level factor were nonsignificant and not marginal (p>.10).

separation, with the strongest relationship between the lower-level factor in cue-based and serial-order formats. Importantly, even for the higher-level factors in which variance profiles may be more different across format because of the requirements of the cue-based and serial-order tasks on these levels, a large amount of the variance was shared across format. This finding indicates that not only do individual differences in performance of hierarchical tasks conform to analogous latent structures across task format, but they actually share variance within levels. Clearly there is a strong relationship between performance of tasks defined by external cues and performance of tasks defined by memorized sequences, and such relationships exist for both level-general and level-specific variance.

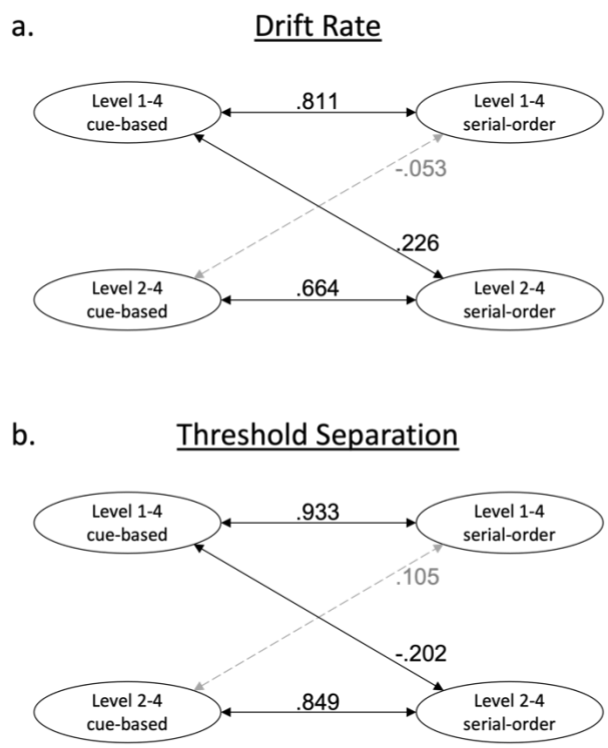


Figure 4.5. Structural models of cross-format relationships for (a) drift rate, and (b) threshold separation. Double headed arrows connecting the latent factors represent correlations between the constructs. Gray dashed lines represent non-significant paths. All other paths were significant.

3.2.1. Alternative Models

The modeling procedure yielded a clearly winning structure for threshold separation. However, the results were not quite as strong for drift rate. In both cue-based and serial-order formats, there were other models that proved to be close contenders. In the cue-based format, the alternative model was the one in which the higher-level latent factor contained loadings only to the integration variables for levels three and four. In line with our predictions concerning updating or integration costs in the cue-based format, this indicates that there may be variance unique to upper-level integration processes, at least in cue-driven hierarchical tasks. For serial-order, a similar "runner up" model was identified, in which the higher-level latent factor contained loadings to all observed variables in levels three and four (including both integration and non-integration variables). Though these models were close in fit, based on the order of model testing and on the results of chi-square tests, it was determined that the models discussed in the previous section were "better." It is important to note, however, that this finding should be interpreted carefully, govem that, with higher sample sizes, additional differenitations along the lines hinted at through the close contender models might have easily emerged. Further, determining the variance structures of these hierarchical task parameters might depend on the influence of other factors not accounted for in this initial set of tests. Therefore, though we will continue here with the assumption that the '1-4 | 2-4' model is currently the most parsimonious for all four parameter-by-format model sets, it should be understood that this would need to be replicated in future work in order to determine with more confidence the true variance structures of hierarchical settings.

3.3. The Impact of Fluid Intelligence, Working Memory Capacity, and Long-Term Memory

Once we identified the best model in each format and parameter context, we added all the secondary measures into each of the final models and allowed their latent factors to covary with each other and with both level factors. The goal of this was twofold: We wanted to determine whether hierarchical levels (as defined by their latent factors) maintain unique variance, even after accounting for shared variance with other related constructs, and if so, to characterize the relationships among them. All four models indicated that the two hierarchical level factors are unique from fluid intelligence (gF), working memory capacity (WMC), and long-term memory

(LTM) constructs. Further, the factor loadings remained largely unaffected by the addition of these new factors: The same loading patterns identified in the levels-only models were again observed in these full models.

The pattern of relationships between level factors and the secondary measures was also consistent across format, but not parameter. The drift rate models showed positive correlations between the level factors and psychometric factors (see Figure 4.6). In both cue-based and serial-order formats, these correlations were higher for the higher-level factor, indicating that the other cognitive processes become more important to the rate of evidence accumulation in decision making as hierarchical task level increases, but that they are involved regardless of task level. Interestingly, a very different pattern emerged in the threshold separation models. For both cue-based and serial-order formats, there were negative correlations between the lower-level general factor and all psychometric factors, but positive correlations between the higher-level factor and all psychometric factors (see Figure 4.7). This indicates that the amount of information required for a given decision may be more sensitive to other cognitive processes, such that individuals who have higher gF, WMC, and/or LTM ability may be able to more effectively adjust their decision-making thresholds to be closer (less information needed) when dealing with lower-level processes and farther (more cautious; more information needed) when using the cognitive resources necessary to deal with higher-level processes.

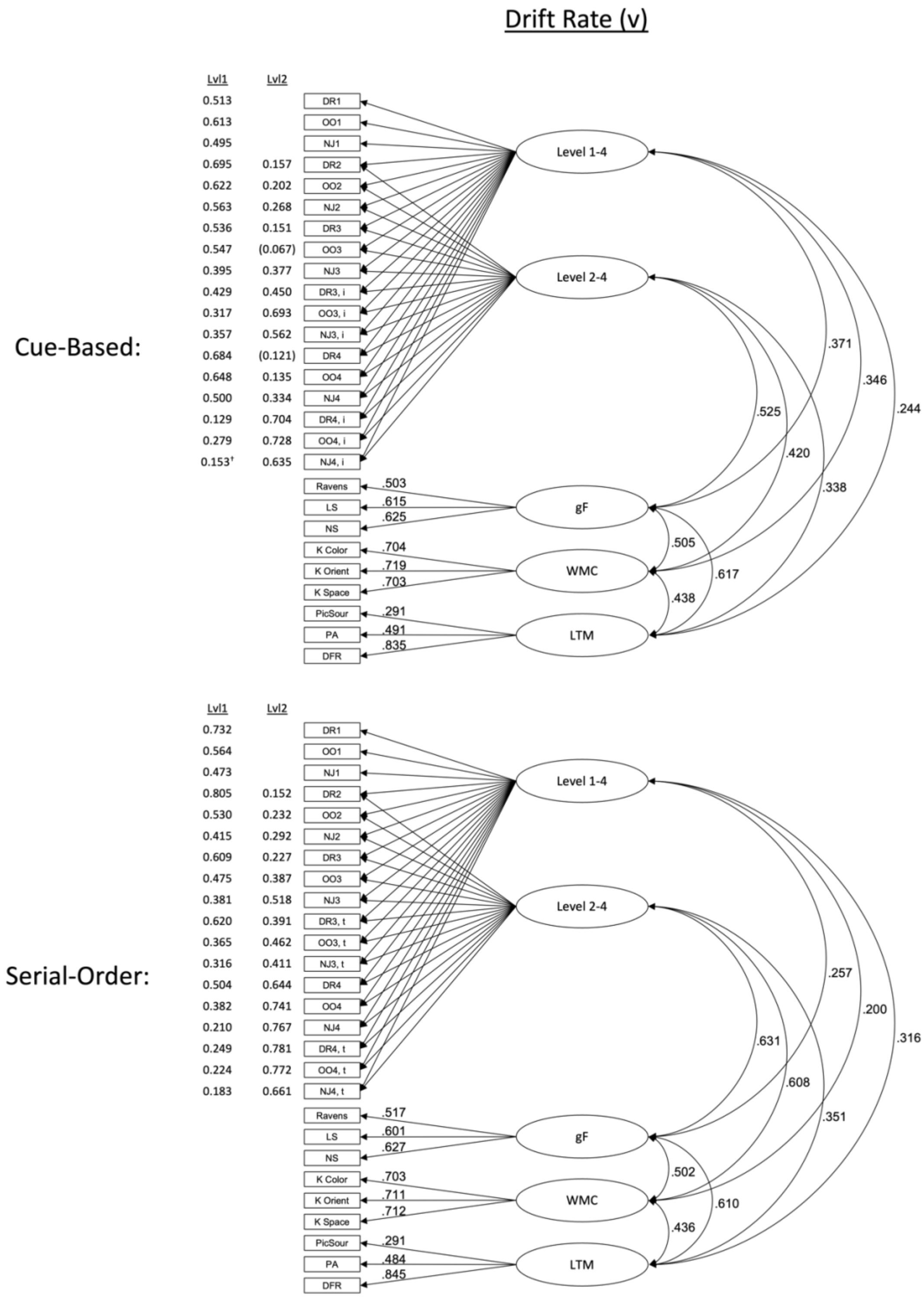


Figure 4.6. Best drift rate models with fluid intelligence (gF), working memory capacity (WMC), and long-term memory (LTM). Paths connecting latent factors represent correlations between them. Paths between observed and latent factors indicate which observed variables load onto which factors. The numbers to the left of the observed variables represent their standardized loadings onto the factors (Lvl1 = Level1-4; Lvl2 = Level2-4). Factor loadings in parentheses were non-significant, and those with a † were trending (p<.10). All others were significant (p<.05).

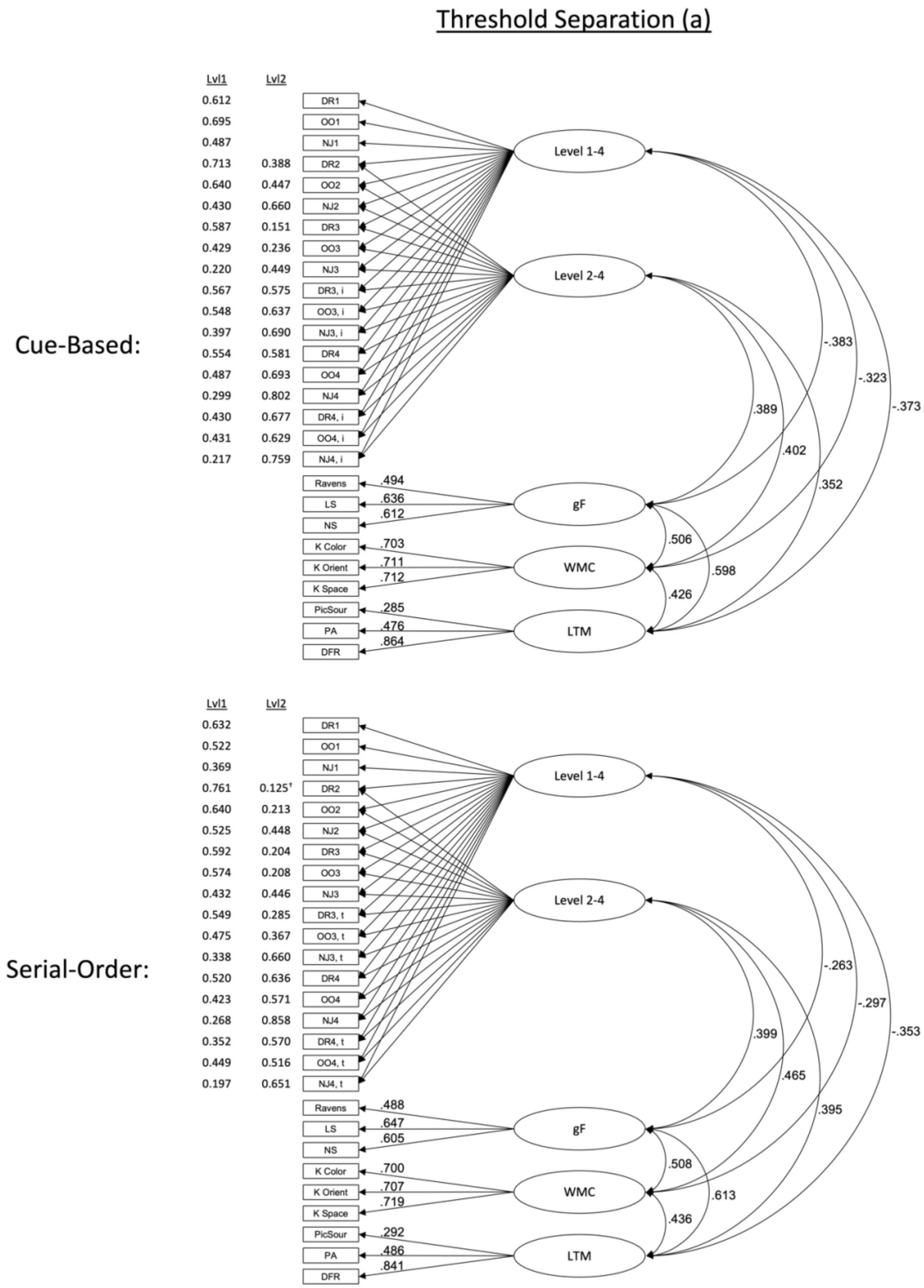


Figure 4.7. Best threshold separation models with fluid intelligence (gF), working memory capacity (WMC), and long-term memory (LTM). Paths connecting latent factors represent correlations between them. Paths between observed and latent factors indicate which observed variables load onto which factors. The numbers to the left of the observed variables represent their standardized loadings onto the factors (Lvl1 = Level1-4; Lvl2 = Level2-4). Factor loadings in parentheses were non-significant, and those with a † were trending (p<.10). All others were significant (p<.05).

Taking these relationships a step further, we looked at the amount of variance in each of the level factors accounted for independently by each of the different additional constructs using structural regression. This allowed us to parse out the direct effects of each of the three other cognitive constructs on performance in the different hierarchical levels.

When holding the others constant, none of the cognitive constructs significantly predicted performance on the lower-level factor, indicating that whatever processes are common to performance in all hierarchical task levels are not affected by any single cognitive construct alone. Again here, the pattern of results was consistent within estimated parameter, across both cue-based and serial-order formats. In the drift rate models, the relationship between the higher-level factor and both WMC and gF remained significant in structural regressions, indicating that WMC and gF may each account for unique variance in higher-level processing efficiency (see Figure 4.8). For threshold separation, only the path from WMC to

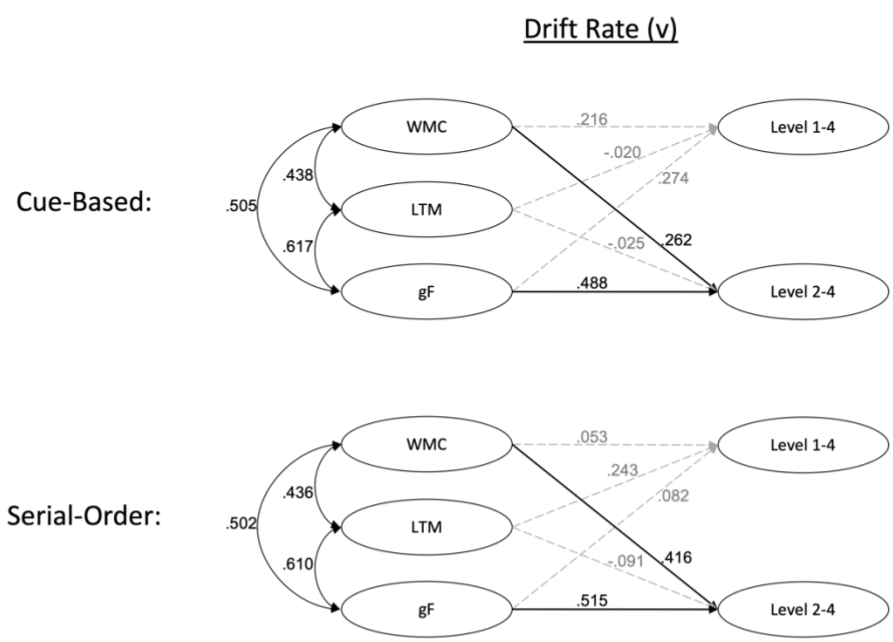


Figure 4.8. Structural regression models for drift rate in both formats, with working memory capacity (WMC), long-term memory (LTM), and fluid intelligence (gF) predicting both Level factors. Single headed arrows connecting the latent factors represent standardized path coefficients, i.e., the amount of variance in one factor that is predicted directly by the other factor. Gray dashed lines represent non-significant paths. All other paths were significant.

the higher-level factor remained significant in structural regressions for both cue-based and serial-order formats, meaning that WMC may be the only of these three cognitive abilities that independently contributes to the process of determining decision threshold when handling higher hierarchical levels (see Figure 4.9).

These relationships cannot account for the specific patterns of shared contributions from WMC, LTM, and gF, and they do not allow for any strong general interpretation about the directional relationships between these constructs. However, they do indicate that gF and WMC are independently involved in higher-level processing efficiency, while only WMC is independently involved in determining decision thresholds in the higher level. Although none of these cognitive abilities factors could predict variance in the lower level factor on their own, it is important to remember that they showed significant positive correlations in the non-directional models of drift rate and significant negative correlations in

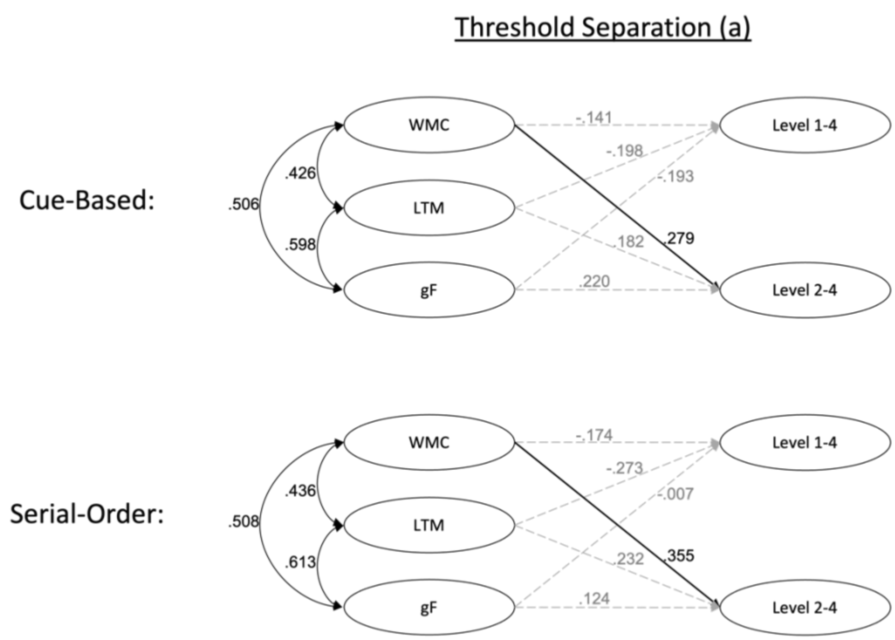


Figure 4.9. Structural regression models for threshold separation in both formats, with working memory capacity (WMC), long-term memory (LTM), and fluid intelligence (gF) predicting both Level factors. Single headed arrows connecting the latent factors represent standardized path coefficients, i.e., the amount of variance in one factor that is predicted directly by the other factor. Gray dashed lines represent non-significant paths. All other paths were significant.

the non-directional models of threshold separation. Taken together, this implies that individuals with higher cognitive abilities generally tended to have lower decision thresholds when handling lower-level information (i.e., require less information to make the easier decisions), and higher decision thresholds when faced with more complex task requirements. Higher working memory capacity may also uniquely augment this ability further, to increase decision threshold when handling higher-level processes. However, the ability to flexibly shift strategy depending on hierarchical levels (as indicated by the combined pattern of negative correlations with lower level and positive correlations with higher level) appears to arise mostly from the shared effects of different cognitive abilities on decision threshold.

4. Conclusions

In this project, we attempted to use individual differences information to characterize the architecture of hierarchical control. Our empirical approach would have allowed up to four different levels of control to emerge across three different task domains and two distinct modes of inducing control structures (cue-based and serial-order). In addition, we also tried to differentiate between trials during which control levels did and did not require updating, and to parse out two different performance aspects, namely processing efficiency (i.e., drift rate) and decision threshold, that were derived from drift diffusion modeling of individuals' RT and accuracy information. Our theoretical starting point was the "standard" model of hierarchical control, in which control flows ballistically from higher to lower levels (Miller et al., 1960).

Across condition and parameter constellations, our structural equation modeling provided relatively clear evidence for only two levels. This was expressed in terms of separate bifactor models for each modality/parameter constellation, in which the first factor captures variance common to all task levels and control schemas and a second, independent "hierarchical control" factor captures the unique variance common to any condition in which at least one additional level needs to be considered.

While the validity of the bifactor model is consistent with some degree of modularity between the two levels of organization, the hypothesis of complete additivity could be clearly rejected. Specifically, the loadings on the lower-level factor declined across task levels, indicating that the addition of higher task levels affected lower-level processing. At the same time, loadings for the second, hierarchical control factor increased across task levels. This pattern indicates that while additional task levels beyond the second level did not generate new, unique variance, these additional levels did provide a more robust representation of the common hierarchical control variance. This pattern is most consistent with the concept of a common resource that is required whenever hierarchical control comes into play and becomes the more critical as more of it is required (with level increases). It also appears to be the case that individual differences variance does not follow the expected pattern in terms of integration costs, in which level-specific individual difference variance would become apparent only on trials for which higher-level representations required updating in the cue-based format. Unlike our previous finding that performance is affected by the need to integrate cross-level information for cue-based hierarchical tasks, the best models here did not appear to differentiate between updating and non-updating trials. This implicates global control costs as more important for individual differences in performance within hierarchical control structures.

In line with the standard model, much of the neuroscience research on hierarchical control is consistent with an anatomical gradient that tracks roughly with hierarchical task level. Our evidence of just two levels of individual differences variance, where the second level captures any hierarchical control variance, does not fully support this view. Instead, the emergence of a hierarchical control factor with increasing loadings across levels is most consistent with a multi-demand system view as proposed by Duncan and colleagues (2010). However, while we can be fairly confident about the emergence of the two-factor separation in our paradigm, including the increase of relevance of the second factor depending on number of levels, we need to exert some caution regarding the rejection of more complex models. Our statistical power was limited for discovering variance unique to higher levels, or for updating versus non-updating events. That said, our findings do indicate that individual differences in hierarchical control likely involve some combination of a non-strict additive model and the graded workspace model.

The finding that the best fitting model structure was the same for both cue-based and serial-order task formats indicated that the two formats conform to the same individual differences patterns. In a more direct assessment of the relationship between performance differences in these two formats, we integrated them into the same model and allowed the latent variables to covary across format. These models provided an answer to the question of whether performance in the two different hierarchical task formats is related. Clearly, much of the variance in performance accounted for by each of the Level factors is actually shared between both cue-based and serial-order formats. This means that not only are the performance patterns similar across hierarchical format, but they are directly related. Additionally, the lower-level factors shared more variance across format than the higher-level factors, indicating that most of the common variance across all levels is more strongly related across format, while some of the higher-level factor variance may still rely on format-specific resources.

The best model structure across task formats was also shared across drift diffusion parameters. For both drift rate and threshold separation, the same two-level model yielded similar results: Loadings onto the lower-level factor decreased with task level, while loadings onto the higher-level factor increased with task level. These similar loading patterns might lead one to believe that the different drift diffusion parameters are not really capturing separate decision-making processes in hierarchical task execution. However, the distinct roles of drift rate and threshold separation become clear when viewed in relation to other related cognitive constructs.

One important finding of this work is the relationships between individual differences in performance on different hierarchical levels and individual differences in WMC, gF, and LTM, for different components of the decision-making process. Individual differences in processing efficiency on both hierarchical levels showed positive relationships with the other cognitive constructs (better processing efficiency related to higher scores in the other factors). Further, the independent contributions of working memory capacity and fluid intelligence variance appeared to be particularly relevant for the higher level. On the other hand, decision threshold showed a negative relationship between the lower-level factor and other cognitive constructs, and a positive relationship between the higher-level factor and other cognitive constructs. Here, working memory capacity also provided additional, independent influence on the higher level.

The correlations between factors in Figure 4.7 show that the three constructs are negatively related (at similar strengths) to the lower-level factor and positively related (again, at similar strengths) to the higher-level factor. However, in separating the contributions of each of the three cognitive constructs, only WMC showed a significant independent effect on the higher-level decision threshold process. One interpretation of this finding would be that individuals with higher WMC are more likely able to flexibly shift their strategy depending on which hierarchical level they are dealing with, with lower-level tasks requiring less information, and therefore allowing the decision threshold to lower. There may also be a shared effect of WMC, LTM, and gF when dealing with higher levels of the hierarchical tasks. The shared effects of the different cognitive constructs on decision threshold would indicate that individuals with higher cognitive abilities are more willing (or able) to take on increased risk in the form of higher uncertainty at the decision point, in the lower levels of a hierarchical task, but can adjust this threshold to be more conservative when dealing with the higher, more complex, levels. Together, these findings provide a distinction in the roles of related cognitive processes in different hierarchical levels of control, and demonstrate the importance of parsing out the different aspects of decision-making in order to gain a fuller understanding of individual differences of hierarchical control.

This research was supported by NSF grant 1734264 (PI: Mayr).

CHAPTER V

GENERAL DISCUSSION

The goal of this dissertation was to characterize hierarchical control structures. Though the topic of hierarchical representations is important in the field of cognitive psychology, many questions remain. Specifically, what are the connections between hierarchical components? How do these relationships help or hinder an individual in trying to execute a hierarchical task? And how does performance vary across task formats, across hierarchical task levels, and across individuals?

Traditional models indicate that the different hierarchical components and levels are represented by the cognitive system as distinct, ballistic, and modular. Such a design would indicate a "divide and conquer" approach to complex, hierarchical information, which would allow the system to completely avoid interference across components. By insulating components from each other, the lower levels of the hierarchy could be "programmed" by higher levels, which would increase the overall efficiency with which a hierarchical structure could be traversed. On the other hand, a completely modular system would not be able to take advantage of relationships between hierarchical components. For hierarchical tasks in which these types of connections exist, it would be more beneficial to allow the integration of information across components, in order to identify and use the relationships in a way that could allow more efficient storage and retrieval of related items. I have shown here that although there is some degree of independence across hierarchical components, cognitive representations of complex tasks also utilize integration and interference processes. Our cognitive system seems to benefit from both modularity and integration by allowing a compromise between the two different processes. Chapter 2 used a unique variant of an explicit sequencing paradigm to demonstrate that the cognitive system allows the sharing of information across different chunks in complex sequences. In a modular system as described by Miller and colleagues (1960), the contents of each chunk should be fully insulated from each other chunk. However, we found that the cognitive system can identify patterns in the chunks and utilize them to allow better sequence execution. Further, this process of identifying and integrating information across chunks is not learned gradually, but instead, happens right from the outset. This indicates that not only is the cognitive system able to relate hierarchical components to one another, but it can do so automatically.

In Chapter 3, I broadened the scope from within-level modularity to across-level integration. Findings revealed an interesting structure of cognitive costs in hierarchical control. According to traditional hierarchical models, cognitive costs should only have accrued as a result of level-specific updating requirements. However, we found that in cue-based hierarchical tasks, the cognitive system required integration of multiple levels, such that even if a higher-level setting has already been established, a cue on the level below would still trigger a process in which hierarchical task information needed to be integrated across level before determining a response. In serial-order hierarchical tasks, this integration cost was present on every trial, indicating that simply moving from position to position within a complex sequence is enough of a change to require reintegration of multiple levels of information. This work provided an example of non-ballistic hierarchical control processes, across both serial-order and cue-contingent contexts.

Chapter 4 built on the previous chapter, taking on an individual differences approach. Here, a hierarchical structure with independent, modular levels would most likely take the form of a set of additive "level" latent factors, starting with a general level accounting for common variance across all levels. Each level above this would represent the unique variance in performance on that level, after accounting for the common variance in the level(s) below. Further, the lower level or levels should be insulated from effects of higher levels. We did not find this to be the case. In fact, there appeared to be an inverse relationship between levels, in which, as loadings increase for one level factor, they decrease for the other. These levels may still represent a hierarchical performance structure, but with more relaxed additivity requirements. Additionally, variance in related cognitive constructs significantly predicted performance in the different hierarchical levels.

Characterizing the Relationships within a Hierarchical Task Structure

Through these chapters, I have addressed the questions raised in the introduction. The first question concerning connections between hierarchical components has been heavily debated. Previous theoretical models (e.g., Cooper & Shallice, 2000; Miller et al., 1960) and empirical work (e.g., Badre & D'Esposito, 2007; Povel & Collard, 1982) have argued that the cognitive system divides complex tasks into subcomponents, which occupy distinct representational subspaces. This division into independent "levels" of information protects lower-level decisions from the cognitive demands of handling higher-level information. For instance, when playing a song on the piano, the execution of individual notes is "programmed" by higher-level information (e.g., what song you are playing, where in the song you are, etc.). Thus, the action of pressing the correct key on the piano is informed by previously set higher-level information, but that information does not need to be retrieved in its entirety for each lower-level decision of which key to press. Though there are some convincing arguments for this idea of independent levels in the representation of hierarchical tasks, others have demonstrated that this may not always be the case. Some have argued for a fully non-hierarchical representation in which all represented components are connected through mechanisms such as associative chaining (Botvinick & Plaut, 2002), adaptive context maintenance (Reynolds et al., 2012), or representation of all information within a global workspace (Dehaene et al., 1998; Waltz et al., 2000). Still others have argued that hierarchical representations may not be arranged in perfectly distinct levels (e.g., Farooqui et al., 2012; Yokoi & Diedrichsen, 2019), or that the cognitive representation of hierarchical components may in fact include information concerning relationships between them, meaning that they are not fully independent of one another (e.g., Amalric et al., 2017; Dehaene et al., 2015; Restle, 1970).

In Chapter 2, I provided evidence in favor of this last idea. Though serial-order processing requires a hierarchical representation of the complex sequence, broken down into subsequences, or chunks, of elements, these chunks are not fully insulated from one another. In fact, the cognitive system is able to automatically detect abstract patterns in the chunks and in this way, can identify similarities across chunks, leading to better execution of sequences containing chunks that share patterns. In Chapter 3, I demonstrated that a conventional ballistic updating model in which higher hierarchical levels "program" lower levels did not fit the pattern of performance costs. Instead, some information is shared and integrated across hierarchical level, even when the updating of settings may not be "strictly" necessary. In Chapter 4, I found that individual differences in performance do not fall cleanly into levels that parallel the level structure of the hierarchical task being executed, as would have been expected according to the traditional model of hierarchical control.

The second question addressed here concerns the utility or cost of these relationships. As previously stated, my findings in Chapter 2 supported the idea that there are performance benefits to the cognitive system being able to identify relationships between chunks in a complex sequence. In Chapter 3, I modeled different processes by which hierarchical information is used, in order to determine where processing costs arise. Though it may be necessary to do so, the integration of information across levels is costly in terms of task performance. Fortunately, the cognitive system seems to be able to maintain a conjunctive representation of some higher-level information, at least in a cue-contingent context, so that integration is not required across all levels at every decision point.

Does Hierarchical Control Generalize across Situation?

The third question is fairly broad in scope, concerning how performance varies across contexts, across hierarchical levels, and across individuals. In Chapter 3, I compared the pattern of processing constraints in two different contexts, using both cue-based and serial-order task formats. Importantly, the two formats yielded different patterns: The serial-order format showed evidence of updating cross-level information at every decision-point, regardless of whether higher-level information (e.g., chunk identity) had changed, while the cue-based format showed evidence of integrating cross-level information only when adjacent levels required updating, with further-away level information remaining set, when possible. This interaction of hierarchical structure size and decision level appear to determine the pattern of performance costs in a cue-based context. In Chapter 4, I again compared cue-based and serial-order performance, but in an individual differences context. Here, we found that, although there may not be a direct task-level-to-representation-level relationship, performance variance did fall into lower and higher level factors. After accounting for variance common to all the tasks and levels, a unique higher level could be identified. This structure of individual differences was the same across different components of the decision-making process, as well as the different hierarchical task formats.

Additionally, the level factors showed distinct relationships with other cognitive constructs. However, both levels remained distinct, even after accounting for the variance they shared with related cognitive processes, indicating that hierarchical processing is unique, and not simply a combination of these other related cognitive processes.

This important and often ignored question in hierarchical control about whether different types of hierarchical tasks rely on the same underlying structures is not easy to answer. Though Chapter 3 provided ambiguous results in this aspect, Chapter 4 showed clearly parallel individual difference structures across serial-order and cue-based contexts. Given the mixed results across these two studies, clearly the question of how different hierarchical formats relate to one another and whether they are established from the same building blocks is a gap in the current literature around hierarchical control that needs to be addressed.

Limitations and Future Directions

So, where do we go from here? In addition to expanding the research around hierarchical control in different contexts, there are many lingering questions that future research needs to address. This work concerns hierarchical control only in explicit contexts. In all the experiments presented here, task rule structures were explicitly instructed. The findings cannot be used to address any questions of gradual learning and implicit acquisition of hierarchical structures. The focus of these studies has been about execution of a given hierarchy and not the application of a hierarchical structure onto a neutral task. For instance, tasks in which individuals are not presented with a hierarchical rule structure and need to spontaneously impose a hierarchy on the provided information could yield different results. The mechanism by which individuals identify situations in which hierarchical structures are useful is unknown. However, given our finding that the cognitive system automatically extracts and utilizes abstract patterns in the chunks of a complex sequence, it is possible that there are other ways in which humans are programmed to handle such complex information in the form of hierarchical representations.

Conclusion

An important contribution of this paper is the direct assessment of different types of hierarchical structures, which are usually presented in disparate literatures. Serial-order and cue-driven hierarchical tasks are typically either assumed to utilize the same cognitive representation structures, or their potential relationship is completely ignored. I used a multi-level, multi-format, multi-task paradigm in order to directly address the question of how the structures of these different task formats relate to one another, and to bring an important and under-studied question to the surface. There appear to be parallels between serial-order and cue-based hierarchical structures, but they may also possess some features that are specific to the requirements of their different contexts. Beyond the relationships between types of hierarchical task structures, I have provided evidence of the ways in which our cognitive system can balance the disparate processes of modularity and integration, enforcing some independence between hierarchical components, but also allowing useful information to be shared across them. This arrangement is advantageous and may be what allows individuals to more efficiently represent and use hierarchical information.

APPENDIX A

INSTRUCTIONS SCRIPT USED IN EXPERIMENT 1 OF CHAPTER 3

Instructions for Hierarchical Control Experiment

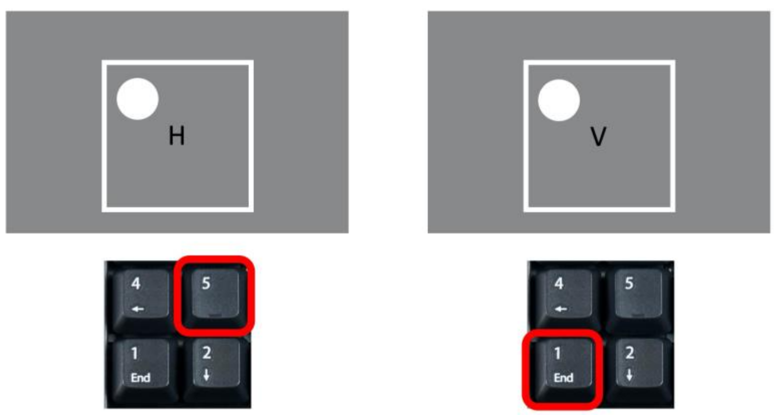
1. Cued Switching (first half)

For this experiment, you will be doing 3 different tasks. I'm going to explain the first 2 to you and then let you get through those before I explain the third task, which is a little different.

There will be a white frame on the screen for all 3 tasks, and all the stimuli will be displayed inside this frame.

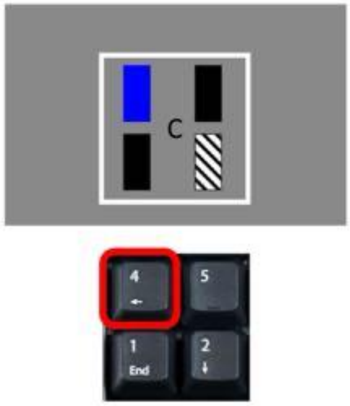
For the first task, a white circle will appear in one of the 4 corners of the white frame, and you will need to indicate where the dot should end up after it is moved according to the specific rule (either horizontal or vertical) that is displayed at the center of the screen. The rule will be represented by a letter: H for horizontal and V for vertical. The rule letter will appear first, and then both the dot and the rule letter will stay on the screen until you respond, using the 4, 5 1, and 2 on the number pad to represent the corresponding corners of the frame (4=top left, 5=top right, 1=bottom left, 2=bottom right).

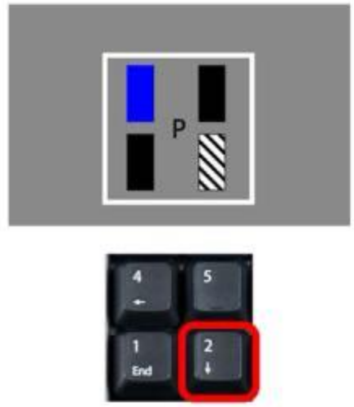
[Go over example with H vs. V, below]



For the second task, you will see 4 rectangles in the frame: 1 in each quadrant. Three of the rectangles will be black (1 will be colored), and three will be solid (1 will have a pattern, e.g., stripes). The color and pattern 'odd ones out' will always be different (there won't be 1 rectangle that is different in both color and in pattern). For this task, you will need to indicate which rectangle is the 'odd one out' based on the rule displayed at the center of the screen (either color or pattern). As in the first task, the rectangles and the rule letter (C for color and P for pattern) will stay on the screen until you respond, using the 4, 5 1, and 2 on the number pad to represent the corresponding quadrants of the frame (4=top left, 5=top right, 1=bottom left, 2=bottom right).

[Go over example with C vs. P, below]





If your response is incorrect, the rule letter at the center of the screen will turn red, and you will need to answer correctly before moving on to the next trial.

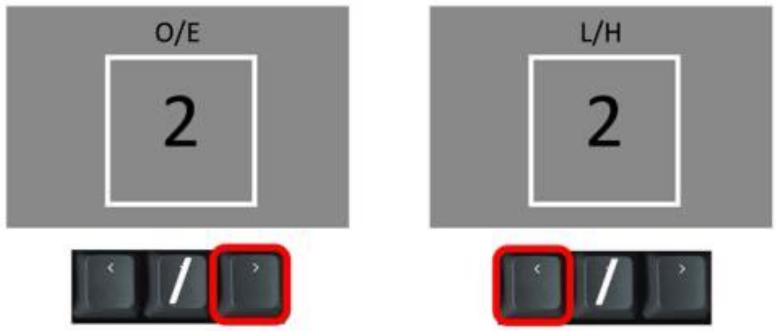
For every block that you have >95% accuracy, you will earn money (in addition to the base rate of $10/hour).

This part of the experiment will take 5-10 minutes. After you have completed both tasks, please let me know so I can explain the third task.

After they get to the stop screen for Number Judgment task:
Now you are going to do a third task. In this one, you will be judging a number that appears in the middle of the screen. The number will be between 1 and 9 (excluding 5), and you will be assessing the number in 1 of two possible ways: whether it is lower or higher than 5 (L/H), or whether it is odd or even (O/E). You will do this using the left and right arrow keys (low=← & high=→ I odd=← & even=→). A helpful way to remember this is, the word is to the left of the slash is the left arrow press, and the word to the right of the slash is the right arrow press.

The judgment you are supposed to make will appear above the number (either L/H for ‘low/high’ or O/E for ‘odd/even’). Both the number and the rule letters will stay on the screen until you respond (same as in the first 2 tasks).

[Go over example with O/E vs. L/H, below]



If your response is incorrect, the rules above the number will turn red, and you will need to answer correctly before moving on to the next trial. As with the first 2 tasks, you can earn additional money for every block that you have >95% accuracy. This part of the experiment consists of 6 blocks and should take 5 minutes.

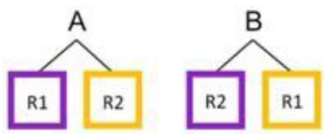
<u>2. Cue-Based Format</u>

Now you are going to do the same 3 tasks, but with a twist: There are 4 ‘levels’ of each task.

On the first level, you will use a single rule for the whole block. The rule will be given to you at the beginning of the block (it will no longer appear at the center of the screen for every trial).

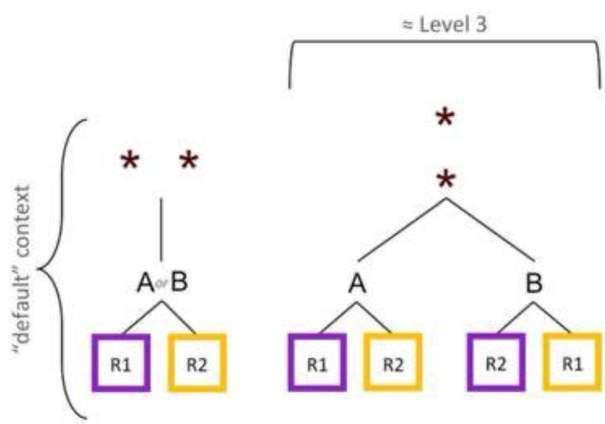
For the second level, frame color will indicate which rule to use for each trial. For instance, if the frame flashes purple, you are supposed to use the odd/even rule to judge the number in the middle, and if it flashes yellow, you are supposed to use the low/high rule. It is important to note that you’ll only see the frame color approximately every 3 trials. *[give example]*

For the third level, there will be a context letter above the frame, which will indicate which color goes with which rule. For instance, if there is an ‘A’ above the frame, purple frame means use odd/even rule and yellow frame means use low/high, but if there is a ‘B’ above the frame, the color-rule pairing is flipped, such that purple means low/high and yellow means odd/even. So on level 3, you need to pay attention to both the letter and the frame color.



As with the frame color, you will not see the context letter every trial- it will appear approximately every 6 trials. This means that you need to keep in mind which rule context you are supposed to be following. For instance, if the ‘A’ shows up and the frame flashes purple, you would do whatever rule purple indicates in A context. Then, in the next trial, if you don’t get any cues, you will use the same rule you used for the trial before. If the frame flashes yellow in the following trial (but the context letter doesn’t appear), you’re still in context A and should use the yellow rule for that context.

The fourth level will look like the third level, with letter above the frame indicating which context to use. However, you will also need to pay attention to the stars around the context letter. If the stars are above and below the letter (vertical), you will do the same thing you did on level 3, with letter indicating which frame color goes with which rule. However, if the stars are on either side of the letter (horizontal), you will use one frame-rule pairing context (default context), regardless of what letter appears above the frame. For instance, using the same example from level 3, with ‘A’ indicating purple means use odd/even rule and yellow means use low/high, and ‘B’ indicating purple means low/high and yellow means odd/even... The default context in this case is the same as context ‘A,’ so if the stars are **horizontally** oriented, you’d use the ‘A’ context, regardless of whether the letter itself is an A or a B.



So on level 4, you need to pay attention to the stars around the letter, the letter itself, and the frame color. As with the frame color and context letter, you will not see the stars on every trial- they will appear about half as often as the context letter. So you need to remember which orientation you saw the stars at most recently (horizontal or vertical around the letter) in order to know what the current letter/frame combination means you should do. For instance, if there are horizontal stars and the letter ‘A’, and the frame flashes purple, you would do whatever rule purple indicates in the default context (which is the same as ‘A’). Then, in the next trial, if you don’t get any cues, you will keep using the same rule. If the letter B appears in the following trial (but the stars don’t appear), you’re still in the default context, and you should keep using the same color-rule pairing as before (even though ‘B’ corresponds to the other color-rule pairing).

Error screens look pretty much like the instruction screens. They’re there to remind you what rules you’re supposed to be following, and then after you correct your mistake, the rules will disappear and you’ll move on to the next trial, like normal.

For each of the three tasks, you will do all the levels, in a mountain structure. So, you’ll start with level 1 of the dot moving task, then level 2, then 3, then 4, then back down to 3, 2, 1. Then, you will move on to the odd one out task and do the same thing (1-2-3-4-3-2-1), and then you’ll do the number judgment task the same way (1-2-3-4-3-2-1).

Accuracy is important, and you can earn money every block for high accuracy: >95% on level 1, >85% on level 2, and >75% on levels 3 and 4.

This part of the experiment should take about 45 minutes. Try to stay focused the whole time you are working. If you need to take a short break, please do so at one of the pause screens (i.e., an instructions screen) and **not** in the middle of a trial.

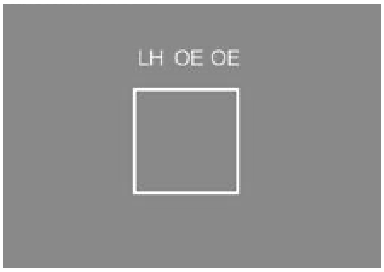
[Have participant come out into the hallway and stretch/shake it out before you begin giving instruction for the next part]

3. Serial-Order Format

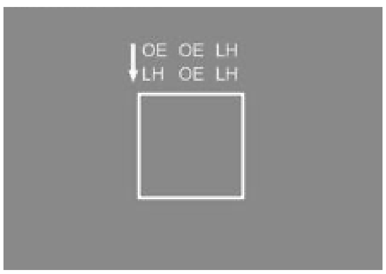
Now you are going to do the same 3 tasks, but the 4 ‘levels’ will be different.

Level 1 will look exactly like it did before: You will be given a single rule at the beginning of the block, and then you will use that rule until you reach the next block and get a new rule.

On the second level, you will be given a sequence of 3 rules at the beginning of the block, and you will use one rule per trial. For example, if the rules were LH-OE-OE, you’d have to indicate whether the number is low or high in the first trial, odd or even in the second trial, odd or even again in the third trial, and then you’ll loop back to the beginning and use the low/high rule in the fourth trial.



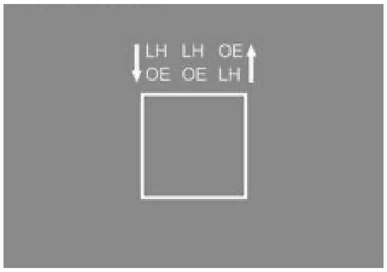
On the third level, you will be given a sequence of 6 rules at the beginning of the block, and you will go through them (1 rule per trial) like in level 2. Here's what the instruction screen for Level 3 looks like [a downward arrow on the left, and LH OE OE, OE LH OE]. So in Level 3, you'll go through the sequence like this: (point) trial 1, 2, 3, 4, 5, 6, and then you'll loop back to the beginning and do the first rule again for trial 7.



On the fourth level, you will be given a sequence of 6 rules at the beginning of the block, and you will begin going through them (1 rule per trial) like in level 3. However, instead of simply looping through the rules, you will go down and back up the set of chunks. This means you will do chunk 1, then chunk 2, and then (reversing the chunk order) chunk 2, then chunk 1.

The instruction screen for level 4 is similar to the one for level 3. The only difference is that for level 4, there are two arrows on the side of the sequence: one going down and the other going up.

In Level 4, instead of looping back through, you go down and back, like this: [point] trial 1, 2, 3, 4, 5, 6, and then back up (always starting from the left), 7, 8, 9, 10, 11, 12. And then for trial 13, you'll start back at the beginning.

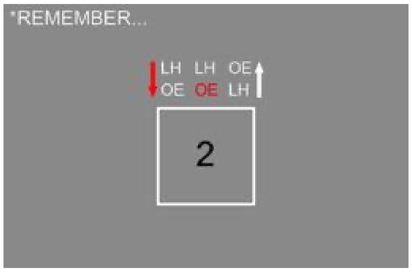


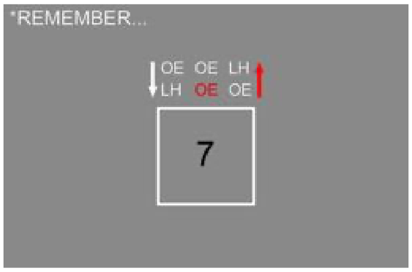
[use pictures or white board drawings to give examples above and explain error screen(s) below]

When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence.

For example, if the downward arrow is red, that indicates that you were going down the sequence. So you did trial 1, 2, 3, 4, and then made a mistake on trial 5. So you'll need to correct your mistake (do trial 5 correctly) before moving on to the next trial.

Conversely, if the upward arrow is red, that means you were going back up the sequence. So you did trial 1, 2, 3, 4, 5, 6 (down), and then going back up, you did trial 7 and then made a mistake on trial 8. Again, you'll need to correct your mistake before continuing through the sequence





As with the set of tasks you just did, you will do all the levels of each task in a mountain structure (1-2-3-4-3-2-1).

Accuracy is important, and the same accuracy-based earnings apply to this set of tasks: You need >95% accuracy on level 1, >85% on level 2, and >75% on levels 3 and 4.

This part of the experiment should take about an hour. Try to stay focused the whole time you are working. If you need to take a short break, please do so at one of the pause screens (i.e., an instructions screen).

4. Cued Switching (second half)

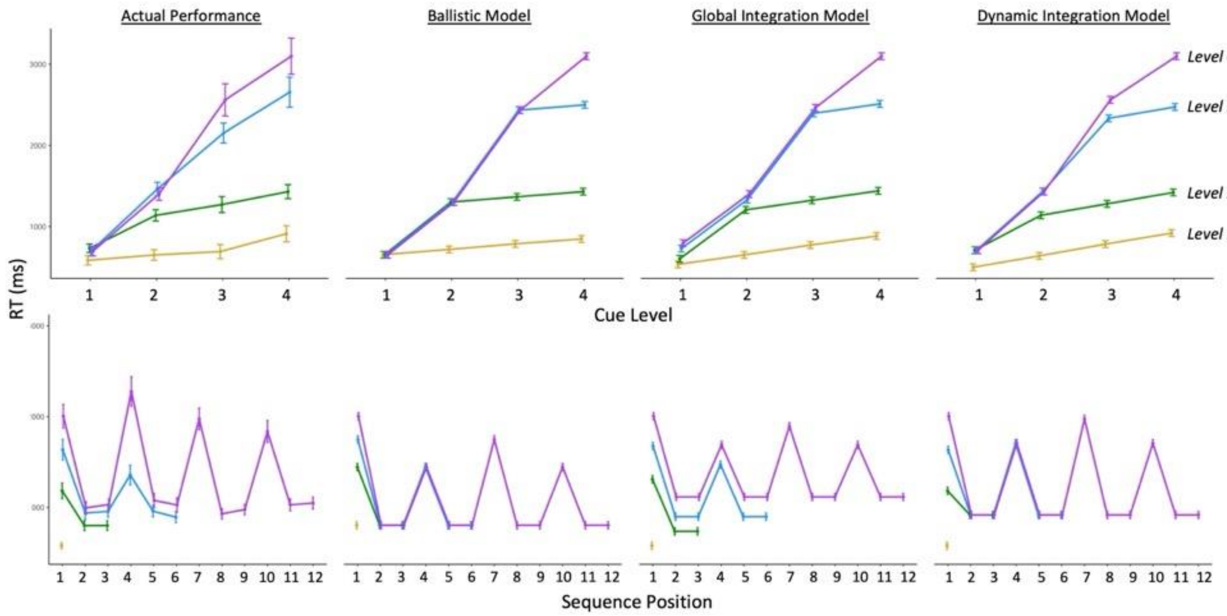
This is going to be exactly the same as the very first thing you did in this session. Now, as it was before, the rule you are supposed to use will appear in the middle of the screen on every trial, and it will stay on the screen until you respond.

For every block that you have >95% accuracy, you will earn additional money. This part of the experiment won't take you more than 5 minutes.

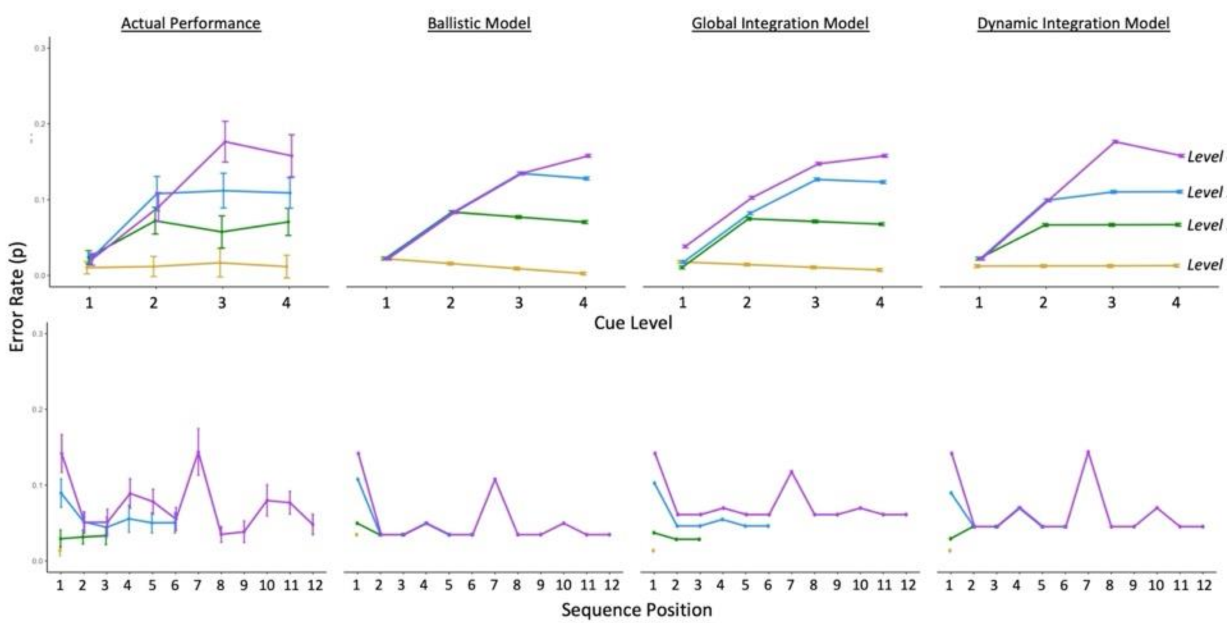
APPENDIX B

SUPPLEMENTAL REGRESSION MODELS FOR CHAPTER 3

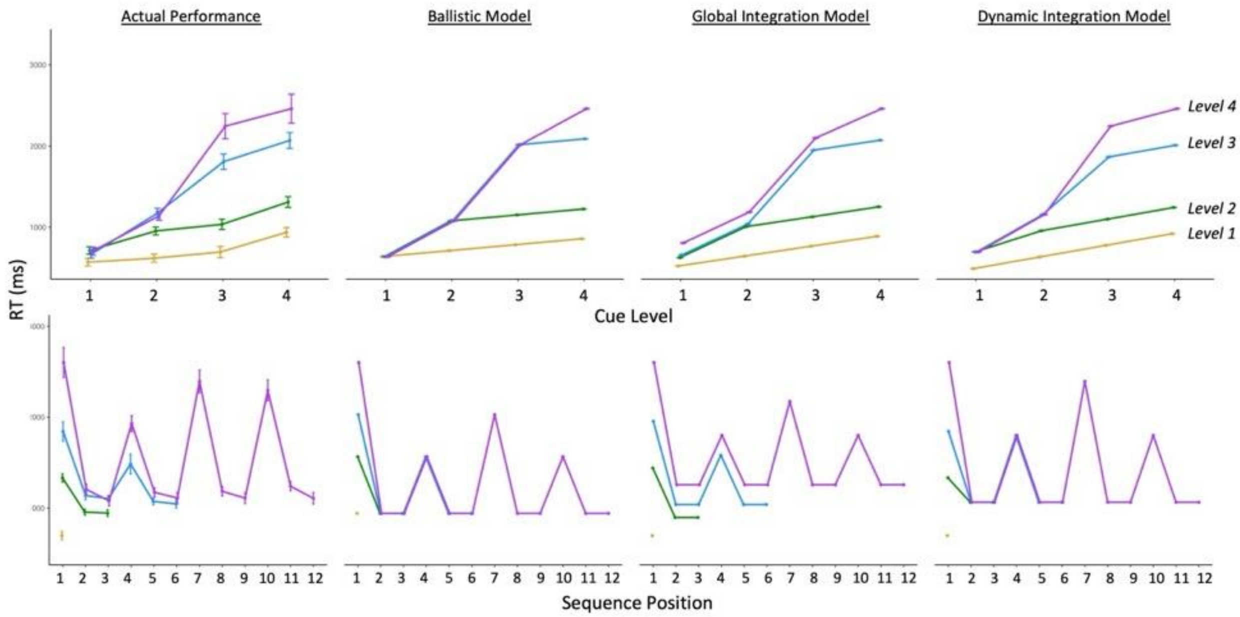
B1. Performance results and model fits with RT as the dependent variable, for Experiment 1 in cue-based (top panels) and serial-order (bottom panels) formats. Error bars indicate 95% within-subject confidence intervals.



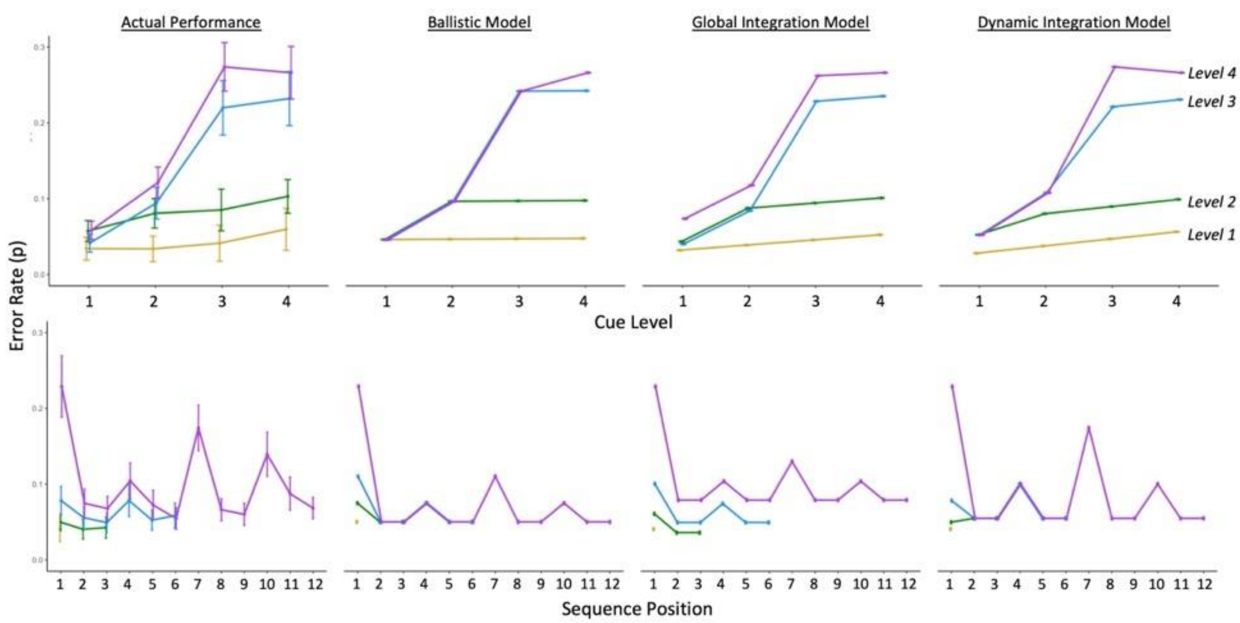
B2. Performance results and model fits with error rate as the dependent variable, for Experiment 1 in cue-based (top panels) and serial-order (bottom panels) formats. Error bars indicate 95% within-subject confidence intervals.



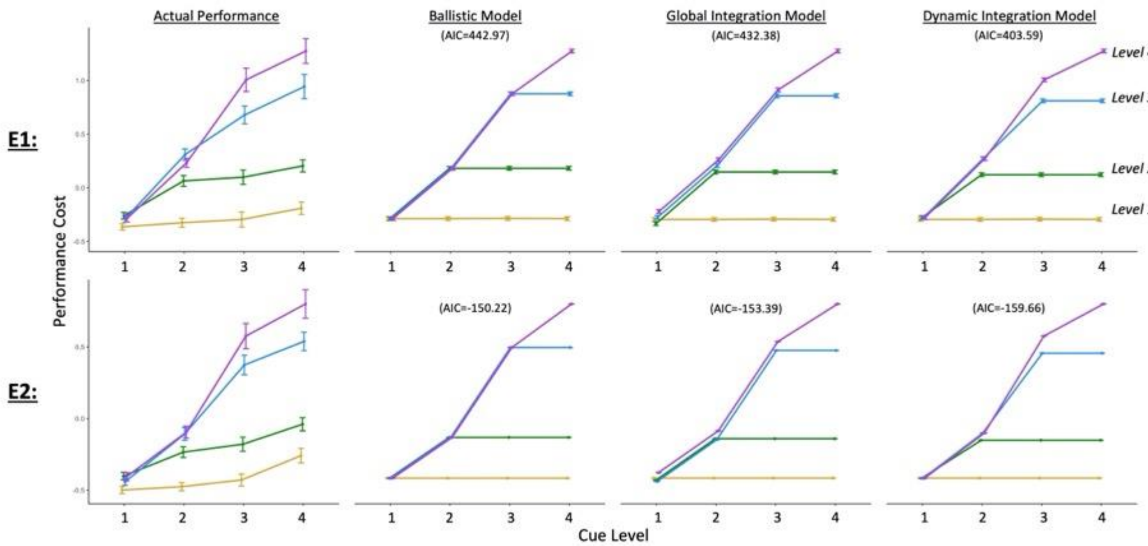
B3. Performance results and model fits with RT as the dependent variable, for Experiment 2 in cue-based (top panels) and serial-order (bottom panels) formats. Error bars indicate 95% within-subject confidence intervals.



B4. Performance results and model fits with error rate as the dependent variable, for Experiment 2 in cue-based (top panels) and serial-order (bottom panels) formats. Error bars indicate 95% within-subject confidence intervals.

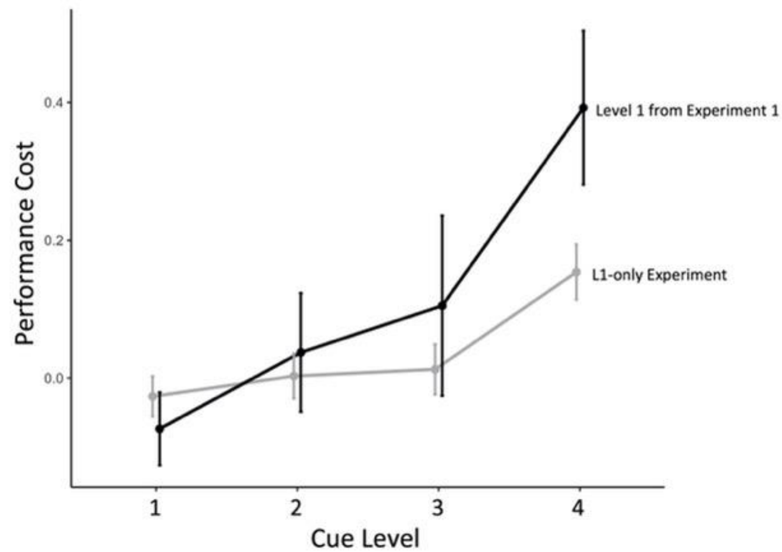


B5. Performance results and model fits in the cue-based format, for Experiment 1 (top panels) and Experiment 2 (bottom panels) with the performance cost measure as dependent variable. Models were constructed without including the filtering variable. We include these analyses here because the preregistration for Experiment 2 had been submitted before this variable was added to the analysis of cue-based models in Experiment 1. Error bars indicate 95% within-subject confidence intervals.



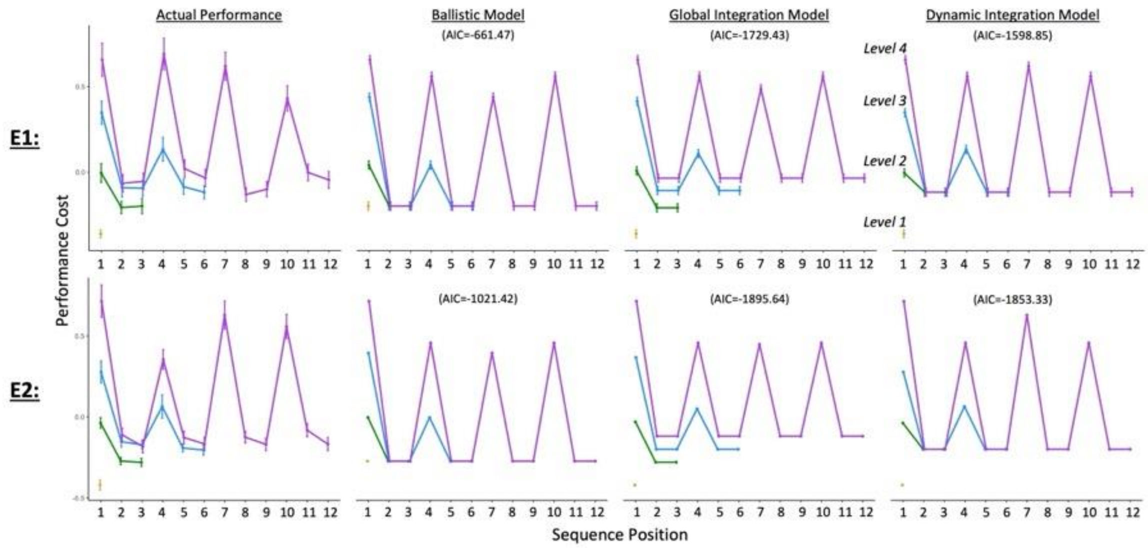
Models that included the filtering variable (see Fig. 3 and 4) produced better fits than the corresponding models without the filtering variable (shown here). We also tested each of the models in S6 against the corresponding models that included the filtering variable. In Experiment 1, the ballistic ($X^2(1)=4.48$, $p=.03$), global integration ($X^2(1)=19.54$, $p<.001$), and dynamic integration ($X^2(1)=35.82$, $p<.001$) models all provided significantly better fit when the filtering variable was included. The filtering models also showed significantly better fit in Experiment 2 (ballistic $X^2(1)=31.19$, $p<.001$; global integration $X^2(1)=59.84$, $p<.001$; dynamic integration $X^2=76.36$, $p<.001$).

B6. Performance at each cue level in each task, for Experiment 1 (at structural level 1 only), versus the experiment in which all cue levels were shown in a level 1 structure. Error bars indicate 95% within-subject confidence intervals.



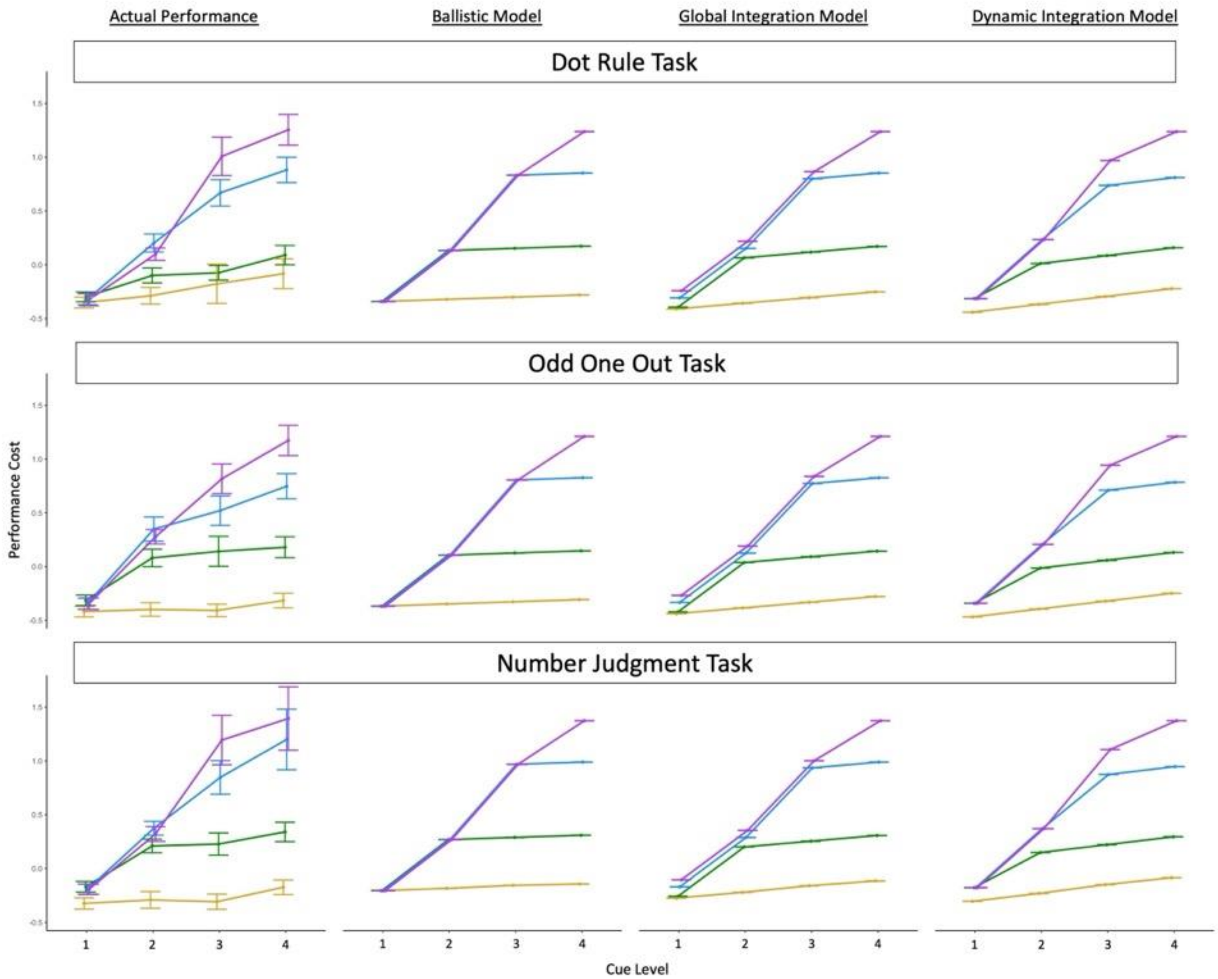
For a statistical test of the “pure” filtering effects, we used a linear contrast for the cue-level factor to predict performance costs in the control experiment (L1-only). This linear contrast was highly reliable, t=9.39, p<.001, suggesting perceptual/attentional costs even in the complete absence of hierarchical control demands. In a second step, we included the data from the corresponding level 1 structural condition in Exp. 1 (Level 1 from Experiment 1). A robust interaction between experiment and the linear cue-level contrast emerged, t=4.78, p<.001, indicating that the filtering effect was amplified in the hierarchical situation. We interpret this increased filtering effect in the hierarchical context as a result of proactive interference from the instructed and/or in previous blocks experienced, hierarchical control structure. Inspection of the figure suggests that the filtering-cost pattern might be driven mainly by the contrast between level 3 and level 4 cues. However, when we repeated the above analysis after excluding level 4 data, we obtained a qualitatively similar, though somewhat muted pattern as for the full data set (control experiment only: t=2.25, p<.05, interaction between experiments: t=2.39, p<.05).

B7. Results and model fits in the serial-order format, for Experiment 1 (top panels) and Experiment 2 (bottom panels). Models were constructed with an additional chunk repetition variable included. For both experiments, this model produced better fit than the original model. Error bars indicate 95% within-subject confidence intervals.

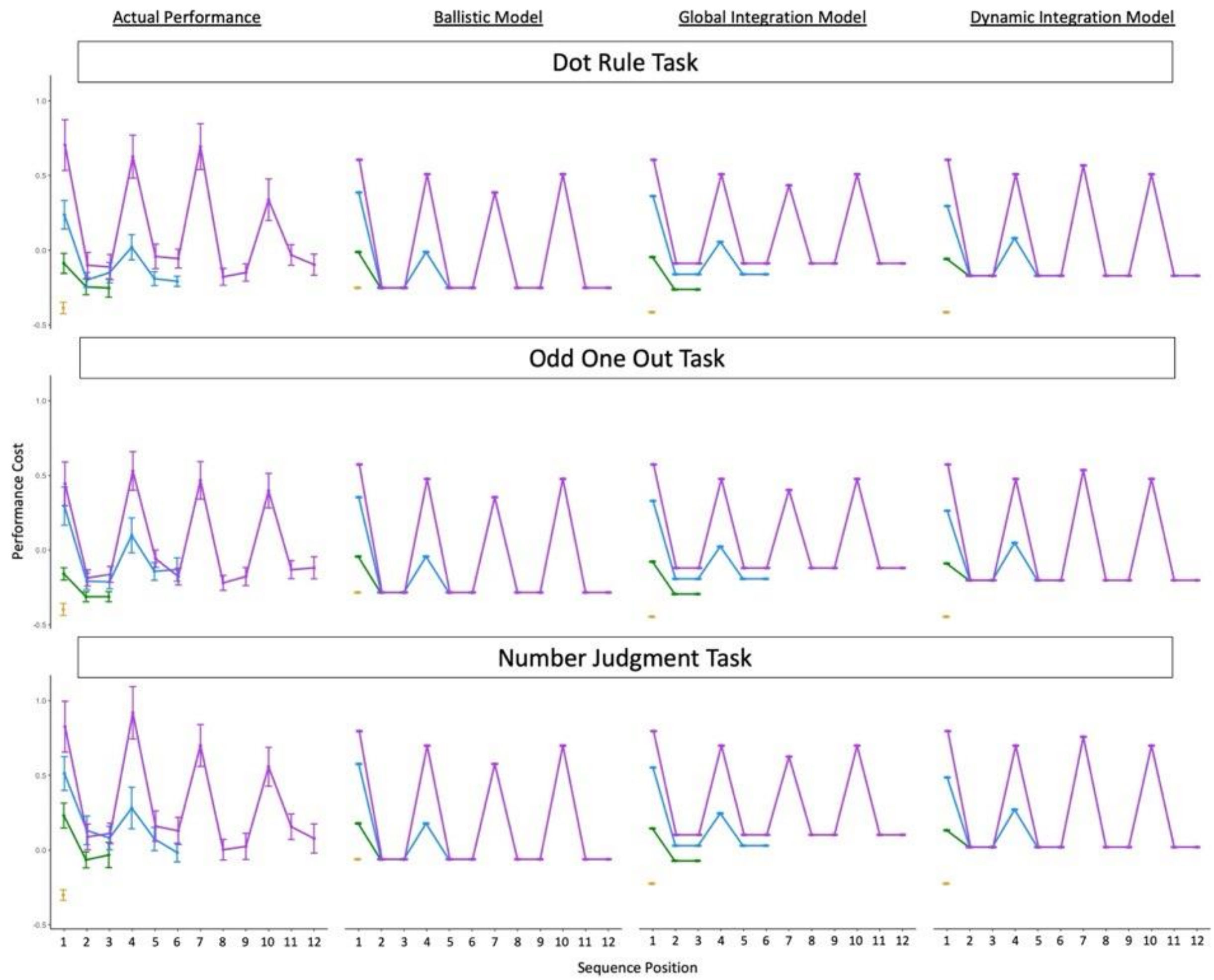


In addition to the clear differences in AIC between models that included the chunk repetition variable and those that did not, we tested each of the models presented in Figures 3 and 4 against the corresponding chunk repetition models in S8. In Experiment 1, the ballistic ($X^2(1)=554.24$, $p<.001$), global integration ($X^2(1)=335.62$, $p<.001$), and dynamic integration ($X^2(1)=344.00$, $p<.001$) models all provided significantly better fit when the chunk repetition variable was included. The chunk repetition models also showed significantly better fit in Experiment 2 (ballistic $X^2(1)=425.58$, $p<.001$; global integration $X^2(1)=249.65$, $p<.001$; dynamic integration $X^2(1)=296.51$, $p<.001$).

B8. Performance results and model fits in the cue-based format, separately for each task in Experiment 1. Error bars indicate 95% within-subject confidence intervals.



B9. Performance results and model fits in the serial-order format, separately for each task in Experiment 1. Error bars indicate 95% within-subject confidence intervals.

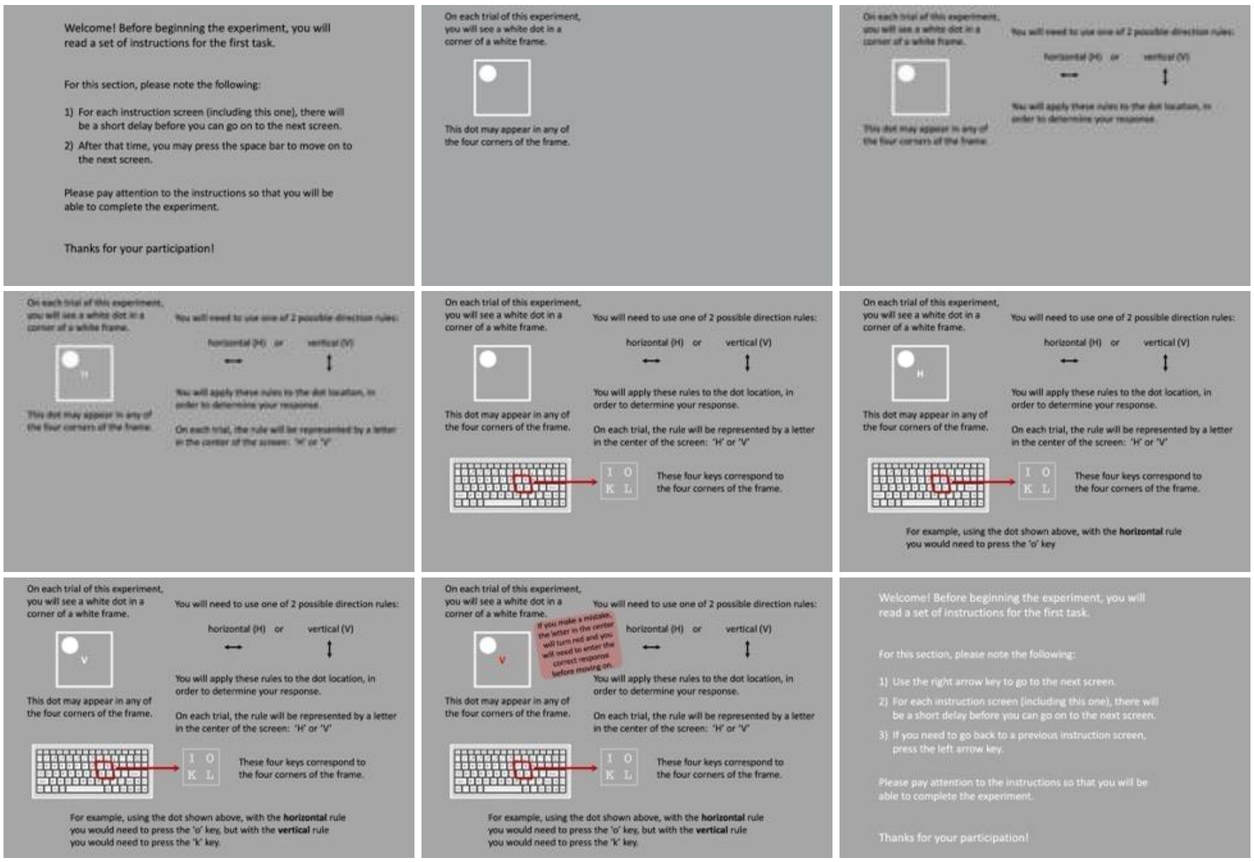


APPENDIX C

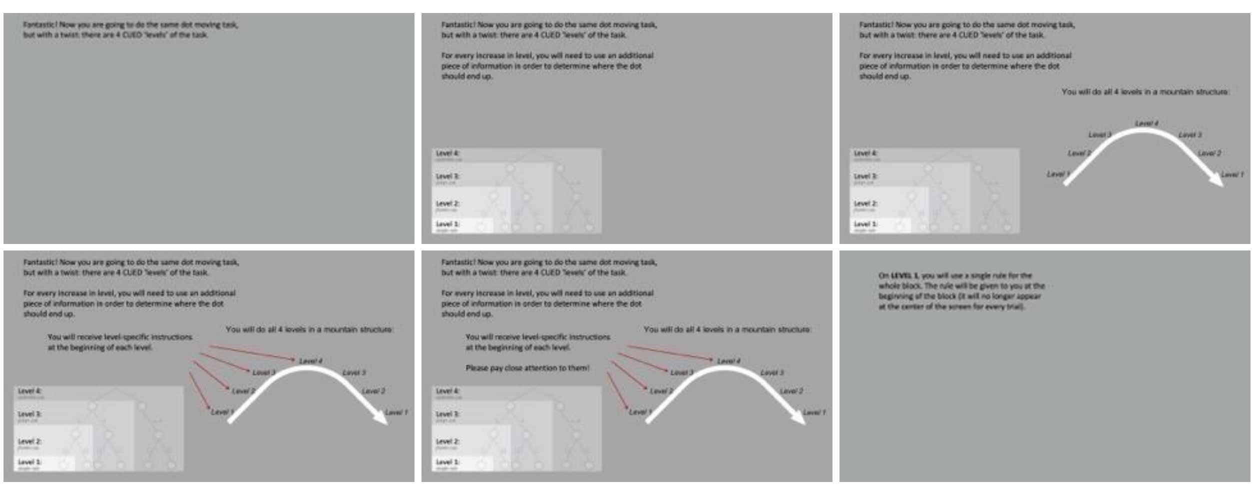
INSTRUCTION SLIDES USED IN EXPERIMENT 2 OF CHAPTER 3

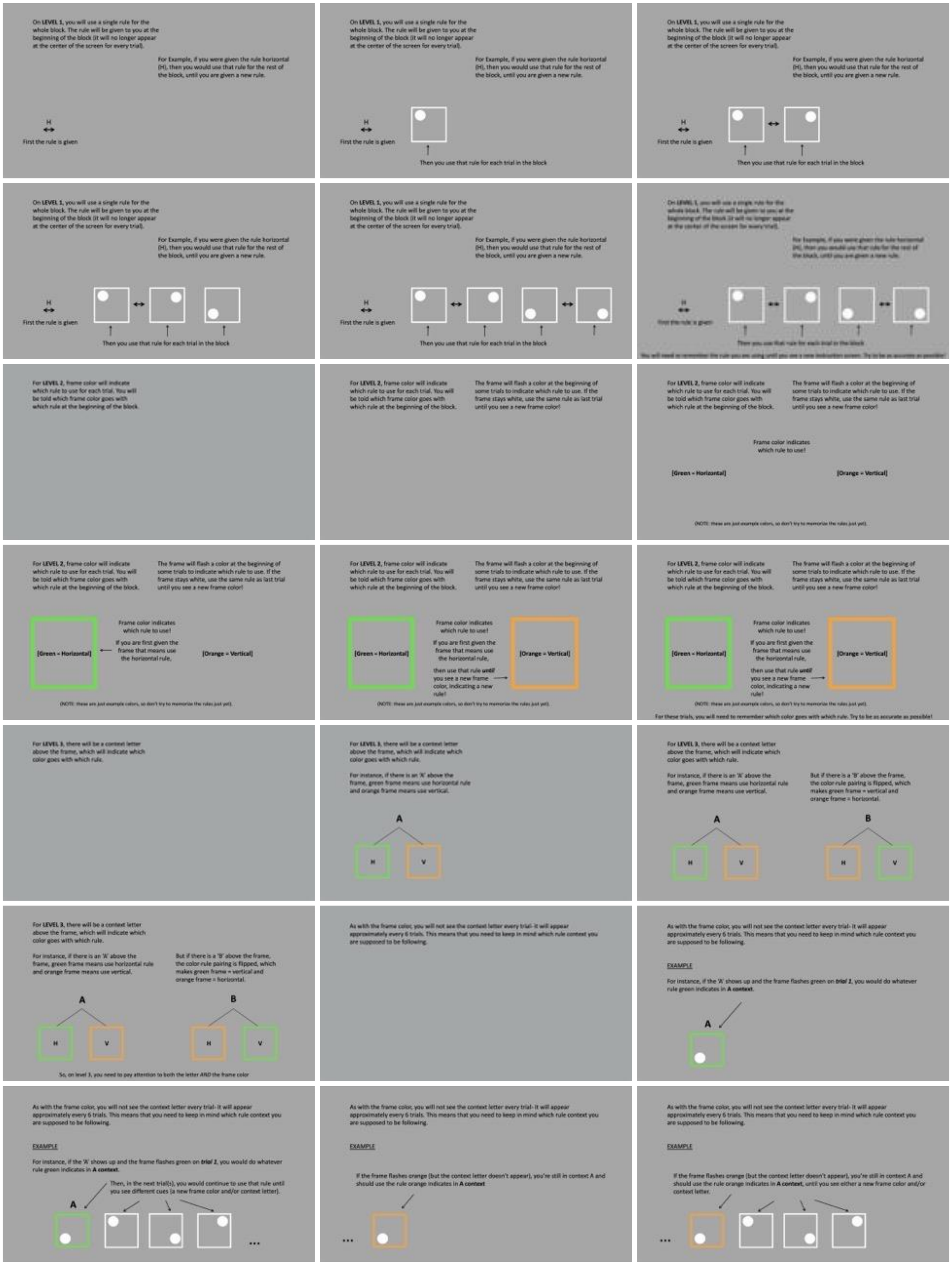
Instruction slides used in Experiment 2. Each instruction screen was self-paced, and participants had to click through them in order, to learn the tasks.

Beginning of experiment instruction slides:



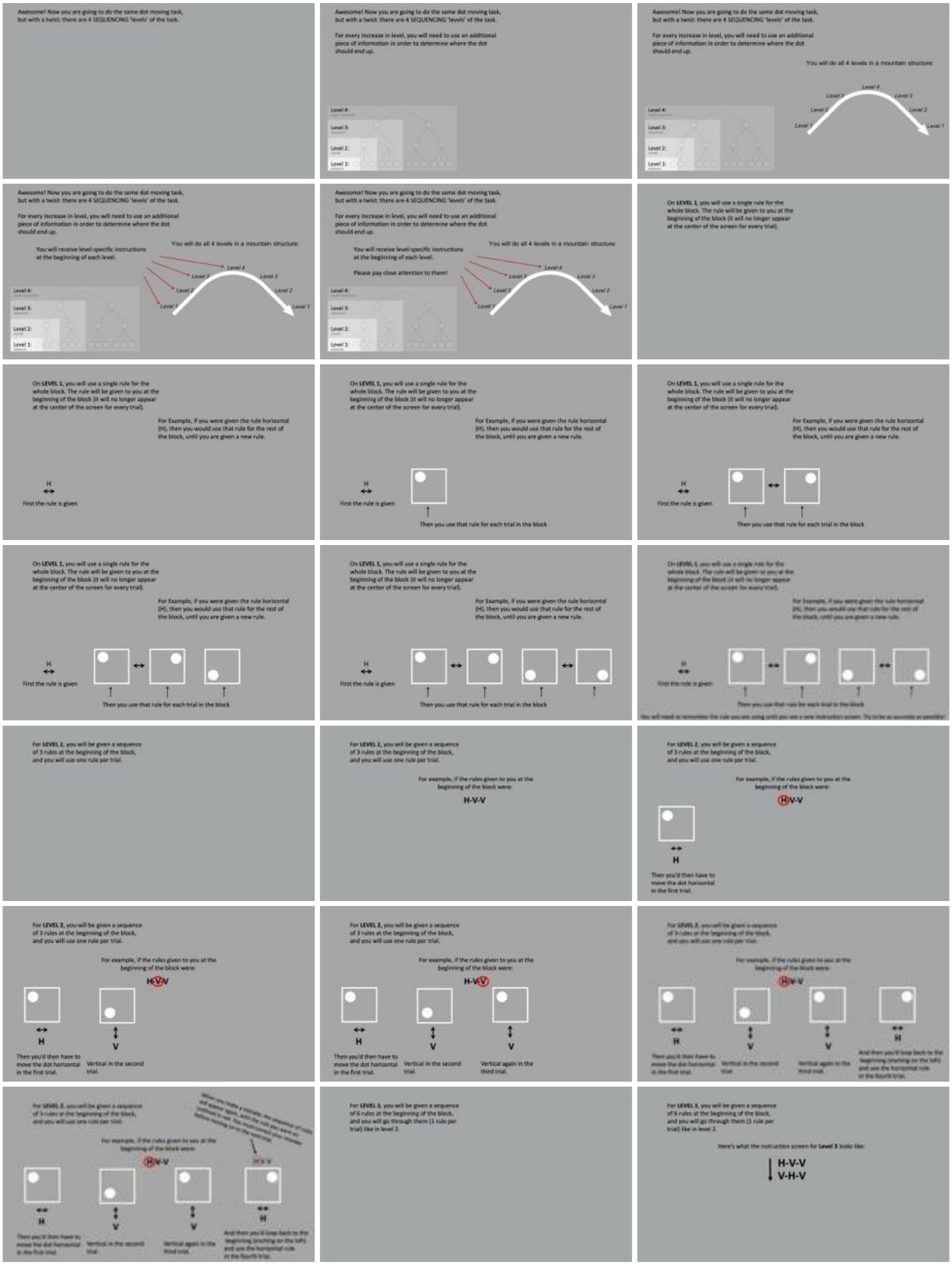
Instruction slides for cue-based format:

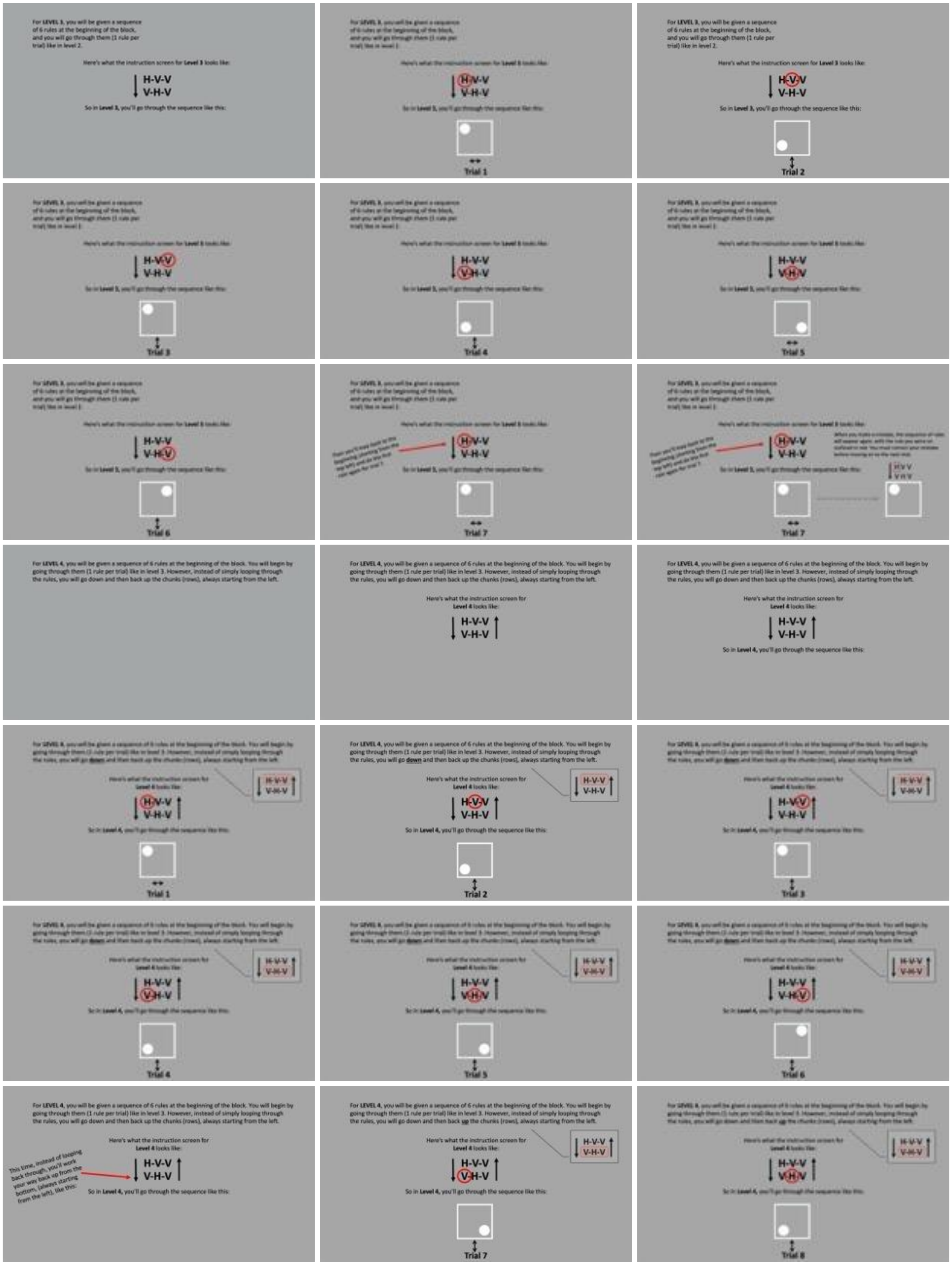


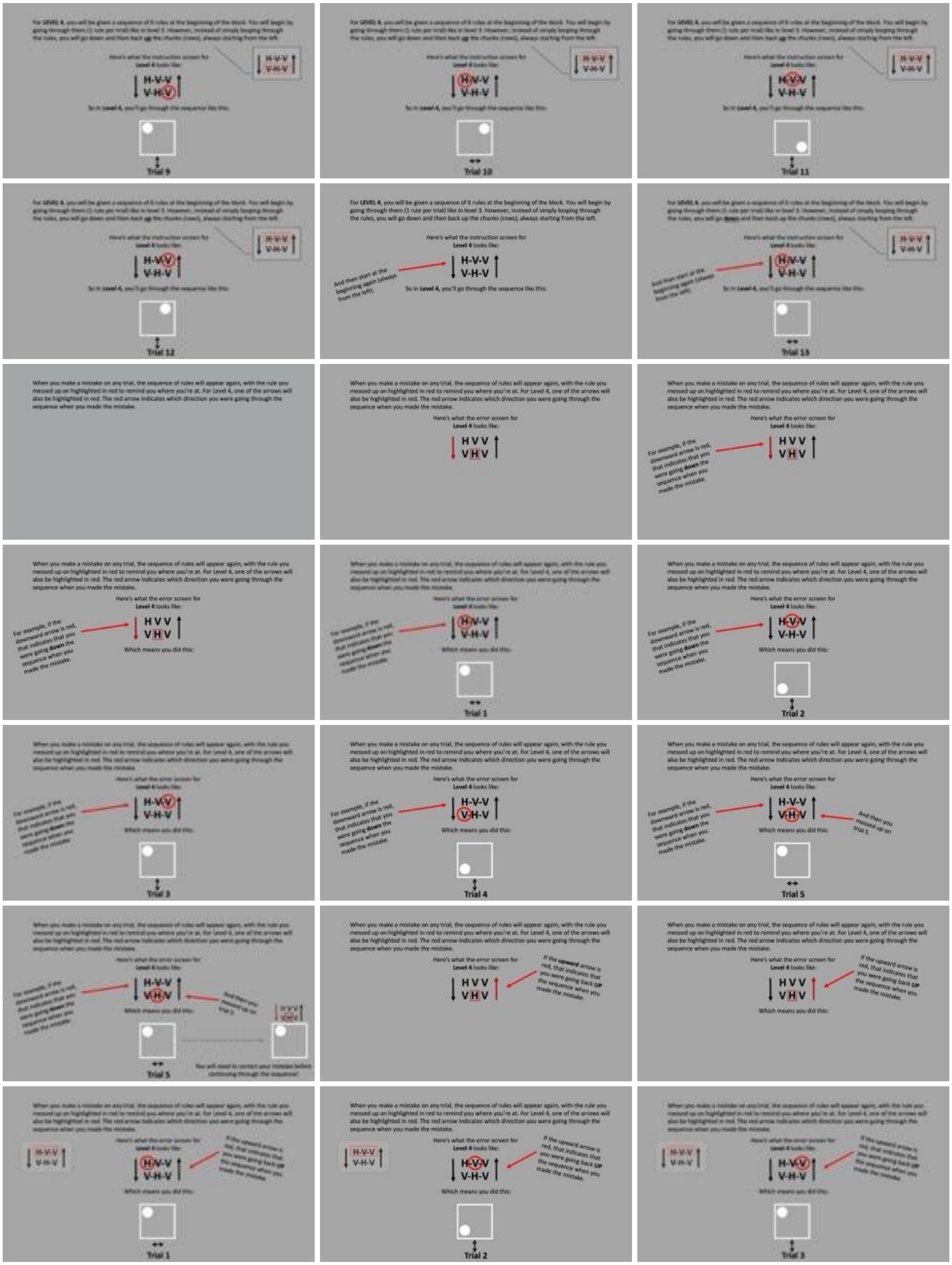

On LEVEL 1, you will use a single rule for the whole block. The rule will be given to you at the beginning of the block (it will no longer appear at the center of the screen for every trial).
For Example, if you were given the rule horizontal (H), then you would use that rule for the rest of the block, until you are given a new rule.
H
First the rule is given
Then you use that rule for each trial in the block
For LEVEL 2, frame color will indicate which rule to use for each trial. You will be told which frame color goes with which rule at the beginning of the block.
The frame will flash a color at the beginning of some trials to indicate which rule to use. If the frame stays white, use the same rule as last trial until you see a new frame color!
Frame color indicates which rule to use!
[Green = Horizontal]
[Orange = Vertical]
If you are first given the frame that means use the horizontal rule,
then use that rule until you see a new frame color, indicating a new rule!
(NOTE: these are just example colors, so don't try to memorize the rules just yet).
For these trials, you will need to remember which color goes with which rule. Try to be as accurate as possible!
For LEVEL 3, there will be a context letter above the frame, which will indicate which color goes with which rule.
For instance, if there is an 'A' above the frame, green frame means use horizontal rule and orange frame means use vertical.
But if there is a 'B' above the frame, the color-rule pairing is flipped, which makes green frame = vertical and orange frame = horizontal.
A
B
H
V
So, on level 3, you need to pay attention to both the letter AND the frame color
As with the frame color, you will not see the context letter every trial- it will appear approximately every 6 trials. This means that you need to keep in mind which rule context you are supposed to be following.
EXAMPLE
For instance, if the 'A' shows up and the frame flashes green on trial 1, you would do whatever rule green indicates in A context.
Then, in the next trial(s), you would continue to use that rule until you see different cues (a new frame color and/or context letter).
If the frame flashes orange (but the context letter doesn't appear), you're still in context A and should use the rule orange indicates in A context, until you see either a new frame color and/or context letter.
...

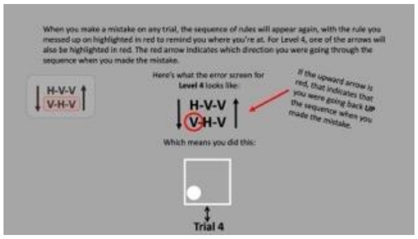

On LEVEL 4, you will see stars around the context letter.
If the stars are above and below the letter (vertical), you will do the same thing you did on LEVEL 3, with letter indicating which frame color goes with which rule.
A
OR
B
H
V
H
V
However, if the stars are on either side of the letter (horizontal), you will use one frame-rule pairing context (default context), regardless of what letter appears above the frame.
* A * or * B *
Think back to the previous example…
On LEVEL 4, the default context might be the same as context 'A.'
This means that if the stars are horizontally oriented, you'd use the 'A' context, regardless of whether the letter itself is an A or a B.
Default context is the same!
As with the frame color and context letter, you will not see the stars on every trial. They will appear about half as often as the context letter. This means you need to remember which orientation you saw the stars in most recently (horizontal or vertical around the letter) so you know whether you are operating in the default context or not.
EXAMPLE
For instance, if you see horizontal stars on trial 1, you would do whatever rule green indicates in the default context, regardless of context letter.
Then, in the next trial(s), you would continue to use that rule until you see different cues (new stars and/or frame color and/or context letter).
If context letter B appears in the following trial (but the stars don't appear), you're still in the default context
If context letter B appears in the following trial (but the stars don't appear), you're still in the default context, and you should keep using the same color-rule pairing as before (even though 'B' would correspond to the other color-rule pairing, if the stars were vertical).
…
QUICK RECAP OF CUED LEVELS:
horizontal
vertical
Level 1
Level 2
Level 3
Level 4
You will do all 4 levels in a mountain structure:
Try to be as accurate as possible!

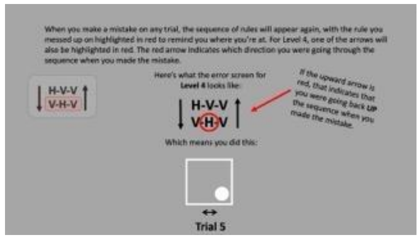
Instruction slides for serial-order format:

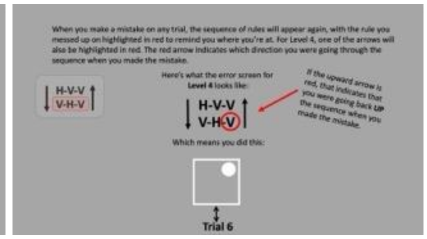


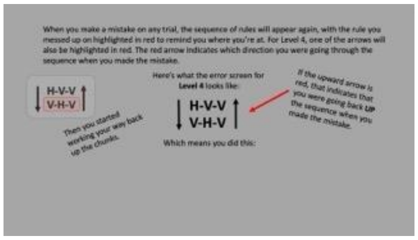

For LEVEL 3, you will be given a sequence of 6 rules at the beginning of the block, and you will go through them (1 rule per trial) like in level 2.
Here's what the instruction screen for Level 3 looks like:
H-V-V
V-H-V
So in Level 3, you'll go through the sequence like this:
Trial 1
Trial 2
Trial 3
Trial 4
Trial 5
Trial 6
Trial 7
For LEVEL 4, you will be given a sequence of 6 rules at the beginning of the block. You will begin by going through them (1 rule per trial) like in level 3. However, instead of simply looping through the rules, you will go down and then back up the chunks (rows), always starting from the left.
Here's what the instruction screen for Level 4 looks like:
H-V-V
V-H-V
So in Level 4, you'll go through the sequence like this:
Trial 1
Trial 2
Trial 3
Trial 4
Trial 5
Trial 6
Trial 7
Trial 8

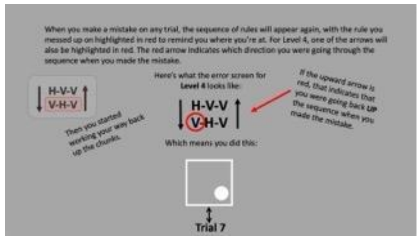


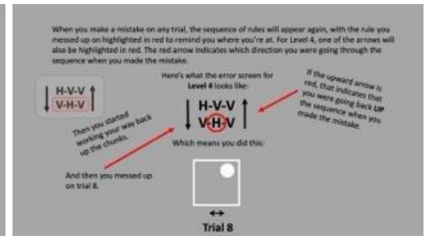

When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence when you made the mistake.
H-V-V
V-H-V
Here's what the error screen for **Level 4** looks like:
H-V-V
V-H-V
If the upward arrow is red, that indicates that you were going back **UP** the sequence when you made the mistake.
Which means you did this:
Trial 4

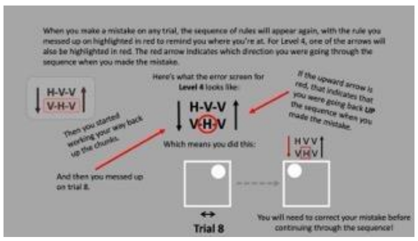

When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence when you made the mistake.
H-V-V
V-H-V
Here's what the error screen for **Level 4** looks like:
H-V-V
V-H-V
If the upward arrow is red, that indicates that you were going back **UP** the sequence when you made the mistake.
Which means you did this:
Trial 5


When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence when you made the mistake.
H-V-V
V-H-V
Here's what the error screen for **Level 4** looks like:
H-V-V
V-H-V
If the upward arrow is red, that indicates that you were going back **UP** the sequence when you made the mistake.
Which means you did this:
Trial 6


When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence when you made the mistake.
H-V-V
V-H-V
Here's what the error screen for **Level 4** looks like:
H-V-V
V-H-V
If the upward arrow is red, that indicates that you were going back **UP** the sequence when you made the mistake.
Then you started working your way back up the chunks.
Which means you did this:


When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence when you made the mistake.
H-V-V
V-H-V
Here's what the error screen for **Level 4** looks like:
H-V-V
V-H-V
If the upward arrow is red, that indicates that you were going back **UP** the sequence when you made the mistake.
Then you started working your way back up the chunks.
Which means you did this:
Trial 7


When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence when you made the mistake.
H-V-V
V-H-V
Here's what the error screen for **Level 4** looks like:
H-V-V
V-H-V
If the upward arrow is red, that indicates that you were going back **UP** the sequence when you made the mistake.
Then you started working your way back up the chunks.
And then you messed up on trial 8.
Which means you did this:
Trial 8


When you make a mistake on any trial, the sequence of rules will appear again, with the rule you messed up on highlighted in red to remind you where you're at. For Level 4, one of the arrows will also be highlighted in red. The red arrow indicates which direction you were going through the sequence when you made the mistake.
H-V-V
V-H-V
Here's what the error screen for **Level 4** looks like:
H-V-V
V-H-V
If the upward arrow is red, that indicates that you were going back **UP** the sequence when you made the mistake.
Then you started working your way back up the chunks.
And then you messed up on trial 8.
Which means you did this:
H V V
V H V
Trial 8
You will need to correct your mistake before continuing through the sequence!

APPENDIX D

CORRELATIONS OF MEASURED VARIABLES IN CHAPTER 4

Table D1. Correlation matrix with all drift rate variables and secondary measured variables.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
1. c1 D	–																																											
2. c1 O	.43	–																																										
3. c1 N	.29	.33	–																																									
4. c2 D	.42	.44	.37	–																																								
5. c2 O	.21	.35	.35	.47	–																																							
6. c2 N	.22	.26	.38	.44	.51	–																																						
7. c3 D	.38	.34	.19	.43	.33	.33	–																																					
8. c3 O	.17	.30	.31	.39	.32	.29	.37	–																																				
9. c3 N	.14	.26	.32	.3	.29	.42	.33	.30	–																																			
10. c3i D	.28	.19	.26	.43	.29	.34	.37	.30	.24	–																																		
11. c3i O	.08	.11	.28	.27	.41	.33	.21	.32	.46	.42	–																																	
12. c3i N	.09	.20	.23	.33	.43	.45	.25	.20	.43	.26	.54	–																																
13. c4 D	.44	.41	.29	.48	.38	.39	.51	.43	.30	.46	.32	.22	–																															
14. c4 O	.30	.36	.25	.48	.49	.38	.31	.33	.27	.31	.35	.30	.51	–																														
15. c4 N	.18	.30	.31	.36	.36	.50	.33	.24	.46	.30	.38	.48	.38	.40	–																													
16. c4i D	.09	.04	.13	.26	.19	.25	.20	.11	.28	.55	.52	.35	.33	.17	.26	–																												
17. c4i O	.11	.14	.18	.28	.24	.33	.30	.24	.40	.51	.67	.49	.33	.36	.37	.57	–																											
18. c4i N	.02	.06	.18	.20	.23	.30	.19	.10	.26	.41	.45	.44	.17	.18	.50	.51	.45	–																										
19. s1 D	.40	.40	.39	.45	.24	.20	.32	.37	.17	.19	.06	.14	.35	.20	.20	.06	.12	.07	–																									
20. s1 O	.44	.34	.27	.29	.19	.13	.30	.38	.18	.15	.05	.15	.35	.26	.21	.10	.12	.07	.46	–																								
21. s1 N	.27	.24	.41	.42	.27	.22	.30	.44	.34	.28	.31	.17	.35	.33	.33	.14	.29	.19	.31	.41	–																							
22. s2 D	.42	.39	.33	.52	.37	.37	.39	.39	.36	.27	.22	.25	.43	.33	.32	.17	.15	.19	.46	.43	.39	–																						
23. s2 O	.35	.35	.43	.43	.4	.33	.37	.38	.31	.27	.25	.27	.34	.36	.34	.09	.19	.16	.36	.35	.37	.47	–																					
24. s2 N	.11	.30	.36	.37	.50	.44	.20	.35	.32	.26	.33	.34	.29	.32	.35	.19	.25	.25	.26	.28	.38	.40	.33	–																				
25. s3 D	.37	.31	.24	.32	.23	.30	.37	.26	.27	.31	.15	.20	.40	.20	.20	.18	.23	.15	.40	.31	.30	.58	.37	.30	–																			
26. s3 O	.28	.30	.37	.36	.27	.27	.32	.25	.30	.34	.29	.27	.30	.29	.35	.28	.32	.34	.39	.31	.31	.39	.35	.32	.35	–																		
27. s3 N	.15	.33	.29	.36	.34	.41	.29	.23	.36	.30	.32	.39	.30	.31	.43	.27	.36	.37	.24	.20	.34	.32	.38	.50	.37	.39	–																	
28. s3i D	.41	.34	.26	.46	.28	.3	.37	.26	.28	.27	.23	.26	.42	.33	.33	.19	.23	.20	.45	.25	.30	.59	.34	.34	.59	.40	.45	–																
29. s3i O	.18	.28	.21	.29	.29	.26	.28	.16	.16	.34	.21	.32	.27	.24	.42	.23	.23	.37	.26	.23	.23	.33	.42	.30	.32	.45	.39	.41	–															
30. s3i N	.09	.29	.24	.28	.32	.31	.23	.22	.34	.20	.25	.32	.23	.20	.37	.13	.19	.27	.19	.17	.29	.30	.28	.46	.29	.26	.66	.36	.39	–														
31. s4 D	.32	.31	.26	.39	.33	.34	.37	.28	.32	.43	.29	.32	.40	.30	.35	.42	.37	.36	.27	.25	.25	.56	.36	.33	.56	.50	.51	.64	.42	.42	–													
32. s4 O	.24	.33	.28	.38	.38	.34	.35	.25	.33	.52	.37	.35	.41	.37	.45	.44	.44	.45	.20	.25	.28	.44	.41	.37	.37	.55	.54	.49	.55	.36	.71	–												
33. s4 N	.16	.26	.22	.27	.41	.41	.33	.16	.43	.42	.42	.45	.29	.26	.49	.41	.46	.47	.06	.10	.25	.30	.30	.41	.31	.43	.64	.42	.44	.52	.61	.68	–											
34. s4i D	.33	.30	.31	.36	.38	.31	.31	.18	.28	.47	.32	.41	.41	.28	.36	.43	.48	.40	.15	.19	.26	.33	.30	.31	.35	.39	.42	.41	.41	.35	.68	.64	.63	–										
35. s4i O	.21	.21	.29	.29	.36	.26	.24	.21	.26	.49	.42	.40	.35	.32	.33	.42	.46	.44	.11	.20	.25	.30	.35	.32	.28	.41	.42	.41	.52	.35	.56	.68	.58	.73	–									
36. s4i N	.27	.32	.23	.32	.42	.32	.20	.13	.27	.38	.31	.40	.32	.25	.35	.28	.32	.36	.09	.10	.22	.25	.24	.34	.31	.33	.45	.39	.33	.43	.48	.49	.64	.60	.58	–								
37. Color	.11	.19	.15	.18	.30	.17	.08	.16	.16	.20	.26	.28	.15	.22	.17	.10	.27	.20	.03	.07	.17	.12	.12	.21	.02	.23	.15	.11	.14	.13	.28	.32	.29	.38	.35	.27	–							
38. Orient	.17	.34	.24	.22	.25	.27	.25	.12	.22	.29	.29	.35	.22	.17	.32	.29	.29	.35	.17	.17	.18	.31	.15	.30	.19	.34	.27	.36	.26	.29	.37	.41	.44	.40	.38	.33	.50	–						
39. Space	.16	.19	.18	.20	.26	.08	.12	.02	.10	.23	.26	.27	.09	.12	.14	.25	.22	.32	.06	.05	.08	.14	.18	.23	.12	.26	.23	.22	.21	.19	.38	.40	.41	.35	.34	.28	.52	.49	–					
40. LS	.17	.14	.11	.24	.12	.06	.18	.14	.06	.30	.30	.25	.16	.11	.15	.31	.35	.24	.15	.05	.16	.16	.10	.17	.17	.16	.17	.28	.21	.14	.33	.29	.24	.37	.31	.27	.29	.20	.28	–				
41. NS	.26	.17	.24	.22	.11	.19	.14	.18	.15	.30	.27	.19	.19	.19	.22	.24	.36	.26	.16	.14	.27	.10	.16	.12	.16	.20	.25	.26	.33	.19	.28	.35	.26	.40	.38	.28	.16	.13	.22	.40	–			
42. Raven	.24	.17	.15	.26	.15	.18	.12	.14	.18	.34	.24	.25	.20	.19	.17	.14	.34	.19	.13	.06	.11	.15	.12	.24	.13	.23	.22	.27	.19	.11	.25	.38	.23	.33	.40	.29	.22	.23	.17	.28	.32	–		
43. PicSour	.13	.18	.13	.16	.12	.13	.04	.09	.10	.18	.08	.16	.05	.11	.15	.03	.16	.12	.16	.08	-.01	.06	-.02	.16	.08	.12	.21	.21	.08	.25	.20	.23	.26	.25	.17	.17	.15	.26	.15	.16	.14	.23	–	
44. PA	.04	.06	.20	.06	.09	-.01	.06	.11	.06	-.01	.20	.15	.03	.07	<.01	.11	.16	.02	.12	.11	.20	.07	.17	.07	.13	.13	.07	.09	.03	.04	.12	.09	.08	.15	.14	.02	.12	.14	.11	.24	.20	.08	.06	–
45. DFR	.20	.16	.34	.23	.15	.16	.14	.18	.24	.24	.24	.24	.08	.04	.14	.22	.31	.17	.25	.14	.18	.24	.23	.22	.18	.24	.21	.23	.12	.24	.37	.33	.26	.36	.29	.21	.21	.26	.29	.32	.32	.23	.23	.43

Note. c1-c4 = levels 1-4, in cue-based format; s1-4 = levels 1-4, in serial-order format. In the cue-based format, levels 3-4 include both non-integration (3, 4) and integration (3i, 4i) variables. In the serial-order format, levels 3-4 include both non-transition (3, 4) and transition (3i, 4i) variables. D = dot rule task; O = odd one out task; N = number judgment task; LS = Letter Sets; NS = Number Series; Raven = Raven Advanced Progressive Matrices (with the imputed values); Color = K estimate from color change detection task; Orient = K estimate from orientation change detection task; Space = K estimate from space capacity task; PicSour = picture source recognition task; PA = paired associates task; DFR = delayed free recall task. Correlations > .14 are significant at p < .05 level.

Table D2. Correlation matrix with all threshold separation variables and secondary measured variables.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
1. c1 D	–
2. c1 O	.43	–
3. c1 N	.31	.35	–
4. c2 D	.44	.46	.37	–
5. c2 O	.40	.40	.30	.64	–
6. c2 N	.26	.30	.29	.54	.55	–
7. c3 D	.35	.42	.24	.49	.47	.37	–
8. c3 O	.13	.34	.25	.40	.40	.37	.31	–
9. c3 N	.12	.22	.25	.28	.27	.49	.27	.32	–
10. c3i D	.33	.38	.35	.68	.57	.60	.41	.33	.38	–
11. c3i O	.32	.37	.31	.63	.65	.62	.43	.27	.30	.70	–
12. c3i N	.23	.31	.29	.51	.56	.69	.29	.33	.43	.62	.68	–
13. c4 D	.30	.35	.31	.63	.62	.61	.51	.42	.45	.67	.64	.56	–
14. c4 O	.28	.31	.29	.60	.66	.69	.38	.33	.41	.66	.70	.64	.73	–
15. c4 N	.15	.19	.21	.52	.58	.69	.31	.32	.50	.60	.65	.68	.63	.73	–
16. c4i D	.21	.29	.27	.62	.53	.62	.35	.32	.42	.72	.69	.67	.63	.63	.64	–
17. c4i O	.29	.27	.26	.52	.57	.57	.32	.32	.31	.63	.73	.61	.61	.65	.61	.64	–
18. c4i N	.08	.17	.27	.45	.42	.57	.17	.25	.44	.61	.66	.72	.51	.56	.67	.67	.64	–
19. s1 D	.32	.47	.35	.45	.38	.36	.31	.32	.14	.38	.39	.34	.36	.34	.17	.35	.31	.30	–
20. s1 O	.35	.33	.24	.32	.29	.26	.28	.30	.15	.29	.21	.15	.26	.20	.20	.22	.16	.13	.42	–
21. s1 N	.17	.25	.38	.24	.30	.31	.23	.33	.24	.29	.20	.25	.25	.26	.21	.24	.18	.18	.18	.30	–
22. s2 D	.46	.47	.34	.57	.53	.38	.40	.43	.28	.43	.49	.33	.46	.40	.38	.40	.44	.28	.47	.39	.28	–
23. s2 O	.32	.38	.35	.43	.53	.38	.35	.33	.31	.41	.49	.39	.45	.40	.41	.40	.43	.34	.37	.39	.25	.52	–
24. s2 N	.24	.24	.35	.42	.51	.49	.26	.33	.38	.45	.49	.47	.42	.45	.53	.44	.42	.44	.32	.26	.27	.48	.46	–
25. s3 D	.24	.33	.31	.34	.39	.40	.32	.36	.27	.36	.39	.33	.39	.34	.33	.33	.37	.31	.45	.22	.22	.50	.42	.39	–
26. s3 O	.28	.32	.34	.40	.49	.40	.29	.29	.24	.44	.46	.32	.37	.43	.37	.34	.40	.26	.34	.29	.20	.42	.49	.41	.37	–
27. s3 N	.18	.19	.25	.31	.31	.52	.26	.21	.39	.33	.36	.48	.36	.44	.49	.37	.27	.43	.31	.25	.24	.35	.38	.49	.42	.32	–
28. s3i D	.31	.32	.30	.41	.37	.42	.28	.32	.36	.39	.42	.42	.34	.39	.37	.34	.34	.36	.38	.23	.22	.51	.30	.38	.45	.35	.38	–
29. s3i O	.18	.26	.24	.37	.47	.42	.26	.32	.25	.38	.42	.40	.40	.43	.46	.29	.42	.33	.26	.23	.27	.35	.45	.42	.31	.47	.36	.33	–
30. s3i N	.12	.11	.24	.35	.41	.50	.19	.24	.46	.41	.48	.53	.47	.44	.57	.48	.43	.53	.23	.14	.24	.35	.36	.55	.31	.25	.50	.33	.35	–
31. s4 D	.25	.24	.28	.51	.50	.55	.31	.34	.43	.51	.56	.60	.53	.53	.62	.58	.55	.54	.31	.28	.16	.48	.47	.50	.49	.46	.51	.52	.45	.57	–
32. s4 O	.18	.18	.26	.45	.49	.43	.25	.33	.41	.42	.48	.42	.45	.45	.55	.35	.44	.42	.13	.21	.17	.40	.43	.46	.32	.47	.36	.39	.56	.44	.65	–
33. s4 N	.03	.03	.20	.34	.36	.51	.17	.29	.50	.42	.46	.58	.47	.56	.66	.50	.45	.61	.18	.11	.19	.31	.33	.56	.35	.31	.57	.40	.42	.67	.68	.59	–
34. s4i D	.13	.24	.14	.36	.44	.50	.23	.23	.37	.45	.50	.54	.43	.44	.55	.45	.47	.45	.24	.15	.15	.32	.36	.35	.33	.38	.31	.37	.44	.56	.61	.47	.55	–
35. s4i O	.18	.27	.26	.41	.42	.49	.30	.37	.35	.45	.44	.49	.41	.42	.48	.34	.38	.38	.18	.21	.27	.39	.43	.41	.31	.48	.34	.41	.50	.50	.56	.63	.54	.54	–
36. s4i N	.03	.03	.16	.19	.27	.38	.04	.29	.38	.29	.38	.50	.29	.38	.50	.38	.37	.52	.13	.15	.23	.24	.25	.40	.26	.17	.48	.26	.32	.63	.47	.41	.67	.46	.44	–
37. Color	-.12	-.21	-.10	-.04	.04	.03	-.01	-.06	.10	-.03	-.02	.06	.03	.06	.10	.05	.02	.15	-.08	-.03	-.09	-.09	-.04	.02	-.01	-.07	.09	<.01	-.12	.12	.11	.02	.19	.15	.01	.20	–
38. Orient	-.15	-.35	-.21	-.11	-.01	.07	-.12	-.01	.11	-.03	-.03	.15	.01	.10	.15	.08	.02	.15	-.18	-.14	-.09	-.26	-.13	-.06	-.13	-.23	.08	-.04	-.08	.09	.16	<.01	.19	.10	-.07	.22	.50	–
39. Space	-.14	-.20	-.05	.05	.07	.16	.04	.12	.17	.13	.09	.14	.09	.19	.19	.09	.09	.19	-.05	-.04	.07	-.02	-.05	.08	-.02	.01	.09	.02	-.01	.19	.15	.11	.32	.20	.11	.21	.52	.49	–
40. LS	-.15	-.16	-.06	-.01	.03	.11	-.07	-.10	.16	.13	.06	.09	.06	.11	.16	.12	-.04	.17	-.11	-.04	-.06	-.11	-.01	.03	<.01	.08	.11	-.03	.06	.13	.11	.07	.19	.14	.05	.21	.29	.20	.28	–
41. NS	-.23	-.15	-.19	-.09	-.07	-.02	-.08	-.14	.02	-.05	.03	-.02	<.01	.03	.08	<.01	.04	.10	-.14	-.11	-.26	-.11	<.01	-.02	-.01	-.03	-.06	-.05	-.04	.08	.04	<.01	.11	.04	-.01	.12	.16	.13	.22	.40	–
42. Raven	-.21	-.18	-.08	-.11	-.07	<.01	-.01	-.11	.12	-.01	.01	.05	.03	.05	.11	.02	.03	.09	-.12	-.06	-.11	-.10	-.03	-.09	-.04	.04	.12	.04	<.01	.08	.15	.13	.15	.13	.07	.15	.22	.23	.17	.28	.32	–
43. PicSour	-.17	-.19	-.16	-.16	-.03	-.11	-.06	-.06	-.02	-.07	-.07	.04	.01	-.03	.05	-.02	-.10	-.04	-.15	-.07	.08	-.09	.05	-.03	-.02	<.01	.09	-.13	.10	-.05	.08	.17	.15	.06	.09	.12	.15	.26	.15	.16	.14	.23	–
44. PA	-.04	-.05	-.15	-.06	-.01	.02	-.04	-.06	.11	-.03	<.01	-.05	.01	.01	.09	.02	-.06	.02	-.10	-.08	-.20	<.01	-.08	.06	-.11	-.06	-.01	-.02	-.05	.10	-.01	.03	.08	.03	-.05	.17	.12	.14	.11	.24	.20	.08	.06	–
45. DFR	-.18	-.18	-.25	-.17	-.06	.07	-.09	-.13	.06	.01	.07	.07	.03	.09	.15	.06	.05	.16	-.23	-.12	-.10	-.18	-.13	-.04	-.12	-.09	.05	-.04	-.01	.10	.04	.07	.21	.10	.01	.18	.21	.26	.29	.32	.32	.23	.23	.43

Note. c1-c4 = levels 1-4, in cue-based format; s1-4 = levels 1-4, in serial-order format. In the cue-based format, levels 3-4 include both non-integration (3, 4) and integration (3i, 4i) variables. In the serial-order format, levels 3-4 include both non-transition (3, 4) and transition (3i, 4i) variables. D = dot rule task; O = odd one out task; N = number judgment task; LS = Letter Sets; NS = Number Series; Raven = Raven Advanced Progressive Matrices (with the imputed values); Color = K estimate from color change detection task; Orient = K estimate from orientation change detection task; Space = K estimate from space capacity task; PicSour = picture source recognition task; PA = paired associates task; DFR = delayed free recall task. Correlations > .14 are significant at $p < .05$ level.

REFERENCES CITED

Amalric, M., Wang, L., Pica, P., Figueira, S., Sigman, M., & Dehaene, S. (2017). The language of geometry: Fast comprehension of geometrical primitives and rules in human adults and preschoolers. *PLOS Computational Biology*, *13*(1), e1005273. https://doi.org/10.1371/journal.pcbi.1005273

Aslin, R. N., & Newport, E. L. (2012). Statistical learning: From acquiring specific items to forming general rules. *Current Directions in Psychological Science*, *21*(3), 170–176.

Badre, D. (2008). Cognitive control, hierarchy, and the rostro–caudal organization of the frontal lobes. *Trends in Cognitive Sciences*, *12*(5), 193–200. https://doi.org/10.1016/j.tics.2008.02.004

Badre, D. (2012). Opening the gate to working memory. *Proceedings of the National Academy of Sciences*, *109*(49), 19878–19879. https://doi.org/10.1073/pnas.1216902109

Badre, D., & D'Esposito, M. (2007). Functional Magnetic Resonance Imaging Evidence for a Hierarchical Organization of the Prefrontal Cortex. *Journal of Cognitive Neuroscience*, *19*(12), 2082–2099. https://doi.org/10.1162/jocn.2007.19.12.2082

Badre, D., & D'Esposito, M. (2009). Is the rostro-caudal axis of the frontal lobe hierarchical? *Nature Reviews Neuroscience*, *10*(9), 659–669. https://doi.org/10.1038/nrn2667

Badre, D., Hoffman, J., Cooney, J. W., & D'Esposito, M. (2009). Hierarchical cognitive control deficits following damage to the human frontal lobe. *Nature Neuroscience*, *12*(4), 515–522. https://doi.org/10.1038/nn.2277

Badre, D., & Nee, D. E. (2018). Frontal cortex and the hierarchical control of behavior. *Trends in Cognitive Sciences*, *22*(2), 170–188.

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, *67*(1). https://doi.org/10.18637/jss.v067.i01

Bo, J., & Seidler, R. D. (2009). Visuospatial working memory capacity predicts the organization of acquired explicit motor sequences. *Journal of Neurophysiology*, *101*(6), 3116–3125. https://doi.org/10.1152/jn.00006.2009

Botvinick, M. M., & Plaut, D. C. (2002). Representing task context: Proposals based on a connectionist model of action. *Psychological Research*, *66*(4), 298–311. https://doi.org/10.1007/s00426-002-0103-8

Botvinick, M. M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, *111*(2), 395–429. https://doi.org/10.1037/0033-295X.111.2.395

Brewer, G. A., & Unsworth, N. (2012). Individual differences in the effects of retrieval from long-term memory. *Journal of Memory and Language*, *66*(3), 407–415. https://doi.org/10.1016/j.jml.2011.12.009

Bryck, R. L., & Mayr, U. (2005). On the role of verbalization during task set selection: Switching or serial order control? *Memory & Cognition*, *33*(4), 611–623. https://doi.org/10.3758/BF03195328

Burgess, P. W., Veitch, E., de Lacy Costello, A., & Shallice, T. (2000). The cognitive and neuroanatomical correlates of multitasking. *Neuropsychologia*, *38*(6), 848–863. https://doi.org/10.1016/S0028-3932(99)00134-7

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological Review*, *97*(3), 404–431. https://doi.org/10.1037/0033-295X.97.3.404

Collard, R., & Povel, D.-J. (1982). Theory of serial pattern production: Tree traversals. *Psychological Review*, *89*(6), 693–707. https://doi.org/10.1037/0033-295X.89.6.693

Cooper, R. P., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, *17*(4), 297–338. https://doi.org/10.1080/026432900380427

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87–114. https://doi.org/10.1017/S0140525X01003922

Crittenden, B. M., & Duncan, J. (2014). Task difficulty manipulation reveals multiple demand activity but no frontal lobe hierarchy. *Cerebral Cortex*, *24*(2), 532–540. https://doi.org/10.1093/cercor/bhs333

Davachi, L., & DuBrow, S. (2015). How the hippocampus preserves order: The role of prediction and context. *Trends in Cognitive Sciences*, *19*(2), 92–99. https://doi.org/10.1016/j.tics.2014.12.004

Dehaene, S., Kerszberg, M., & Changeux, J.-P. (1998). A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, *95*(24), 14529–14534. https://doi.org/10.1073/pnas.95.24.14529

Dehaene, S., Meyniel, F., Wacongne, C., Wang, L., & Pallier, C. (2015). The neural representation of sequences: From transition probabilities to algebraic patterns and linguistic trees. *Neuron*, *88*(1), 2–19. https://doi.org/10.1016/j.neuron.2015.09.019

Duncan, J. (2010). The multiple-demand (MD) system of the primate brain: Mental programs for intelligent behaviour. *Trends in Cognitive Sciences*, *14*(4), 172–179.

Duncan, J., Parr, A., Woolgar, A., Thompson, R., Bright, P., Cox, S., Bishop, S., & Nimmo-Smith, I. (2008). Goal neglect and Spearman's g: Competing parts of a complex task. *Journal of Experimental Psychology: General*, *137*(1), 131–148. https://doi.org/10.1037/0096-3445.137.1.131

Ehrlich, D. B., & Murray, J. D. (2021). *Geometry of neural computation unifies working memory and planning* [Preprint]. Neuroscience. https://doi.org/10.1101/2021.02.01.429156

Engle, R. W., Tuholski, S. W., Laughlin, J. E., & Conway, A. R. A. (1999). Working memory, short-term memory, and general fluid intelligence: A latent-variable approach. *Journal of Experimental Psychology: General*, *128*(3), 309–331. https://doi.org/10.1037/0096-3445.128.3.309

Farooqui, A. A., Mitchell, D., Thompson, R., & Duncan, J. (2012). Hierarchical organization of cognition reflected in distributed frontoparietal activity. *Journal of Neuroscience*, *32*(48), 17373–17381. https://doi.org/10.1523/JNEUROSCI.0598-12.2012

Fitch, W. T., & Martins, M. D. (2014). Hierarchical processing in music, language, and action: Lashley revisited. *Annals of the New York Academy of Sciences*, *1316*(1), 87–104. https://doi.org/10.1111/nyas.12406

Frank, M. J., & Badre, D. (2012). Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: Computational analysis. *Cerebral Cortex*, *22*(3), 509–526. https://doi.org/10.1093/cercor/bhr114

Frank, M. J., Loughry, B., & O'Reilly, R. C. (2001). Interactions between frontal cortex and basal ganglia in working memory: A computational model. *Cognitive, Affective, & Behavioral Neuroscience*, *1*(2), 137–160. https://doi.org/10.3758/CABN.1.2.137

Henson, R. N. A., Norris, D. G., Page, M. P. A., & Baddeley, A. D. (1996). Unchained memory: Error patterns rule out chaining models of immediate serial recall. *Quarterly Journal of Experimental Psychology*, *49A*(1), 80–115. https://doi.org/10.1080/027249896392810

Just, M. A., & Carpenter, P. A. (1985). Cognitive coordinate systems: Accounts of mental rotation and individual differences in spatial ability. *Psychological Review*, *92*(2), 137–172. https://doi.org/10.1037/0033-295X.92.2.137

Kessler, Y., & Meiran, N. (2008). Two dissociable updating processes in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *34*(6), 1339–1348. https://doi.org/10.1037/a0013078

Kessler, Y., & Oberauer, K. (2014). Working memory updating latency reflects the cost of switching between maintenance and updating modes of operation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *40*(3), 738–754. https://doi.org/10.1037/a0035545

Kikumoto, A., & Mayr, U. (2018). Decoding hierarchical control of sequential behavior in oscillatory EEG activity. *ELife*, *7*, e38550. https://doi.org/10.7554/eLife.38550

Koch, I., & Hoffmann, J. (2000). Patterns, chunks, and hierarchies in serial reaction-time tasks. *Psychological Research*, *63*(1), 22–35.

Koechlin, E., & Jubault, T. (2006). Broca's area and the hierarchical organization of human behavior. *Neuron*, *50*(6), 963–974.

Koechlin, E., Ody, C., & Kouneiher, F. (2003). The architecture of cognitive control in the human prefrontal cortex. *Science*, *302*(5648), 1181–1185. https://doi.org/10.1126/science.1088545

Krampe, R., Mayr, U., & Kliegl, R. (2005). Timing, sequencing, and executive control in repetitive movement production. *Journal of Experimental Psychology: Human Perception and Performance*, *31*(3), 379–397. https://doi.org/10.1037/0096-1523.31.3.379

Kray, J., & Lindenberger, U. (2000). Adult age differences in task switching. *Psychology and Aging*, *15*(1), 126–147. https://doi.org/10.1037/0882-7974.15.1.126

Lashley, K. S. (1951). The Problem of Serial Order in Behavior. In L. A. Jeffress (Ed.), *Cerebral Mechanisms in Behavior: The Hixon Symposium* (pp. 112–146). Wiley.

Lerche, V., von Krause, M., Voss, A., Frischkorn, G. T., Schubert, A.-L., & Hagemann, D. (2020). Diffusion modeling and intelligence: Drift rates show both domain-general and domain-specific relations with intelligence. *Journal of Experimental Psychology: General*, *149*(12), 2207–2249. https://doi.org/10.1037/xge0000774

Lien, M.-C., & Ruthruff, E. (2004). Task switching in a hierarchical task structure: Evidence for the fragility of the task repetition benefit. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(3), 697–713. https://doi.org/10.1037/0278-7393.30.3.697

Liesefeld, H. R., & Janczyk, M. (2019). Combining speed and accuracy to control for speed-accuracy trade-offs(?). *Behavior Research Methods*, *51*(1), 40–60. https://doi.org/10.3758/s13428-018-1076-x

Marcus, G. F., Vijayan, S., Bandi Rao, S., & Vishton, P. M. (1999). Rule learning by seven-month-old infants. *Science*, *283*(5398), 77–80.

Marshalek, B., Lohman, D. F., & Snow, R. E. (1983). The complexity continuum in the radex and hierarchical models of intelligence. *Intelligence*, *7*(2), 107–127. https://doi.org/10.1016/0160-2896(83)90023-5

Mayr, U. (2001). Age differences in the selection of mental sets: The role of inhibition, stimulus ambiguity, and response-set overlap. *Psychology and Aging*, *16*(1), 96–109. https://doi.org/10.1037/0882-7974.16.1.96

Mayr, U. (2002). Inhibition of action rules. *Psychonomic Bulletin & Review*, *9*(1), 93–99. https://doi.org/10.3758/BF03196261

Mayr, U. (2009). Sticky plans: Inhibition and binding during serial-task control. *Cognitive Psychology*, *59*(2), 123–153. https://doi.org/10.1016/j.cogpsych.2009.02.004

Mayr, U., & Keele, S. W. (2000). Changing internal constraints on action: The role of backward inhibition. *Journal of Experimental Psychology: General*, *129*(1), 4–26. https://doi.org/10.1037/0096-3445.129.1.4

Mayr, U., & Kliegl, R. (1993). Sequential and coordinative complexity: Age-based processing limitations in figural transformations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *19*(6), 1297–1320. https://doi.org/10.1037/0278-7393.19.6.1297

Mayr, U., Kuhns, D., & Hubbard, J. (2014). Long-term memory and the control of attentional control. *Cognitive Psychology*, *72*, 1–26. https://doi.org/10.1016/j.cogpsych.2014.02.001

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the Structure of Behavior*. Henry Holt and Co. https://doi.org/10.1037/10039-000

Monsell, S. (2003). Task switching. *Trends in Cognitive Sciences*, *7*(3), 134–140. https://doi.org/10.1016/S1364-6613(03)00028-7

Moss, M. E., Kikumoto, A., & Mayr, U. (2020). Does conflict resolution rely on working memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *46*(12), 2410–2426. https://doi.org/10.1037/xlm0000801

Moss, M. E., & Mayr, U. (2022). What's so hard about hierarchical control? Pinpointing processing constraints within cue-based and serial-order control structures. *Cognitive Psychology*.

O'Reilly, R. C. (2006). Biologically based computational models of high-level cognition. *Science*, *314*(5796), 91–94. https://doi.org/10.1126/science.1127242

O'Reilly, R. C., & Frank, M. J. (2006). Making Working Memory Work: A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia. *Neural Computation*, *18*(2), 283–328. https://doi.org/10.1162/089976606775093909

Povel, D.-J., & Collard, R. (1982). Structural factors in patterned finger tapping. *Acta Psychologica*, *52*(1–2), 107–123. https://doi.org/10.1016/0001-6918(82)90029-4

R Core Team. (2018). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Ranti, C., Chatham, C. H., & Badre, D. (2015). Parallel temporal dynamics in hierarchical cognitive control. *Cognition*, *142*, 205–229. https://doi.org/10.1016/j.cognition.2015.05.003

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, *20*(4), 873–922. https://doi.org/10.1162/neco.2008.12-06-420

Redick, T. S., Shipstead, Z., Meier, M. E., Montroy, J. J., Hicks, K. L., Unsworth, N., Kane, M. J., Hambrick, D. Z., & Engle, R. W. (2016). Cognitive predictors of a common multitasking ability: Contributions from working memory, attention control, and fluid intelligence. *Journal of Experimental Psychology: General*, *145*(11), 1473–1492. https://doi.org/10.1037/xge0000219

Restle, F. (1970). Theory of serial pattern learning: Structural trees. *Psychological Review*, *77*(6), 481–495. https://doi.org/10.1037/h0029964

Restle, F., & Burnside, B. L. (1972). Tracking of serial patterns. *Journal of Experimental Psychology*, *95*(2), 299-. https://doi.org/10.1037/h0033619

Reynolds, J. R., O'Reilly, R. C., Cohen, J. D., & Braver, T. S. (2012). The function and organization of lateral prefrontal cortex: A test of competing hypotheses. *PLoS ONE*, *7*(2), e30284. https://doi.org/10.1371/journal.pone.0030284

Riddle, J., Vogelsang, D. A., Hwang, K., Cellier, D., & D'Esposito, M. (2020). Distinct oscillatory dynamics underlie different components of hierarchical cognitive control. *Journal of Neuroscience*, *40*(25), 4945–4953. https://doi.org/10.1523/JNEUROSCI.0617-20.2020

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., & Fusi, S. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, *497*(7451), 585–590. https://doi.org/10.1038/nature12160

Rosenbaum, D. A., Kenny, S. B., & Derr, M. A. (1983). Hierarchical control of rapid movement sequences. *Journal of Experimental Psychology: Human Perception and Performance*, *9*(1), 86–102. https://doi.org/10.1037/0096-1523.9.1.86

Rubin, O., & Meiran, N. (2005). On the origins of the task mixing cost in the cuing task-switching paradigm. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*(6), 1477–1491. https://doi.org/10.1037/0278-7393.31.6.1477

Schneider, D. W., & Logan, G. D. (2006). Hierarchical control of cognitive processes: Switching tasks in sequences. *Journal of Experimental Psychology: General*, *135*(4), 623–640. https://doi.org/10.1037/0096-3445.135.4.623

Unsworth, N., & Engle, R. W. (2007). On the division of short-term and working memory: An examination of simple and complex span and their relation to higher order abilities. *Psychological Bulletin*, *133*(6), 1038–1066. https://doi.org/10.1037/0033-2909.133.6.1038

Unsworth, N., Fukuda, K., Awh, E., & Vogel, E. K. (2014). Working memory and fluid intelligence: Capacity, attention control, and secondary memory retrieval. *Cognitive Psychology*, *71*, 1–26. https://doi.org/10.1016/j.cogpsych.2014.01.003

Verwey, W. B., Shea, C. H., & Wright, D. L. (2015). A cognitive framework for explaining serial processing and sequence execution strategies. *Psychonomic Bulletin & Review*, *22*(1), 54–77. https://doi.org/10.3758/s13423-014-0773-4

Voss, A., Voss, J., & Lerche, V. (2015). Assessing cognitive processes with diffusion model analyses: A tutorial based on fast-dm-30. *Frontiers in Psychology*, *6*. https://doi.org/10.3389/fpsyg.2015.00336

Waltz, J. A., Lau, A., Grewal, S. K., & Holyoak, K. J. (2000). The role of working memory in analogical mapping. *Memory & Cognition*, *28*(7), 1205–1212. https://doi.org/10.3758/BF03211821

Wasylyshyn, C., Verhaeghen, P., & Sliwinski, M. J. (2011). Aging and task switching: A meta-analysis. *Psychology and Aging*, *26*(1), 15–20. https://doi.org/10.1037/a0020912

Wu, X., He, M., Zhou, Y., Xiao, J., & Luo, J. (2017). Decomposing a chunk into its elements and reorganizing them as a new chunk: The two different sub-processes underlying insightful chunk decomposition. *Frontiers in Psychology*, *8*. https://doi.org/10.3389/fpsyg.2017.02001

Yokoi, A., & Diedrichsen, J. (2019). Neural organization of hierarchical motor sequence representations in the human neocortex. *Neuron*, *103*(6), 1178-1190.e7. https://doi.org/10.1016/j.neuron.2019.06.017