On the Rational Boundedness of Cognitive Control:
Shared Versus Separated Representations

Sebastian Musslick[*,1],

Andrew Saxe[2],

Abigail Novick[3],

Daniel Reichman[4],

Jonathan D. Cohen[1,3]

[1]Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544, USA.

[2]Department of Experimental Psychology, University of Oxford, Oxford OX1 2JD, UK.

[3]Department of Psychology, Princeton University, Princeton, NJ 08544, USA.

[4]Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA.

[*] Correspondence: musslick@princeton.edu

Abstract

One of the most fundamental and striking limitations of human cognition appears to be a constraint in the number of control-dependent processes that can be executed at one time. This constraint motivates one of the most influential tenets of cognitive psychology: that cognitive control relies on a central, limited capacity processing mechanism that imposes a seriality constraint on processing. Here we provide a formally explicit challenge to this view. We argue that the causality is reversed: the constraints on control-dependent behavior reflect a rational bound that control mechanisms impose on processing, to prevent processing interference that arises if two or more tasks engage the same resource to be executed. We use both mathematical and numerical analyses of shared representations in neural network architectures to articulate the theory, and demonstrate its ability to explain a wide range of phenomena associated with control-dependent behavior. Furthermore, we argue that the need for control, arising from the shared use of the same resources by different tasks, reflects the optimization of a fundamental tradeoff intrinsic to network architectures: the increase in learning efficacy associated with the use of shared representations, versus the efficiency of parallel processing (i.e., multitasking) associated with task-dedicated representations. The theory helps frame a formally rigorous, normative approach to the tradeoff between control-dependent processing versus automaticity, and relates to a number of other fundamental principles and phenomena concerning cognitive function, and computation more generally.

On the Rational Boundedness of Cognitive Control: Shared Versus Separated Representations

1 Introduction

One of the most remarkable features of human cognition is the ability to override habitual (automatic) responses to successfully guide behavior in the service of current task goals. Mechanisms underlying this function are summarized under the term cognitive control. They are engaged across various domains of cognition, including perception, attention, learning, memory and action selection (Anderson, 1982; Badre & Wagner, 2007; Lavie, Hirst, De Fockert, & Viding, 2004; Posner & Snyder, 1975; Ridderinkhof, Van Den Wildenberg, Segalowitz, & Carter, 2004; Shiffrin & Schneider, 1977), and appear to be fundamental to many of the faculties that distinguish human mental function from other species (and continue to distinguish it from machines), including problem solving, planning and language processing (Miyake & Friedman, 1998; Otto, Skatova, Madlon-Kay, & Daw, 2014; Shah & Miyake, 1996; Sweller, 1988).

Cognitive control has often been treated as an undifferentiated construct. However, recent work has begun to focus on a distinction between mechanisms responsible for the *execution* of control, that is the regulation of processes subject to control; and mechanisms responsible for the *allocation* of control, that is monitoring internal states and/or the environment, including the outcome of processing, and determining based on that information how control should be allocated. For example, when confronted with the opportunity to perform one or more of several control-demanding tasks, before committing to performing any of them there may be an initial phase during which the individual considers which (and possibly how many) it is best to perform (Fischer & Plessow, 2015) – that is, how to allocate control. How people make such determinations has been the focus of increasing theoretical interest, including attempts to provide a normative account from a resource rational perspective (Shenhav, Botvinick, & Cohen, 2013; Shenhav et al., 2017; Lieder, Shenhav, Musslick,

& Griffiths, 2018). These proceed from the assumption that the allocation of control is constrained – an assumption that, as we will elaborate below – has been central to virtual all theory concerning cognitive control – and cast the question of how control should be allocated as an optimization problem, that people seek to solve by evaluating candidate opportunities in terms of their expected future value weighed against the cost of allocation. The latter is generally formulated as an opportunity cost: what is lost by forestalling or even forgoing other tasks to pursue a chosen one (or few). However, like virtually all other theoretical work on cognitive control, these theories do not explain *why* the allocation of control is constrained. This article seeks to address that question, with the goal of grounding our broader understanding of cognitive control on a firmer normative foundation. Below, we discuss the constraint associated with control, followed by a brief review of explanations that have been given for it, before providing a formal theory.

1.1 Capacity Constraints

Despite the powerful abilities that cognitive control affords, and its ubiquitous engagement in daily life (e.g., mentally planning one's day at work, or navigating an alternate route to work), it has long been recognized that we have a dramatically limited ability to carry out more than one (or a very few) control-dependent processes at the same time (e.g., the inability to plan and navigate at the same time). This limitation has been literally paradigmatic since the earliest efforts to define cognitive control: it was used to distinguish it from automatic processing (Posner & Snyder, 1975; Shiffrin & Schneider, 1977), and is used universally to operationalize it in the laboratory (i.e., "diagnose" it experimentally) in the form of dual-task interference (Lavie et al., 2004; McLeod, 1977; Meyer & Kieras, 1997b; Welford, 1952). A constraint in the capacity for control-dependent processing has also become a theoretical cornerstone of virtually all major theories of cognitive function (Anderson et al., 2004; Anderson & Lebiere, 2014; Pashler & Sutherland, 1998; Simon, 1957), including ones, noted above, that address how rational choices are made among the limited set of control-dependent behaviors that can be carried out at a given time (Kurzban, Duckworth, Kable, & Myers, 2013; Lieder et al., 2018; Shenhav et al., 2013). Despite the central importance of the constraints associated with the engagement of cognitive control, the source of the constraint *itself* remains a mystery.

1.1.1 Structural Constraints. A widely accepted view is that constraints in the capacity for control-dependent processing arise from structural limitations inherent to the control system itself. One of the earliest, and still most influential views is that cognitive control relies on a centralized, limited capacity mechanism that imposes a seriality constraint on processing (e.g., Posner & Snyder, 1975; Shiffrin & Schneider, 1977). This reflects two strong influences. One is an analogy with the traditional computer architecture (e.g., von Neumann, 1958), that has at its core a single, general purpose central processing unit (CPU) with a limited buffer that allows it to execute a single program instruction at a time (Kerr, 1973). A second, convergent influence comes from the longstanding tradition of work on selective attention, in which the earliest theories proposed an attentional filter that limits information processing to selected stimulus features (Broadbent, 1957, 1958; Craik, 1948; Welford, 1952). These ideas have matured and been refined by an extensive literature on dual-task interference, that provide compelling evidence for a central processing bottleneck (e.g., Pashler, 1984, 1994).

The idea of a structural constraint has also been suggested by mechanistic models of cognition in which control relies on the active maintenance in working memory of representations needed to guide task performance (such as task instructions, goals, etc.; e.g., Anderson, 1984; E. K. Miller & Cohen, 2001). Accordingly, constraints on control could be due to the well characterized limitations in the capacity of working memory, such as a limited number of discrete slots for working memory representations (Cowan, Rouder, Blume, & Saults, 2012; Kriete, Noelle, Cohen, & O'Reilly, 2013; Luck & Vogel, 1997; Schneider, Detweiler, et al., 1987), their passive decay (Jensen, 1988; Page & Norris, 1998), interference among representations held in a common working memory buffer (Nairne, 1990; Oberauer & Kliegl, 2006; Usher & Cohen, 1999), or the related idea that there is a tradeoff between the number and precision of representations that can be actively maintained (Ma & Huang, 2009; Ma, Husain, & Bays, 2014) — for a comparative review of these accounts, see Oberauer, Farrell, Jarrold, and Lewandowsky (2016).

Even if dependence on working memory were responsible for the constraints on cognitive control, this leaves at least two mysteries unsolved: (1) Whereas the exact limits of working memory capacity are actively debated (is it 7, 4 or even just 2? Cowan, 2001, 2010; Luck & Vogel, 1997; G. A. Miller, 1956; Palmer, 1990; Turner & Engle, 1986), constraints on the simultaneous execution of controlled-dependent processing is even more severe: it is almost universally considered to be a *single* task (e.g., Anderson et al., 2004; Anderson & Lebiere, 2014; Pashler & Sutherland, 1998); (2) Why would a system with processing resources as vast as those of the human brain (with billions of neurons in the human cortex alone; Herculano-Houzel, 2009; Pelvig, Pakkenberg, Stark, & Pakkenberg, 2008) suffer from such a Draconian limitation on a function as adaptive as the capacity for cognitive control? In the face of modern compute clusters, with 1000s of "cores" or more, the analogy between cognitive control and an architecture with a single CPU has become as quaint as the architecture itself.

1.1.2 Multiple Resource Theory. An alternative to the idea that capacity constraints arise from the resource limitations of a centralized, control mechanism — that is, that they reflect a limitation of the control system *itself* — is the idea that they reflect, instead, properties of the processes that are being controlled. This idea was first expressed in the form of the *multiple resource theory* (Allport, 1980; Allport, Antonis, & Reynolds, 1972; Kinsbourne & Hicks, 1978; Navon & Gopher, 1979; McCracken & Aldrich, 1984; Walley & Weiden, 1973; Wickens, 1991). This proposes that control-demanding tasks, like any others, rely on a constellation of "local" resources (e.g., task-specific representations)[1], and that the inability to perform more than one

[1] . The terms "shared resource" and "shared representation" describe similar concepts in different models of human multitasking. In symbolic architectures, such as ACT-R (Anderson & Lebiere, 2014) or EPIC (Meyer & Kieras, 1997b), two tasks are considered to share a resource if both of the tasks

task at a time may reflect the conflict that arises within *local* resources when the tasks involved rely on the same local resources, but demand that they be used for different purposes, rather than reliance on a single *centralized* control mechanism. A classic demonstration of this was provided by Shaffer (1975), who contrasted two dual-task conditions. In one condition, participants were asked to repeat an auditory input stream out loud (echoing) while manually typing visually presented text (copy-typing); they were able to do this reasonably well after a modicum of practice. The other condition involved the same stimulus modalities (auditory and visual streams of verbal information) and response modalities (speaking and typing), in this case they were asked to type the auditory input (dictation) while reading aloud the visually presented text (reading); this proved virtually impossible to do, even after extensive practice. What is particularly striking is that one of the tasks in the second condition — word reading — is considered to be a canonical example of an automatic process (Warren, 1972; Posner & Snyder, 1975; R. F. West & Stanovich, 1978; Seidenberg, Tanenhaus, Leiman, & Bienkowski, 1982); since the response it demanded (verbal) was different from the dictation task (manual), it should not have produced interference, no less been subject to interference itself.

Fig. 1 illustrates these tasks and offers an explanation of the findings, in a manner consistent with the multiple resource theory. In the first condition, the two tasks each make independent use of two distinct "resources" (orthographic and phonological representations of verbal materials); in the second condition, both tasks must make use of both resources, each for a different purpose (i.e., to process different, competing

require engagement of the same processing component. A processing component may be used to represent declarative information (e.g., sensory information or more abstract semantic knowledge) or to manipulate information (e.g., productions for updating the activity of representations in declarative memory and/or taking actions). In connectionist models — consisting of multiple interconnected processing units, often grouped into modules that are used to represent and process a given type of information — two tasks can be considered to share a resource if they make use of the same units in a module (i.e., they "share representations") but require different representations to be active at the same time (cf. Fig. 3C).

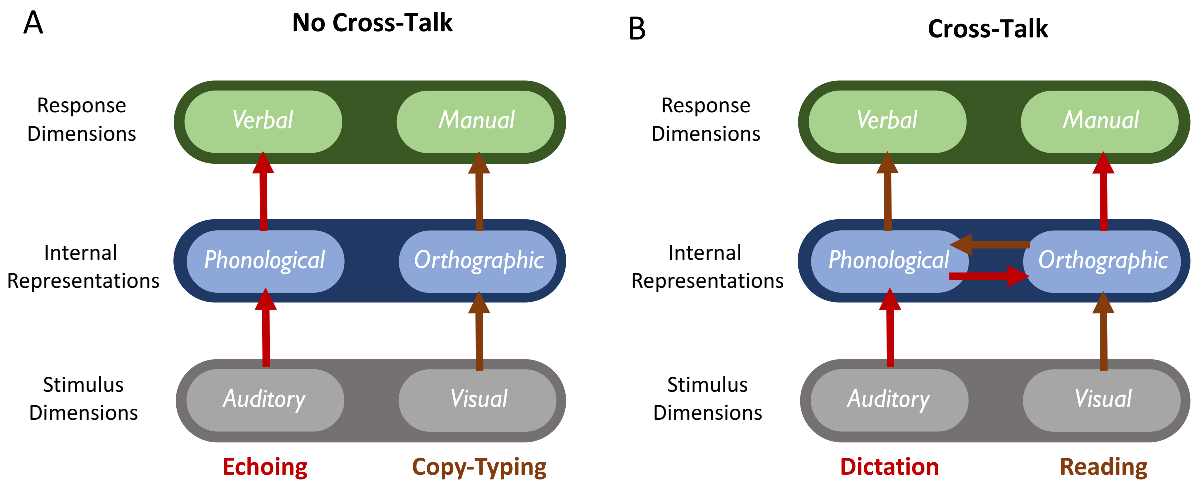

Figure 1. **Two dual-tasking conditions contrasted in the experiment by Schaffer (1975).** (A) In the first condition, participants were asked to repeat spoken words (echoing) while typing visually presented text (copy-typing). (B) In the second condition participants were asked to type spoken words (dictation) while vocalizing visually presented text (reading). Participants were able to learn to multitask in the first condition but were unable to do so in the latter. The difficulty of the second condition can be explained in terms of interference that arises from the shared use of representations for two different purposes. In that condition (B), the phonological and orthographic representations must each be used for two tasks (reading and echoing), leading to interference between them. No such interference is present in the first condition (A).

stimuli). From this perspective, the dual-task interference that arose in the second condition did not necessarily reflect the limited capacity of a centralized control mechanism, but rather the conflict that arose from making competing demands of the same local "resources" (on the assumption that each resource could not be used to simultaneously represent different information). Similar effects reflecting the sensitivity of dual-task interference to the particularities (often referred to as the "compatibility") of the stimulus-response mappings involved have continued to be widely reported in the literature (Greenwald, 1970; Greenwald & Shulman, 1973; Göthe, Oberauer, & Kliegl, 2016; Halvorson, Ebner, & Hazeltine, 2013; Hazeltine, Ruthruff, & Remington, 2006; Lien & Proctor, 2002; Liepelt, Fischer, Frensch, & Schubert, 2011).

Several computational models of cognitive function have implemented the idea that constraints on the number of tasks that can be performed at the same time arise due to the sharing of local resources, rather than a limitation in the mechanisms responsible for control. For example, the executive-process interactive control (EPIC) framework (Meyer & Kieras, 1997b; Kieras & Meyer, 1997) implements a control mechanism that schedules tasks, without any upper limit on the number that it can schedule for execution in parallel. Bottlenecks arise from seriality constraints within individual processing resources, when these are required for performance by more than one task at a time. Salvucci and Taatgen (2008) have described a similar view in the context of a theory of threaded cognition. Such modeling efforts based on symbolic architectures have been successful in predicting when multitasking performance is possible, and when constraints arise, based on assumptions about which resources are shared between specific tasks (Byrne & Anderson, 2001; Kieras, Meyer, Ballas, & Lauber, 2000; Meyer & Kieras, 1997a; Salvucci & Macuga, 2002; Salvucci, 2006). While these efforts have focused on people's ability to multitask, connectionist models have addressed the conflict that can arise from shared representations even when performing a single task (i.e., when information from a competing source impinges on the shared representations, such as in the Stroop and Erisken Flanker tasks), and the role that control plays in managing such conflict (e.g. Botvinick, Braver, Barch, Carter, & Cohen, 2001; J. D. Cohen, Dunbar, & McClelland, 1990).

1.1.3 Guilt by Association: Control as a Solution Rather than a Cause. The modeling efforts above all emphasize the point that a fundamental purpose of control mechanisms is to manage the potential for cross-talk between tasks, by restricting the engagement of representations shared by multiple processes to the one(s) relevant for a single process at any given time. That is, they make the point that the constraints on the simultaneous execution of multiple control-dependent processes, usually ascribed to the mechanisms responsible for control, can instead be viewed as the *purpose* of control — to limit cross-talk — rather than a *limitation* of control mechanisms *themselves.* Ascribing the constraints to a limitation in control mechanisms is mistaking correlation for causation, akin to blaming the fire fighters for the fire, since they are always at the fire. The real constraint is the sharing of representations by different processes, rather than assigning dedicated representations to each, not the control mechanisms responsible for adjudicating their use in a particular setting. However, this perspective does beg the following question: Why, if the sharing of resources leads to conflict, constraints on processing, and reliance on control, should such sharing arise in the first place, no less be as prevalent as the bottlenecks associated with controlled processing seem to be?

One potential answer to this question, and several closely related ones, is suggested by a different analogy, between the role of cognitive control in information processing and that of a traffic controller in a transit system. Think of each process in the cognitive system as a vehicle, conveying goods (by analogy, information) from a source to a destination. Ideally, each vehicle travels on a thoroughfare that runs directly from its source to its destination, without crossing any others. In this case, the system can function independently (i.e., automatically), without any need for a traffic controller. However, as the number of goods or, perhaps more importantly, the number of uses to which they are put, increases, it becomes increasingly difficult to avoid the crossing of routes. Where this occurs, there are two options. One is to build an overpass, so that the vehicles can continue to operate independently of one another or a controller. However, this can be costly and take time. Alternatively, the thoroughfares can be allowed to intersect. However, this introduces the risk of collisions, so the intersection must be accompanied by a traffic signal, and a traffic controller engaged to manage it. The role of traffic controller becomes increasingly important as the number of crossings and vehicles traversing them grows.

This analogy brings several critical points to light. First, using traffic signals rather than overpasses is faster and cheaper to implement, but restricts the flow of traffic. More specifically, it is the number of stop signals that must be imposed at any one time that constrains the traffic flow, and it is the responsibility of the traffic controller to impose these. The fact that the traffic controller imposes this restriction does not reflect a limitation of the controller (there is no practical limit to the number of signals available to it, nor any intrinsic limit on how many can be used to signal “go” vs. “stop” at any time); but rather, the restriction in the number of “go” signals reflects its *purpose* in preventing collisions. Analogously, the purpose of cognitive control is to limit cross-talk that arises from those parts of the processing system that involve "crossings" — that is, shared representations.

1.1.4 Shared vs. Separated Representations.

The analogy above suggests a qualitative answer to the question of why the cognitive system should favor shared representations: Like traffic intersections, they may be easier, quicker, and/or cheaper to construct, and also more flexible (e.g., allowing processing to be quickly re-directed in a number of different directions), as compared to separated representations dedicated to each process (e.g., overpasses). This qualitative answer brings into focus two more specific, quantitative questions.

The first question is: How does multitasking[2] capacity scale with the size of the processing system, and the frequency of shared representations within it? By way of the analogy above, how does the risk of collisions scale with the number of crossings in the system? One might imagine that in a processing network with the capacity of the human brain, the likelihood of a given set of tasks relying on a shared set of representations might be relatively low, and should therefore play an insignificant role in constraining the number of tasks that can be performed in parallel. However, provisional numerical work suggests otherwise (Feng, Schwemmer, Gershman, & Cohen, 2014), motivating the need for a more rigorous analysis of the impact of representational sharing on network performance.

The second question is: How does the human cognitive system balance the costs and benefits for shared vs. separated representations? As noted above, previous computational modeling efforts have addressed the consequences of shared representations with respect to cross-talk and attendant constraints on multitasking, showing that mechanistically-explicit implementations of the multiple resource theory can provide quantitatively accurate accounts of human performance in task domains

[2] Here, we define multitasking as the *simultaneous* execution of two or more tasks to distinguish it from broader uses of the term, such as the switching between multiple tasks (Koch, Poljac, Müller, & Kiesel, 2018).

where there appear to be constraints on concurrent multitasking (Byrne & Anderson, 2001; Meyer & Kieras, 1997a; Salvucci & Taatgen, 2008). However, these have assumed a stationary resource taxonomy (see Wickens, 1991), based on pre-specified representations for the tasks involved, without specifying how or why those representations arose in the first place (Botvinick et al., 2001; Byrne & Anderson, 2001; J. D. Cohen et al., 1990; Laird, 2012; Meyer & Kieras, 1997b; Salvucci & Taatgen, 2008). That is, they have not provided an account of the factors that drive the system to rely on shared representations, at the cost of a reliance on control, versus the development of separated, task-dedicated representations that provide the efficiency of parallel processing and multitasking (i.e., automaticity).

1.2 Overview

The purpose of this article is to directly address both of the questions raised above: How does multitasking capability scale with the prevalence of representational sharing and the size of the processing system; and what are the factors that determine the tradeoff between shared and separated representations? For most of the article we focus on the domains of skill acquisition and task performance, however in the General Discussion we consider the extent to which the principles involved generalize to, or relate to others concerning the cognitive system more broadly.

We begin, in Part I, by describing a formal framework in which the balance between shared and separated representations, and the corresponding constraints on multitasking capability and demand for cognitive control, can be quantified. Next, we apply the framework to empirical findings from experimental tasks that have been used to study control-dependent processing, from classic “attentional” tasks (such as the Stroop paradigm) to dual-task and task switching paradigms. We show how the constraints imposed by shared representations can provide a unified framework for explaining behavioral effects commonly observed in these domains. Then, in Part II, we examine the influence that learning has on this balance, and illustrate how this can be used to provide a quantitative, and potentially normative account of the trajectory from controlled to automatic processing over the course of training.

We conclude by suggesting that the tradeoff between shared and separated representations, and its interaction with learning, represent a fundamental principle of adaptive network architectures that underlies and shapes all domains of psychological function, from perception and inference to task execution, and extends equally to artificial systems. Moreover, we discuss how solutions to this tradeoff can be approximated by considering a "cost of control" that has begun to receive considerable attention in theories of control allocation (e.g. Kool & Botvinick, 2018; Kurzban et al., 2013; Lieder & Griffiths, 2015; Shenhav et al., 2013, 2017), as well as in theories of planning and decision making (Callaway et al., 2018; Kool, Gershman, & Cushman, 2017; Lieder et al., 2018). We also consider how the tradeoff between shared and separated representations may help provide a unified understanding of a wide range of psychological phenomena that, to date, have been treated largely as distinct from one another — including the role of "chunking" in skill acquisition (G. A. Miller, 1956; Servan-Schreiber & Anderson, 1990), interference in working memory (Bouchacourt & Buschman, 2019; Usher & Cohen, 1999; Wilken & Ma, 2004), attention in "binding" (Treisman, 1996, 1999; Treisman & Gelade, 1980), facilitation in creativity (Kajić, Gosmann, Stewart, Wennekers, & Eliasmith, 2017; Schatz, Jones, & Laird, 2018), and the tradeoff between pattern separation vs. pattern completion in episodic vs. semantic memory (McClelland, McNaughton, & O'Reilly, 1995) — and discuss its relationship to similar principles that have begun to emerge from machine learning, such as the bias-variance tradeoff and regularization.

2 Part I: Shared vs. Separated Representations and Constraints on Multitasking Capability

We begin by describing a simple neural network model that has been used widely to implement a fundamental function of cognitive control: configuration of information processing in the service of performing a specified task. We use this model to define what we mean by the terms "task", "process", and "shared representation"; and how the configuration of processes used to perform tasks constrain the multitasking capability of a network, and consequently the demands for control. We show how constructs from graph theory can be used to analyze how the cross-talk associated with these different configurations impacts performance, and how these effects scale with the size of the network. We then demonstrate how these graph-theoretic methods can be used to predict the multitasking capability of a network from measures of single task representations. We also examine how the amount of conflict induced by shared representations interacts with the persistence characteristics of those representations to produce constraints on multitasking and dependence on control. We show that these interactions can account for patterns of reaction time (RT) that have been proposed to index the degree of parallel processing in task performance (Townsend & Wenger, 2004). Finally, we demonstrate how the constraints on parallel processing imposed by shared representations, and concomitant demands for control, provide a unifying account of phenomena associated with the sequential execution of multiple tasks, such as the psychological refractory period (PRP; Telford, 1931) and task switch costs (Allport, Styles, & Hsieh, 1994; R. D. Rogers & Monsell, 1995), and discuss how this can be used to define multitasking behavior along a continuum from pure sequential processing, through rapid task switching, to pure parallelism (Fischer & Plessow, 2015; Salvucci, Taatgen, & Borst, 2009).

2.1 A Simple Neural Network Model

We base our work on a family of neural network models that have been used previously to capture a wide range of empirical findings concerning controlled processing in attention and conflict tasks (e.g. J. D. Cohen et al., 1990; Botvinick et al., 2001; Gilbert & Shallice, 2002; Kalanthroff, Davelaar, Henik, Goldfarb, & Usher, 2018). In this section we describe the network architecture and processing in a canonical example of these models, and use this to illustrate some of the subtle, in addition to obvious ways in which shared vs. separated representations impact multitasking performance.

2.1.1 Architecture. The basic model consists of two input layers, one of which represents the stimulus presented to the network and another that indicates the task the network is required to perform on the stimulus. The stimulus information is transformed by a matrix of connection weights from the stimulus input layer to a hidden (associative) layer, where it is represented as a pattern of activity over the units in the hidden layer. A simple version of this model is depicted in Fig. 2. The pattern of activity over units in the hidden layer is used to determine the pattern of activity over the output layer that represents the response to a given stimulus. Control is implemented by projections from the task input layer to the hidden and output layers, that bias processing towards task-relevant representations in each of these layers, thus allowing the network to elicit different responses to the same stimulus, depending on the task specified. [3]

2.1.2 Tasks and Processes. Note that the stimulus input layer is comprised of several subsets of units, one for each dimension of information in the stimulus. Similarly, distinct subsets of output units are generally used to represent different response dimensions, although in the example shown in Fig. 2 there is only a single such dimension (for verbal responses; see Fig. 3 for an example with two response dimensions). We define a *task* as a one-to-one mapping from representations within a single stimulus dimension to ones in a single response dimension (for example, each color to a verbal response; i.e., its name). A *process* is the set of units and connections within a network used to implement a task. Thus, the model shown in Fig. 2, with two stimulus dimensions in its input layer and one response dimension in its output layer, is configured with two processes that can be engaged to perform either of two tasks: color naming or word reading. Fig. 3 shows an extended version of the Stroop model that adds a dimension for manual responses in the output layer, allowing the network to

[3] Note that all of our examples use one-hot ("localist") representations of input and output features within each dimension, but all of our findings apply equal to cases in which features are represented in a more distributed form (see Simulation Studies 4 and 6), so long as each feature is orthogonal to all the others.

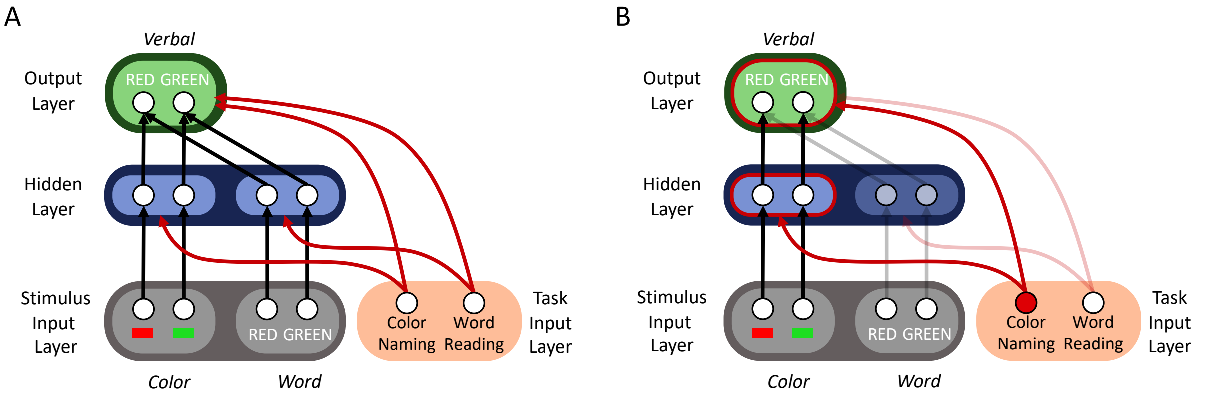

Figure 2. **Neural network model of the Stroop paradigm (adapted from Cohen et al., 1990).** (A) Model architecture: The input layer has two partitions: one represents the current stimulus (shown in gray) and projects to a hidden layer (shown in blue), and the other encodes the current task (shown in orange) and projects to both the hidden and output layers. The hidden layer projects to the output layer (shown in green). The output layer represents the network's response. Stimulus input units are structured according to stimulus dimensions (subvectors of the stimulus pattern), each of which is comprised of a set of feature units with one input unit activated per dimension corresponding to the stimulus feature in that dimension; in the present example there are two units in each dimension, one for red and the other for green (see Footnote 3). Similarly, output units are organized into response dimensions, with only one output unit permitted to be active per dimension corresponding to a selected response in that dimension; in the present example, there are two units, one for each of the two responses (there is also only a single response dimension — verbal; see Fig. 3 for an example with additional response dimensions). All units in the model are assumed to be inhibited at rest. Projections from each unit in the task input layer act as control signals that engage task-relevant units in the hidden and output layers by placing them in a more sensitive range of their activation function (see Cohen et al., 1990 for a more detailed explanation). (B) To execute the color naming task, a unit in the control layer is activated, which engages units in the hidden layer representing color input features (thus allowing them to overcome any interference from word features at the output layer); the control unit also engages units representing the verbal response dimension in the output layer, licensing a verbal response (relative to others that are not shown here).

perform two additional tasks: manually pressing a button to a particular color, and similarly for words. Thus, the model can now be instructed to perform any of four tasks: color naming, word reading, color mapping or word mapping. Importantly, however, whereas there is only one way to configure the two tasks as distinct processes in the Stroop model, there are several ways to configure the processes for the four tasks in the extended Stroop model, as discussed in the section that follows.

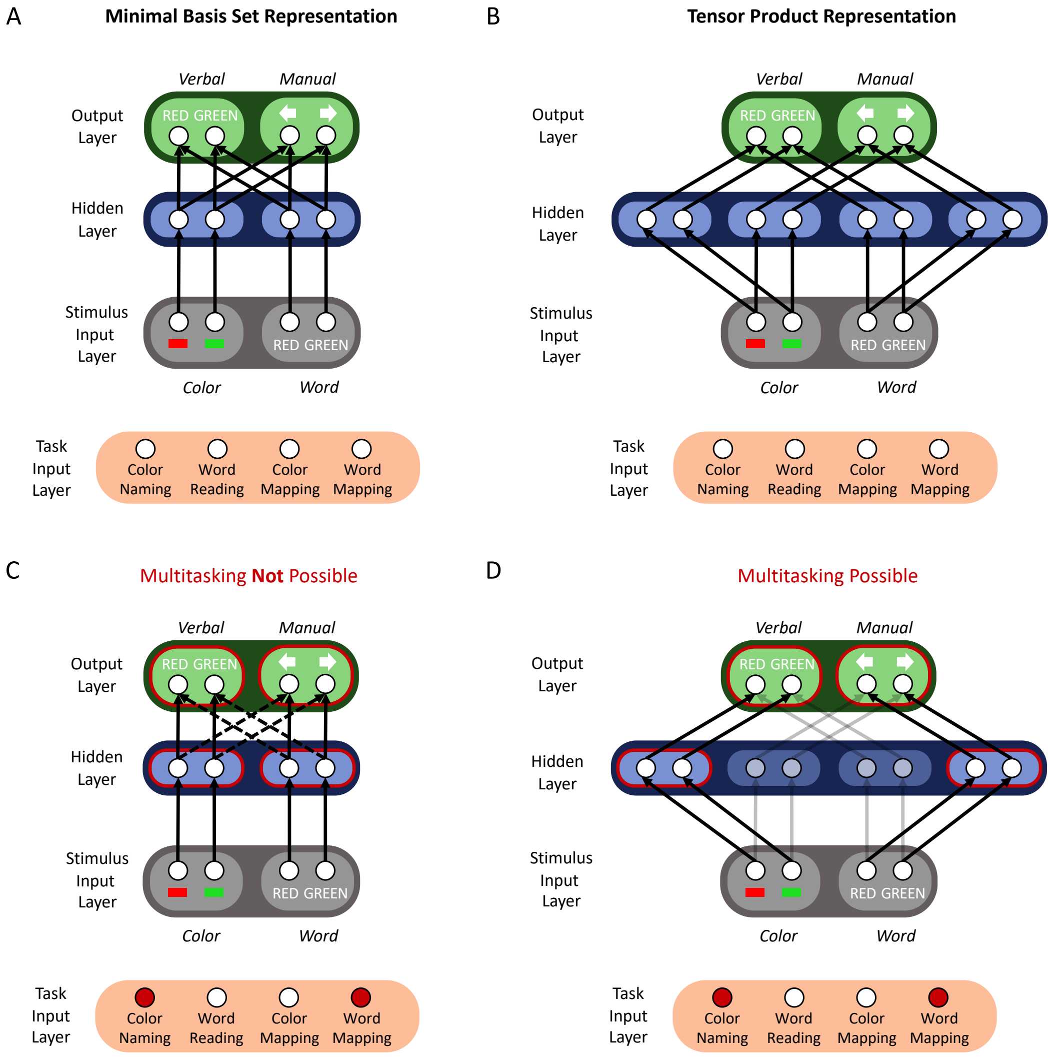

Figure 3. **Minimal basis set representation versus tensor product representation.** In a task-environment with two stimulus dimensions (e.g. ,color and word) and two response modalities (e.g., verbal and manual responses) the system can perform four tasks — that is, mappings from stimulus to response dimensions: color to verbal (color naming), color to manual (color mapping), word to verbal (word reading) and word to manual (word mapping). In the minimal basis set representation (A, C) tasks with common stimulus dimensions share the same representation in the hidden layer. In the tensor product representation (B, D) a separate representation is dedicated to each task. When asked to multitask (e.g., execute color naming and word mapping at the same time; red lines), the minimal basis set representation (C) leads to cross-talk in both response dimensions, which receive (possibly conflicting) information from each stimulus dimension (dashed lines). No such cross-talk occurs for the tensor product representation. Note that weights projecting from the task input layer are not shown.

2.1.3 Shared vs. Separated Representation: Minimal Basis Set and Tensor Product Configurations. The two panels of Fig. 3 show two ways in which the hidden units in Fig. 2 can be configured for the four processes required to perform the four possible tasks. These represent two extremes along the dimension of shared vs. separated representations, which help illustrate the advantages and disadvantages of each. In Fig. 3A, the hidden units are divided into two pools, as they are in the Fig. 2, each of which represents one of the two stimulus dimensions (for colors and words), and are connected to each of the two response dimensions (for verbal and manual responses). Thus, each pool of hidden units is shared by the processes for tasks involving a given stimulus dimension (e.g., color naming and color pointing), and thus connotes a shared resource in the network (see Footnote 1). This requires the fewest number of units and connections to implement all four tasks (four and eight, respectively). We refer to this as the "minimal basis set" configuration, reflecting the fact that it "spans the space" of all four tasks with the fewest number of elements. This has the advantage of representational efficiency; however, it has the disadvantage of not being able to reliably perform more than a single task at a time. If conflicting information is presented in the color and word stimulus dimensions (e.g., the color red and word GREEN), the model is unable to resolve which information should be conveyed to each of the response dimensions, e.g. when asked to execute color naming and word mapping at the same time (see Fig. 3C). This is an analog of the second condition in the Shaffer (1975) dual-task experiment discussed above, and provides the simplest example of the constraints on multitasking imposed by shared representations.[4] We will return to this at length below.

The configuration in Fig. 3B overcomes this problem, by implementing processes using a dedicated set of hidden units for each task. We refer to this as the "tensor product" configuration, referring to the fact that it assigns a separate set of representations for each pairwise combination ("product") of stimulus and response

[4] This might also be recognized as isomorphic to the "binding" problem that arises from often discussed in the context of perception, to which we will return in the General Discussion.

dimensions. This solves the problem faced by the minimal basis set configuration, allowing the maximum number of tasks to be performed simultaneously; that is, that do not involve competing input and/or output representations. However, this comes at the cost of requiring a greater number of hidden units and weights (eight and sixteen, respectively). It can also take longer to learn — a critical consideration that we address in Part II of this article.

2.1.4 Control. The focus of this article is on the impact that representations within the processing pathways required to perform a task (i.e., those in the input, hidden and output layers) have on the capability to execute multiple tasks at the same time and, to the extent that this must be limited, the resulting requirements for control. However, it is worth taking a moment to briefly consider how different configurations of the processing pathways (such as the two extremes shown in Fig. 3) impact the *representational* demands on the *control* components of the system (i.e., the task layer and its projections). In all cases, the sharing of representations for response dimensions at the output layer introduces the potential for conflict, and thus the need for control at that level (see Simulation 6 in J. D. Cohen et al. (1990)). That is, constraints imposed by shared output representations (also referred to as "peripheral interference"; Wickens, 1991) establish a minimum amount of control needed for performance of any task: we have only one mouth, two hands, one set of eyes, etc., and we must chose how to use them. However, the configuration of hidden representations (i.e., at the hidden layer) introduces interesting differences in the requirements for control. For the network used in Fig. 3A, the minimal basis set can be managed with four control units: one for each of the two stimulus dimensions represented in the hidden layer, and one for each of the two response dimensions in the output layer. Any of the four tasks can be selected for performance by activating one from each pair[5]. More generally, a minimal basis set

[5] Note that a partitioning of control units by stimulus and response dimensions is different from a partitioning of control units by tasks (as depicted in Fig. 3). In this section, we choose the former to illustrate representational demands on control. However, for simplicity, we choose the latter in all simulations reported below.

configuration (for a network with a single hidden layer) requires a number of control units equal to the sum of the stimulus and response dimensions; that is, it scales *additively* with the stimulus and response dimensionality of the network. In contrast, the control requirement of the tensor product configuration, as its name suggests, scales *multiplicatively* with the number of stimulus and response dimensions. In the example in Fig. 3B, the number of control units required is 4: one for each combination of stimulus dimension and response dimension.[6] For this particular example, this is the same as the minimal basis set. However, if the number of stimulus and/or response dimensions increases, the representational requirements for the tensor product configuration grow multiplicatively with the product of those dimensions. For example, for a network with three stimulus and three response dimensions, the minimal basis set configuration requires six control units, but the tensor product configuration requires nine. Therefore, the tensor product configuration has representational requirements — both for hidden units and control — that grow exponentially with the number of possible tasks relative to those of the minimal basis set. This is one factor that may contribute to reduced learning efficiency with tensor product representations, as discussed in Part II.

2.1.5 Multitasking Capability and Network Size.

The minimal basis set and tensor product are two extremes along a continuum of possible configurations, that highlight an inherent tension between representational efficiency (favored by the minimal basis set configuration) and the number of tasks that can be performed concurrently (favored by the tensor product configuration), as a function of the extent to which representations are shared across tasks. We refer to the number of tasks that can be performed concurrently (i.e., multitasked) as the *multitasking capability* of a network, that reflects its processing efficiency (how many tasks it can reliably perform per unit time). As the size of a network increases, so does the number of possible

[6] Whereas the minimal basis set requires that control be engaged at *both* the hidden layer *and* the output layer, the tensor product configuration can be parameterized to require control *only* at the hidden layer — e.g., by assigning a strong negative bias to all of the output units, and insuring that the weights from the hidden layer to the output layer are strong enough to overcome that bias.

configurations, which raises the question of what exactly is the impact of shared representations on the multitasking capability of a network as a function of its size. That is, just how much of a problem is representational sharing in larger networks? It might be assumed that, for a fixed proportion of shared representations, multitasking capability scales with the size of a network; that is, the proportion of representations that are shared across tasks determines the proportion of tasks that can be performed simultaneously. However, recent theoretical results suggest otherwise.

Feng et al. (2014) carried out initial numerical analyses to address this question. They simulated two types of networks: one involving a simple linear mapping from inputs to outputs for each task, and another in which each task was implemented as a drift diffusion process (Ratcliff & Rouder, 1998) to accommodate dynamics of performance and analytic optimization (Bogacz, Brown, Moehlis, Holmes, & Cohen, 2006). In both cases, the processing pathway implementing each task could be engaged or disengaged by a corresponding control signal. Simulations were carried out for both types of networks that varied their size and the degree of overlap among tasks (i.e., sharing of representations). Each simulation involved dense optimization of all processing parameters over control policies to determine the optimal one for a given network configuration — that is the degree of control allocated to each task that optimized performance of the network as a whole. For the linear model, this was the mean error over the output units for all tasks; for the drift diffusion model, this was the aggregate reward rate over all tasks. In both cases, a dramatically sublinear relationship was observed between degree of task overlap (number of tasks that share a representation) and the number of tasks engaged by the optimal control policy, with a fixed asymptotic limit in the *absolute* number of tasks it was optimal to perform at once, *irrespective of the size of the network* (see Fig. 4).

These observations suggest that even modest sharing of representations across tasks can impose dramatic constraints on the number of tasks that can accurately be performed at once. However, the results were obtained using two specific models, each of which made a number of simplifying assumptions. While most of these assumptions

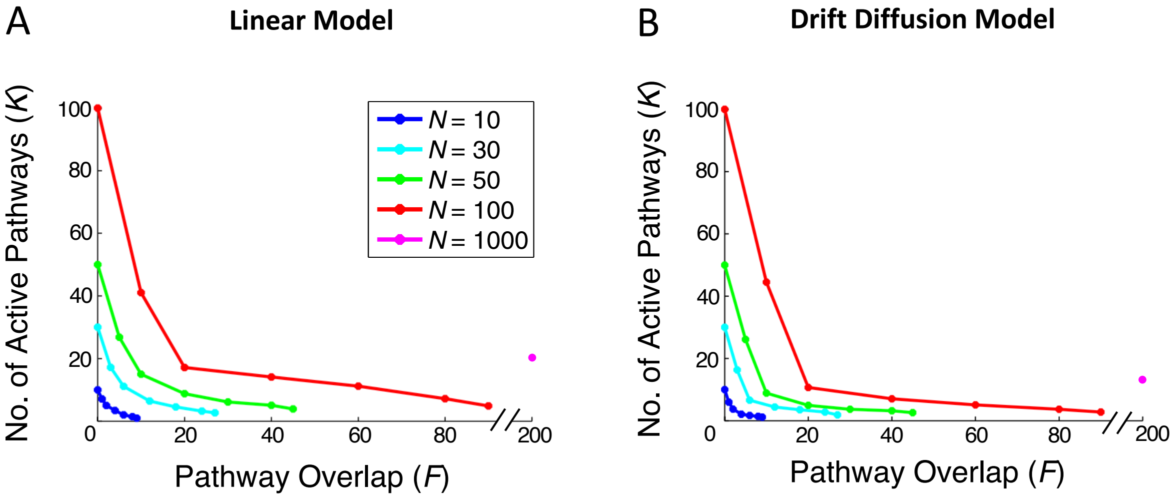

Figure 4. **Shared representations and asymptotic limits in multitasking capability.** Simulation results of Feng et al. (2014) showing the optimal number of simultaneously engaged processing pathways K (multitasking capability) as a function of overlap F between processing pathways (number of tasks sharing the same representation) in a neural network. Results rely on the assumption that tasks interfere 75% of the time if their processing pathways overlap. The optimal number of processing pathways was determined either by (A) minimizing the mean error over all output units in a linear model or by (B) maximizing aggregate reward rate over all tasks, each of which as implemented as a drift diffusion process.

are likely to be conservative (that is, produce an *under*estimate of the effects of interest — see Feng et al. (2014) for a discussion), the generality of the effects observed remained to be determined. Below, we describe the use of graph-theoretic methods to address this challenge. First, we use these methods to provide a formal characterization of multitasking capability as a function of the amount of shared representation and network size in simple linear networks, that also calls attention to two distinct forms of interference that can arise from shared representation. We then demonstrate how these methods can be used to predict both the overall multitasking capability of trained artificial neural networks, as well as behavioral markers of dual-task interference, such as the psychological refractory period (PRP) and task switch costs, from learned, distributed representations.

2.2 Graph-Theoretic Analyses

2.2.1 Definitions. In order to pursue a more rigorous analysis of the relationship between shared representations and the multitasking capability of a network, we first introduce more rigorous definitions of what we mean by a task, how performance is measured, and two distinct types of dependence that can arise between tasks that share representations. These definitions are stated in more rigorous, set-theoretic form in Lesnick, Musslick, Dey, and Cohen (2020).

Tasks and performance. For the purposes of this article, and in accord with the examples discussed above, we focus on simple types of "mapping" tasks that are defined by a set of associations of stimuli with responses. More specifically, we assume that: (1) inputs are structured by stimulus dimensions (e.g., color, shape, location, etc.); outputs are structured by response dimensions (e.g., verbal, left hand, right hand, etc.); (2) all of the stimulus features relevant to a particular task are drawn from the same stimulus dimension, and all of its responses are drawn from the same response dimension; (3) only a single stimulus or response can be represented within a given dimension at a given time. Under the General Discussion we consider how our results may apply to more complex forms of tasks (e.g., sequential tasks).

When performance of a task is evaluated, we assume that, for each trial, a feature is selected from the relevant stimulus dimension and activated in the stimulus input layer, and success is defined by the extent to which the correct unit within the relevant response dimension (that is, the one specified by the mapping that defines that task) is activated in the output layer (and no other output units are activated, within that dimension or any others). When parallel performance of two or more tasks (i.e., multitasking) is evaluated, a single feature is chosen independently for each task from each of the relevant stimulus dimensions, and success is defined by the extent to which all of the correct responses units are activated (and no others).[7] As noted above, our

[7] This precludes mappings that use the same stimulus dimension as independent tasks (e.g., color naming and color pointing in the example shown in Fig. 3). Accordingly, the simultaneous execution of such tasks is not considered as a genuine multitasking condition. This is because tasks are defined as

examples use "localist" representations of input and output features within each dimension, but the same principles apply to distributed representations (see Footnote 3). Based on these definitions, we identify two qualitatively distinct forms of dependence on shared representations that can give rise to conflict, and therefore demand control to avoid or resolve.[8]

Structural dependence. The most obvious way in which shared representations can introduce the risk of conflict is if two or more tasks involve the same response dimension, a classic example of which is the Stroop paradigm (see Fig. 2 and Fig. 5). This follows from the definitions above: If the stimuli for the two tasks are drawn independently from their respective stimulus dimensions, then they have the potential to require different responses within the same response dimension (e.g, verbal) and, according to assumption (3) above, both responses cannot be represented within that dimension at the same time. Furthermore, the likelihood of such interference grows rapidly with both the number of features in the relevant dimensions and the number of tasks to be performed, given the assumption that the stimuli for each task are chosen independently of one another.[9] Such dependence can also arise if tasks to be performed

mappings that draw *independently* from their respective stimulus dimensions. If they share the same stimulus dimension, then simultaneously drawing independently from that dimension would violate the assumption that only a single value of a dimension can be represented at a given time. Given this restriction, performing such tasks at the same time would be limited to conditions in which they always involved the same stimulus, and thus would amount to generating multiple responses to the same stimulus, which could simply be reformulated as a single task with a richer representation of responses.

[8] We use the term "dependence" rather than interference for several reasons: (1) It denotes situations in which inter-task interactions can arise (i.e., cross-talk), irrespective of their consequence (interference generally connotes destructive effects, such as conflict, whereas dependence can sometimes have constructive effects, such as facilitation or "super capacity"; see Townsend and Wenger (2004) and the General Discussion); (2) "dependence" is used in graph theory for similar purposes, where it corresponds to the concept of "independent sets" used below.

[9] Specifically, the likelihood of interference corresponds to the joint probability of selecting any stimuli across the tasks that are associated with different responses (e.g., an "incongruent" stimulus in the Stroop task).

in parallel converge on one or more internal dimensions of representation (e.g., phonological and orthographic in the dictation and word reading tasks of the Shaffer (1975) paradigm; see Fig. 1). We refer to these forms of dependence as *structural*, defined as the potential for interference that arises when two or more instructed tasks make common use of a dimension of representation. This is the type of interference on which the multiple resource theory was focused. However, there is another, indirect way in which dependence can arise in some network configurations.

Functional dependence. This refers to a form of dependence that arises indirectly when the tasks to be performed do not share any representations with one another, but the representations on which they depend can be recombined to form one or more other (currently irrelevant) tasks. As an example, consider a subset of the tasks in the extended Stroop paradigm: color naming, word reading, and word mapping. Fig. 5A shows the minimal basis set configuration for these tasks. Note that color naming and word mapping are not structurally dependent. Nevertheless, they cannot be performed simultaneously. This is because a combination of their stimulus and response dimensions (word stimuli and verbal responses) forms another task (word reading) that shares representations with one of the relevant tasks at the hidden layer (i.e., of words). As a consequence, activating word representations (in the service of word mapping) as well as the verbal output units (for color naming) inadvertently engages the word reading pathway, introducing the potential for interference with the color naming task. Thus, even though color naming and word mapping are not *structurally* dependent, they are *functionally* dependent.

The functional dependence mediated by word reading in this example can be averted if a separate set of representations for words is dedicated to the word mapping tasks, as shown in Fig. 5B.[10] Insofar as those are not associated with verbal responses,

[10] Alternatively, functional dependence between color naming and word mapping can be avoided by configuring the word mapping task as a pathway from word stimuli to a different, existing set of representations in the hidden layer (e.g., for locations). However, as a result, the word mapping task would then be structurally dependent on any task relying on representations for that dimension (e.g. location mapping).

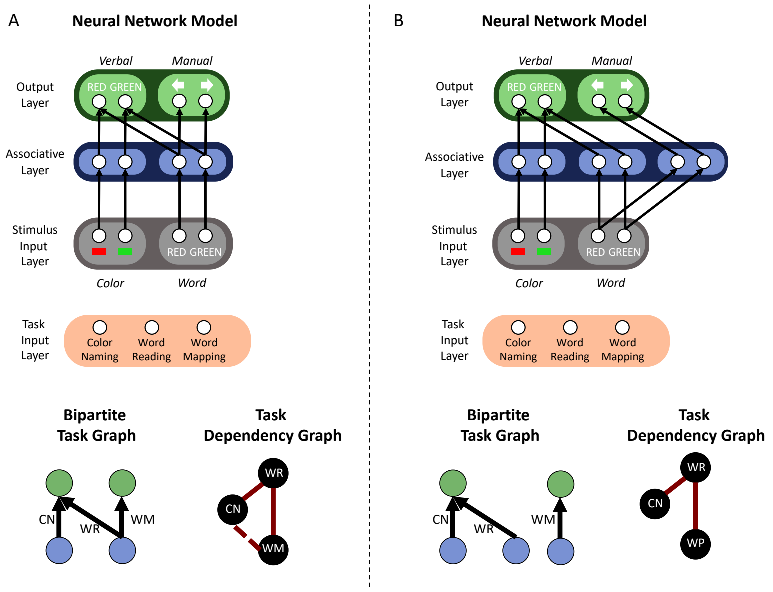

Figure 5. **Structural and functional dependence in the extended Stroop model.** Examples of networks exhibiting each form of dependence among tasks in the extended Stroop paradigm, and their graph-theoretic representation. Each network implements four tasks: color naming (CN), word reading (WR) and mapping a word to a button press (WM). As discussed in the text, color naming and word reading are structurally dependent, since both share the same response dimension. However, color naming and word mapping can be either functionally dependent or fully independent, based on the network configuration – that is, whether a minimal basis set or tensor product representations are used for the word reading and word mapping, as shown in the Neural Network Models in Panels A and B. *(A) Minimal basis set representation for words.* The word mapping task shares a representation for words with the word reading task at the hidden layer, which introduces functional dependence between it and the color naming task. *(B) Tensor product representation for words.* Word Mapping relies on a separate set of representations for words in the hidden layer, rendering color naming and word mapping functionally independent (see text for explanation). Each configuration has a corresponding *bipartite task graph* (lower part of each panel), with nodes representing stimulus and response dimensions, and edges representing the tasks (i.e., the mappings from features in a given stimulus dimension to corresponding responses in the response dimension that define that task). The corresponding *dependency graph* represents the relationship between tasks, with nodes now corresponding to tasks, and edges indicating tasks are dependent on (i.e., interact with) one another. A solid line in the dependency graph indicates structural dependence between two tasks whereas a dashed line indicates functional dependence. The maximum independent set (MIS) of this graph corresponds to the multitasking capability of the network (see text for explanation). The MIS of the dependency graph shown in (A) is 1 whereas the MIS of the graph shown in (B) is 2.

activating them to perform the word mapping task would not engage the word reading task, allowing the color naming task to be performed in parallel without risk of interference. This corresponds to the tensor product configuration for those two tasks. These two ways of representing the word mapping task — using representations for words that are shared with or separate from word reading — provide an example of how the minimal basis set may be efficient to learn (for performing a novel task), but at the cost of the multitasking capability (i.e., ability to multitask) afforded by the tensor product, observations for which we provide empirical support in Part II of this article.

2.2.2 Bipartite and Dependency Graphs. To analyze how structural and functional dependence scale as a function of the prevalence of shared representations and size of a network, we define a graph-theoretic formalism of the relationship among the tasks implemented in a network. This involves two graph representations (shown at the bottom of each panel in Fig. 5). For clarify of exposition, we begin by considering only three-layered networks of the sort shown in the examples thus far, but then go on to consider the case of multilayered networks.

Bipartite graph. This is a simplified representation of the sorts of three-layered networks used in the examples above, that focuses on the hidden and output layers. This simplification is justified by observing that, for the full range of network configurations for a given set of tasks, the hidden and output layers are sufficient to describe the factors of interest: whether, at the hidden layer, representations are shared between tasks with each projecting to all response dimensions (as in the extreme case of the minimal basis set configuration); or whether a separate subset of hidden layer representations is dedicated to each task (i.e., to each pairing of stimulus and response dimension, as in the extreme of the tensor product configuration). Thus, a given network configuration can be represented as directed bipartite graph $G_B = (I, O, T)$ (see Appendix A for an overview of relevant graph-theoretic terms), in which each input node I represents a subset of hidden representations (corresponding to associative dimensions in the original network)[11], each output node O represents a response

[11] As noted above, for the minimal basis set configuration, there is one input node of the bipartite

dimension, and edges between them represent the tasks (see the left bottom of each panel in Fig. 5). The bipartite graph can be used to formalize the distinction between structural and functional dependence described above. Two tasks are considered to be *structurally* dependent if their edges share either an input node or an output node (e.g., the color naming task and the word reading task in Fig. 5 both share the same output node and are thus considered to be structurally dependent). In contrast, two tasks are considered to be *functionally* dependent if they are not structurally dependent, but an edge (a third task) connects the input node of one task to the output node of the other (e.g. an edge representing the word reading task connects the color naming and word mapping tasks in the bipartite task graph in Fig. 5A).

Dependency Graph. Using the bipartite graph described above, a dependency graph can be constructed that directly expresses relationships between tasks. This is constructed by assigning each edge of the original graph G_B to a node in the dependency graph G_D. Thus, each node in G_D represents a task in G_B (and in the original network). Edges are assigned between any two nodes in G_D representing tasks in G_B that are either structurally or functionally dependent (see the right bottom of each panel in Fig. 5), as defined above. For simplicity, we assume that either form of dependence introduces a risk of interference that precludes those two tasks from safely being executed in parallel. This relies on the assumption that two different tasks are unlikely to process congruent pieces of information (see Footnote 9). Thus, the dependency graph G_D can be used to determine which tasks in the original network can be executed safely in parallel. In the analyses described below, we exploit this to determine the maximum number of tasks that a given network can execute in parallel; that is, its multitasking capability.

2.2.3 Analysis of Multitasking Capability. The dependency graph G_D can be used to analyze the multitasking capability of a network, however this poses

graph for each stimulus dimension represented in the hidden layer of the original network, whereas for the tensor product configuration there are as many input nodes in the bipartite graph as there are distinct task-specific sets of hidden layer representations in the original network.

challenges that we consider and address below.

Maximum independent set. The definitive way to determine the multitasking capability of a network is to identify the largest set of nodes (tasks) in G_D that do not share any edges (i.e., that are not dependent on one another). This is known as the *maximum independent set* (MIS) of a graph (Godsil & Royle, 2001). Thus, determining the MIS of G_D provides a general means of examining how factors such as shared representation (i.e., task dependencies) and network size influence its multitasking capability (Musslick et al., 2016), corresponding to the factors that were examined numerically for particular networks in Feng et al. (2014). However, there are practical constraints on doing so. If the bipartite graph G_B representing the network contains only structural interference, or only structural interference is considered when constructing the dependency graph G_D, then G_D is known as the *line graph* of G_B, and calculating its MIS is a well formed and tractable problem (D. B. West et al., 2001). It is equivalent to the matching problem, and can be computed by computationally efficient algorithms (Hopcroft & Karp, 1973). However, when there are functional dependencies in G_B and they are included in G_D, then the latter is known as the *square of the line graph* of G_B, and calculating its MIS is equivalent to solving an *induced matching problem* (Cameron, 1989). This is known to be an “NP hard” problem, the complexity of which scales roughly factorially with the size of the graph, and thus quickly becomes computationally intractable (Berman & Fürer, 1994; Tarjan & Trojanowski, 1977). Since the latter is required to fully characterize the multitasking capability of a network, doing so requires that constraints be placed on the problem. Below, we address this issue (and how it relates to constraints on cognitive control), exploring various ways of constraining the problem for analysis, and then examining their ability to generalize more broadly.

Distribution complexity. One set of measures of the bipartite graph G_B that can be used to quantify the prevalence of shared representations in the network are its *out-degree* and *in-degree*. The out-degree is the “fan out” of an input node; that is, the number of response dimensions with which a stimulus dimension is associated.

Conversely, the in-degree is the "fan in" of an output node, specifying the number of stimulus dimensions that map convergently to the corresponding response dimension. As we show below, the multitasking capability of a network depends both on the mean of these measures of degree across all input and output nodes, as well as the *distribution* of their values over the corresponding sets of nodes. Characterizing these factors provides a basis for simplifications that can help make the enumeration of all possible graphs tractable. Toward this end, we introduce distribution complexity $DC_{in,out}$ as a measure of homogeneity in degree distribution in the bipartite task graph G_B. The distribution complexity of incoming edges of the output nodes DC_{in} is defined as:

$$DC_{in} = -\sum_{i=1}^{N} \left(\left(\frac{d_{in}^{i}}{\sum_{k=1}^{N} d_{in}^{k}} \right) \log_2 \left(\frac{d_{in}^{i}}{\sum_{k=1}^{N} d_{in}^{k}} \right) \right). \tag{1}$$

The distribution complexity for outgoing edges DC_{out} is defined in an analogous manner. The equation above can be read as a measure of the entropy over the sharing of representations across all response dimensions. For a fixed network, DC_{in} is maximized when edges are uniformly distributed among the output nodes and, as we will demonstrate, leads to lower values of multitasking capability. For example, Fig. 6 illustrates two bipartite graphs, both of which have output nodes with the same out-degree $d_{out} = 2$, but one of which has low distribution complexity (most tasks converge on the same output node), and the other of which has high distribution complexity (tasks are uniformly distributed among the output nodes).

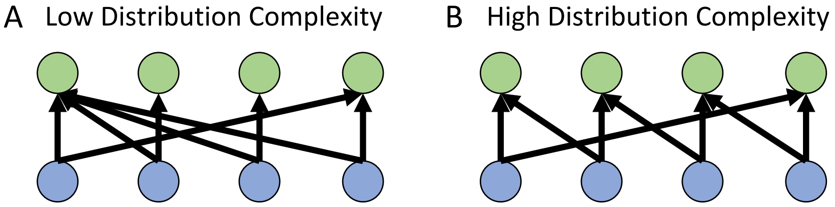

Figure 6. **Distribution complexity.** Two bipartite graphs with output nodes that have the same out-degree ($d_{out} = 2$): (A) low distribution complexity ($DC_{in} = 1.75$); (B) high distribution complexity ($DC_{in} = 2$).

To investigate the effect of shared representations and distribution complexity on multitasking capability, we considered networks with N stimulus and N response dimensions. We fixed the out-degrees of each input node such that $d^i_{out} = S$ where S is a proxy for the number of tasks that rely on the same representation in the hidden layer of the network (or, equivalently, input layer of the bipartite graph). We constrained the in-degree of the output nodes to be uniform (i.e. $d^i_{in} = S, \forall i \in \mathcal{V}_{in}$), which made it tractable to enumerate all possible networks of a given size N and shared representation S. For each enumerated network, we computed its multitasking capability by computing the MIS of the associated dependency graph. Fig. 7A-D summarizes the results for networks of size $N = 5, 6, 7$ and 8, respectively. The results show that multitasking capability (averaged over all possible network configurations with a given size N and fixed out-degree d^i_{out}) dropped precipitously with the number of tasks sharing the same stimulus representation S. This was observed over a wide range of distribution complexities, from maximum (red lines, corresponding to values used in Feng et al., 2014), to the average value (black lines). Thus, the observations based on the numerical analyses of a particular set of networks reported in Feng et al. (2014) appear to generalize over a much broader range of possible networks. Nevertheless, it is of interest to observe that, at the extremes, distribution complexity did impact multitasking capability, with a minimum in DC_{in} diminishing shared representation between tasks and thus maximizing multitasking capability. For example, multitasking capability is maximized when all sharing in the network occurs on a single output component (shown in blue; also see Fig. 6A). In contrast, multitasking capability is minimized when the sharing of representations is distributed more uniformly over the network (maximum DC_{in}, shown in red; also see Fig. 6B).

One might intuitively guess that the multitasking capability of a system is largely dependent on the size of the network (i.e. the number of stimulus and response dimensions). The computational intractability of enumerating all possible networks, and limits of currently available computational power, preclude an exact analysis of networks beyond size $N = 8$.[12] However, by constraining enumeration to networks with

[12] Even for a network of size $N = 8$ with out-degree $d^i_{out} = 4$, the number of possible network

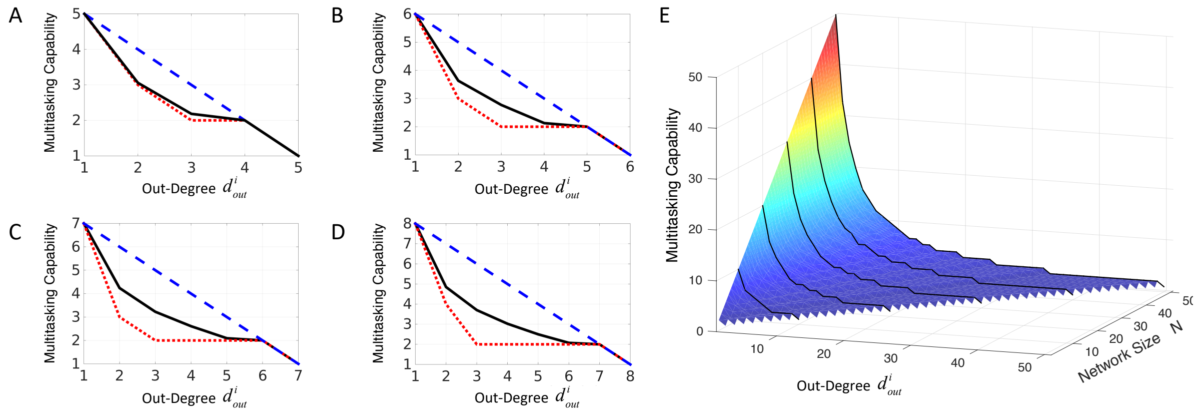

Figure 7. **Effect of distributional complexity on multitasking capability.** Graph-theoretic analysis of multitasking capability. Panels (A)-(D) show variation in multitasking capability (measured as MIS of the dependency graph for networks of size 5, 6, 7 and 8) as a function of out-degree d^i_{out} for all network configurations, corresponding to the average value of distribution complexity (black line in panels (A)-(D)). Panel (E) shows multitasking capability (higher values correspond to warmer colors) with maximum value of distribution complexity for networks of sizes 1-50 and a corresponding range of out-degree d^i_{out}.

maximum DC_{in} analyses can be extended to much larger networks. For example, Fig. 7E shows results for networks up to size 50, which exhibited the same qualitative effects (see Fig. 7A-D). In particular, they reaffirm the observation that even modest amounts of shared representation impose dramatic constraints on multitasking capability, virtually irrespective of network size. Although Fig. 7E shows the effect when processes were distributed uniformly over the network, the results shown in Fig. 7A-D indicate that the dramatically sublinear scaling of multitasking capability with network size prevailed for a wide range of distribution complexities.

These results are consistent with those of similar, but complementary approaches to computing the multitasking capability of network architectures as a function of representational sharing (e.g., Petri et al., 2020; Alon et al., 2017). Together with those of Feng et al. (2014), they strengthen the conjecture that, for control-dependent processes — that is, those involving shared representations that require control for disambiguation — the number that can be concurrently executed is dramatically

configurations exceeds 2.25 trillion.

limited in a manner that is relatively insensitive to network size.

Effective multitasking capability. The computation of MIS described above provides a theoretical maximum for the multitasking capability of a network. In reality, the number of control-dependent tasks that a network can be expected to carry out in a given setting is likely to be considerably lower. This is because the MIS refers to one or more *particular* sets of tasks that are independent of one another. However, even if there is more than one such set, the network can only realize the multitasking capability indicated by the MIS when those particular tasks are available to be performed. The likelihood of this occurring is of course determined by other factors, such as the affordances of each task (i.e., the current availability of the stimuli and feasibility of the responses), and the motivation for performing them (i.e., their current value to the agent). It is easy to see that, even with liberal assignments of probabilities to these individual factors, their joint probability diminishes quickly with the size of the MIS, its proportion to the overall size of the network (i.e., total number of tasks it can perform) and the scope of the environment. A more general characterization of effective multitasking capability would take account not only of the MIS, but all smaller independent sets of tasks. One such calculation, that considers smaller sets of tasks sampled uniformly at random, strongly suggests that, like the MIS, the effective multitasking capability of a network decreases dramatically with the extent of shared representations, and grows sub-linearly with the size of the network (Petri et al., 2020).

Multitasking capability and network depth. The analyses described above all pertain to three-layered networks, with a single hidden layer represented by the input nodes of the bipartite graph. A natural question is how multitasking capability is impacted by the number of layers (i.e., "depth") of a network — a factor that is of obvious importance to understanding both the brain, as well as artificial systems ones that have become increasingly important in machine learning. For example, one advantage of deep architectures is that they are more economical in expressing real functions (Goodfellow, Bengio, & Courville, 2016). A greater number of layers in a network allows it to encode a larger set of mappings between a given pair of input and output nodes. Thus, the number of meaningful tasks that one might be able to perform may increase with the number of layers. However, a greater number of layers in a network may also increase the opportunity for cross-talk. To assess the influence of these factors, we generalized the graph-theoretic analysis described above for bipartite graphs, to consider networks with multiple layers. For simplicity, we considered networks with r disjoint layers, in which every layer was an independent set (i.e., there were no connections between nodes within the same layer), and all of which had the same size N. In such graphs, a task corresponded to a path from a node in the input layer to one in the output layer.[13] The definitions of structural and functional independence can be extended by direct analogy to the bipartite case: A pair of tasks are structurally dependent if their paths share a node at any layer in the network; and a pair of tasks are functionally dependent if they are structurally independent but are connected by an edge (that is, there is at least one edge that connects a node of one task to a node of the other). As in the bipartite case, we sought to determine the multitasking capability of the network, that is, the largest set of tasks that were both structurally and functionally independent. Note that, in these networks, the multitasking capability can only be as large as the smallest multitasking capability between any two layers.

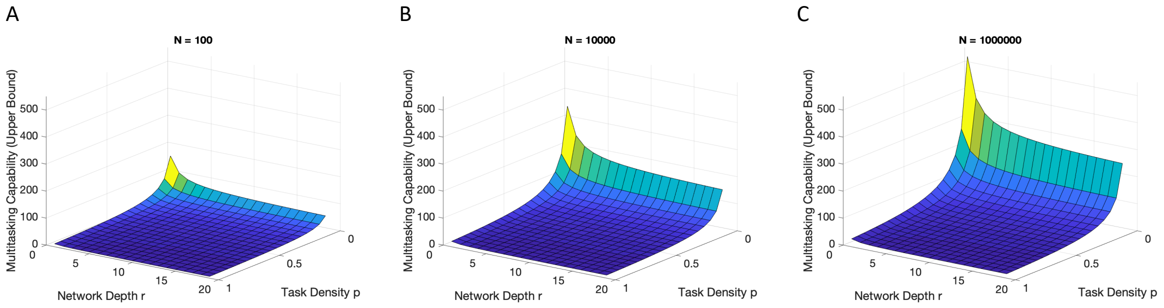

Figure 8. **Effects of network depth.** Upper bound of multitasking capability as a function of task density (the probability of an edge p between any two layers) and network depth (the number of layers r). The number of nodes per layer was varied across networks: (A) 100, (B) 10,000 or (C) 1,000,000 nodes per layer.

[13] Note that in contrast to the case of three-layered networks, there may be *multiple* paths between an input node and an output node, which could correspond to multiple realizations of the same task (i.e., using different intermediate representations).

In Appendix B we show, using mathematical analysis, that the constraints on multitasking capability are robust to network depth. The results are shown in Fig. 8. It should be noted, however, that the probabilistic manipulation of task density (i.e., edge probability) used in these analyses is not formally equivalent to directly manipulating degree of shared representation. That is, the results are limited to network architectures that are defined by randomly connecting layers. However, recent work using similar graph-analytic methods that control for both the number and distribution of tasks in the network have generated similar results (Alon et al., 2017). That work, together with the results presented here, suggest that shared representations have *even greater* constraining effects on multitasking as the depth of a network increases.

2.3 Toward a Mechanistic Account of Constraints on Control-Dependent Processing: Shared Representation, Conflict and Persistence

In the previous section we introduced graph-theoretic methods for analyzing the influence of shared representation on multitasking capability. These analyses relied on a number of simplifying assumptions. First, they assumed that tasks either share or don't share a set of representations. However, many of the most important contributions that neural network models have made to psychological research have relied specifically on representations of concepts that are distributed over many processing units that allow for *graded degrees* of sharing (Hinton et al., 1986; Kriegeskorte, Mur, & Bandettini, 2008; McClelland, Rumelhart, Group, et al., 1986; T. T. Rogers & McClelland, 2004; A. M. Saxe, McClelland, & Ganguli, 2019; Yamins et al., 2014); and neuroimaging studies have provided strong support for this in the brain (Albers, Kok, Toni, Dijkerman, & De Lange, 2013; Kosslyn et al., 1999; Notebaert, Gevers, Verguts, & Fias, 2006; Salamoura & Williams, 2007; Decety & Sommerville, 2003). It remains to be shown whether and how the graph-theoretic formalisms described above can be applied to such networks. Second, all of the networks were all pre-configured, either deterministically, or connections were assigned according to general statistical constraints that did not reflect anything specific or meaningful about natural task environments. However, networks that learn representations through experience have played a critical role in explaining human cognitive function (Botvinick et al., 2001; Brown, Reynolds, & Braver, 2007; J. D. Cohen et al., 1990; Gilbert & Shallice, 2002; Herd et al., 2014; McClelland et al., 1986; O'Reilly & Frank, 2006; T. T. Rogers & McClelland, 2004; A. M. Saxe et al., 2019), and have become a mainstay of research in artificial intelligence (Goodfellow et al., 2016; Schmidhuber, 2015). Finally, the analyses focused exclusively on interference arising from the *simultaneous* execution of two or more tasks (see Footnote 7). They did not address performance costs known to be associated with the processing of multiple tasks even when they are executed *in sequence*, such as the psychological refractory period (Telford, 1931) and switch costs (Allport et al., 1994); nor do they address the continuum from pure parallelism, through rapid task switching, to pure sequential processing that has been described by others (Fischer & Plessow, 2015; Salvucci et al., 2009; Townsend & Wenger, 2004). Below, we present the results of simulations studies showing how the effects associated with all of these factors can be explained in terms of the sharing of representations, by considering the influence of three graded properties that are intrinsic to neural network architectures: the similarity of representations, the strength of connections, and the persistence characteristics of patterns of activity.

In Simulation Study 1, we demonstrate that the graph-theoretic methods described above can be used to predict the multitasking performance from distributed representations of tasks in trained neural network models, by quantifying the degree of representation sharing in terms of the similarity between patterns of activity associated with each task. One motivation for this is the potential use of such methods for analyzing brain imaging data, to predict multitasking performance from patterns of activity associated with individual tasks. In Simulation Study 2, we investigate how degree of representation sharing interacts with connection strength (manipulated by training) to produce conflict, and evaluate its effects both on multitasking accuracy as well as established measures of reaction time distributions that have been used to infer parallelism of multitask processing from human behavioral data. Finally, in Simulation

Study 3, we show that interference effects arising from the interaction between representation sharing and the persistence characteristics of representations in neural networks can explain costs associated with the sequential performance of multiple tasks (such as the PRP and task switch costs). Furthermore, we discuss how these interactions can be used to define a continuum from pure parallelism, through rapid task switching, to pure sequential processing.

2.3.1 Neural Network Model of Multitasking Performance. We begin by defining the general network architecture and the task environment used to simulate both concurrent and sequential multitasking performance. We then describe the network's processing and training procedure, as well as performance metrics used across simulations.

Architecture. As in the examples above, the models used here were comprised of three layers of processing units: an input layer with two partitions, one of which represented the current stimulus and projected to a hidden layer, and another that encoded the current task and projected to both the hidden and output layers; a hidden layer (100 units) that projected to the output layer; and an output layer that represented the network's response. Stimulus input units were grouped by the stimulus dimensions relevant to performing each task, and used a one-hot encoding (i.e., a single unit was used to represent each stimulus, with the current stimulus clamped to 1 and all others clamped to 0). The number of units in the input and output layer varied across simulations studies, as determined by the corresponding task environment. Fig. 9 illustrates a network with three stimulus dimensions (each with three features) and five tasks. The task input units used a similar one-hot encoding, with one unit used to represent each task. Output units were grouped by response dimensions, and trained (see below) using a one-hot encoding for each response within a dimension.

Processing. The network was instructed to perform a given task by specifying the current stimulus and task to be performed in the respective partitions of the input layer. These stimulus and task input values were multiplied by a matrix of connection weights from each partition of the input layer to a shared hidden layer, and then passed

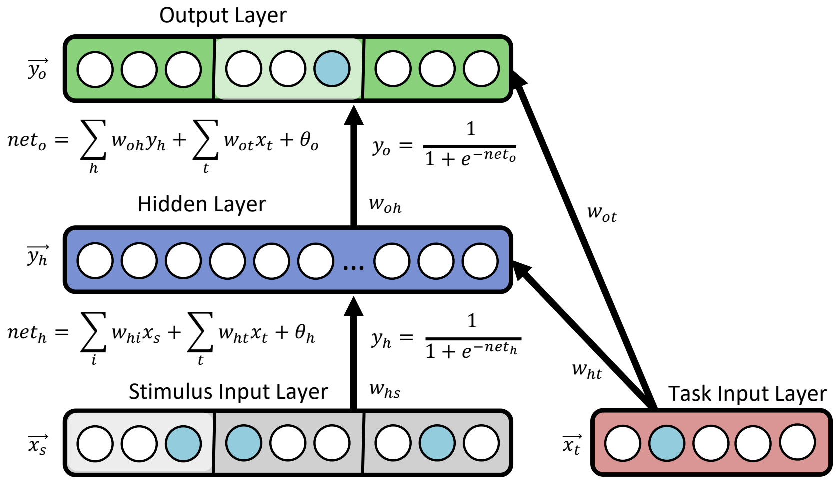
Output Layer
$\vec{y_o}$
$net_o = \sum_h w_{oh}y_h + \sum_t w_{ot}x_t + \theta_o$
$y_o = \frac{1}{1+e^{-net_o}}$
Hidden Layer
w_{oh}
w_{ot}
$\vec{y_h}$
...
$net_h = \sum_i w_{hi}x_s + \sum_t w_{ht}x_t + \theta_h$
$y_h = \frac{1}{1+e^{-net_h}}$
w_{ht}
Stimulus Input Layer
w_{hs}
Task Input Layer
$\vec{x_s}$
$\vec{x_t}$

Figure 9. **Neural network used for simulations of multitasking.** The input layer was composed of a stimulus vector $\vec{x_s}$ and a task vector $\vec{x_t}$. The activity of each element in the hidden layer $y_h \in \vec{y_h}$ was determined by all elements x_s and x_t and their respective weights w_{hs} and w_{ht} to y_h. Similarly, the activity of each output unit $y_o \in \vec{y_o}$ was determined by all elements y_h and x_t and their respective weights w_{oh} and w_{ot} to y_o. A fixed bias of $\theta = -2$ was added to the net input of all units y_h and y_o, to implement the assumption that units are inhibited at rest. Thus, without additional input from the task layer, units are relatively insensitive to information from the previous layer. Additional input from the task layer puts these units on a more sensitive part of their activation function, making them more susceptible to incoming information from preceding layers (see J. D. Cohen et al., 1990). Filled input and output units (circles) correspond to unit values of > 0, and illustrate an example stimulus and task input pattern with its respective response pattern: The task indicated by the activated unit in the task layer requires the network to map the vector of values in the three stimulus input units in the first stimulus dimension (shaded in light grey) to one out of the three units in the second response dimension (also shaded in light grey).

through a logistic function to determine the pattern of activity over the units in the hidden layer. This pattern was then used, together with the set of direct projections from the task input layer to the output layer, to determine the pattern of activity over the latter. The activation values of units in the hidden and output layer were computed as a function of their net input. The net input net_i of unit i in a given processing (hidden or output) layer was calculated based on the connectivity and the activation from preceding layers as

$$net_i = \sum_j w_{ij} x_j - \theta \tag{2}$$

where x_j is the activity value of the sending unit, w_{ij} is the projection weight from sending unit j and $\theta = -2$ is a constant negative bias. The net input of each unit in the hidden and output layers was then then passed through a logistic function to determine its activity y_i

$$y_i = \frac{1}{1 + e^{-net_i}} \tag{3}$$

The response within a given response dimension of the network was determined by a leaky competitive accumulator (LCA, Usher & McClelland, 2001) layer, implementing the assumption that the network could only provide one response per response dimension (e.g. the network cannot say "RED" and "GREEN" at the same time).[14] One LCA layer was assigned to each response dimension k, which was comprised of a set of units r_i that received as their input the activity of corresponding units in that response dimension. The winning response in each dimension was determined by the accumulation of activity by each LCA unit and the competition among them, the dynamics of which were governed by

$$dr_i = [y_o - \lambda r_i + \alpha f(r_i) - \beta \sum_{j \neq i} f(r_j)] \frac{dt}{\tau} + \xi_i \sqrt{\frac{dt}{\tau}} \tag{4}$$

where y_o is the activity of the corresponding response unit in response dimension k, λ is the decay rate of r_i, α is the recurrent excitation weight of r_i, β is the inhibition weight between LCA units, τ is the rate constant, and ξ is noise sampled from a Gaussian distribution with zero mean and standard deviation σ. The activity of each

[14] This one-winner-take-all constraint is in agreement with our formal definition of a task in Lesnick et al. (2020). While this constraint was not explicitly imposed on other layers of the network (since they did not include recurrent connections), nevertheless it could arise through the feedforward inhibition acquired through learning. We return to this issue in the General Discussion.

LCA response unit was lower bounded by zero with a threshold such that $f(r_i) = r_i$ for $r_i \geq 0$ and $f(r_i) = 0$ for $r < 0$. The response for dimension k was determined by the unit within the corresponding LCA layer the activity $f(r_i)$ of which first reached threshold z. The accuracy for each response dimension k corresponded to the probability of generating the correct response for that dimension $P(\text{correct})_k$ across 100 simulations of the LCA, and the reaction time RT_k for that dimension was the average number of time steps t required for the response to reach threshold, plus a fixed non-decision time of $T_0 = 0.15s$. The following parameter values were used for all reported simulations: $\lambda = 0.4$, $\alpha = 0.2$, $\beta = 0.2$, and $\sigma = 0.2$; z for each LCA layer was chosen as the threshold that maximized reward rate $P(\text{correct})_k/(ITI + RT_k)$ for that dimension, where ITI corresponds to an inter-trial interval of 0.5s.

Task environment. Each task was comprised of a pair of input and output vectors. The input vector in each pair was composed of subvectors specifying the stimulus and task, and the associated output vector specified the correct response for the stimulus for each task. All of the stimuli for a given task were drawn from the same stimulus dimension and all of the responses for that task were drawn from the same response dimension. Each stimulus was associated with a single, unique response, a task was comprised of all of the unique pairs of stimulus-response vectors for its specified stimulus and response dimensions, and there was one task for each unique combination of stimulus and response dimensions. These implementations conform to the formal definition of a task described in Lesnick et al. (2020). The number of stimulus and response dimensions varied across simulation studies. In all tasks, the stimulus dimension and response dimension each had three features (i.e., stimuli and responses, respectively).

Training. Networks were initialized with a set of small random weights[15] and then trained using the backpropagation algorithm (Linnainmaa, 1970; Rumelhart, Hinton, & Williams, 1986; Werbos, 1982) to produce the task-specified response for each stimulus in each task, while suppressing all other responses (both within the task-relevant

[15] We initialized the networks with small random weights to facilitate convergence of learning.

response dimension, and all task-irrelevant response dimensions). The network was trained in epochs, with each epoch sampling all training patterns in random order. The error term used for training was the mean squared error (MSE) of the pattern of activities in the output layer with respect to the correct (task-determined) output pattern. The weights of the network were adjusted[16] with a learning rate of 0.3 after presenting each training pattern within an epoch (online training) until the network reached an MSE of 0.001.

Measures of single and multitask performance. The accuracy of the network on a single task was determined by the probability of responding correctly in the task-relevant response dimension, averaged across all stimuli for that task. Multitasking accuracy for a given set of tasks was determined by the average probability of responding correctly across all task-relevant response dimensions, averaged across all stimuli. Unless otherwise specified, we assessed multitasking performance only for incongruent stimuli.[17] Since all tasks in a multitaskable set are structurally independent (see below), stimulus incongruence is identified with respect to irrelevant tasks that mediated functional interference. Thus, incongruent stimuli were defined as configurations of stimulus features for which the correct response in at least one response dimension was different for at least two tasks that mapped to that response dimension. Conversely, congruent stimuli were defined as configurations of stimulus features for which the correct responses in all task-relevant response dimensions were the same (see Fig. 10).

Multitasking sets. We measured multitasking performance on “multitaskable” sets

[16] Bias weights remain fixed at their initial value of -2.

[17] Testing the network on only incongruent stimuli corresponds to an assumption made by the graph-theoretic analysis above, that cross-talk always results in response conflict. This is not unreasonable, as congruent stimuli are generally unlikely to be sampled from a uniform distribution of stimuli, given that the likelihood of a congruent stimulus decreases with the number of stimulus dimensions as well as with the number of features per stimulus dimension (Feng et al., 2014). Thus, performance on incongruent stimuli is likely to be reasonably representative of behavior in rich task environments.

of tasks on which a network was trained. All tasks within a multitaskable set were structurally independent; that is, each task in the set had input and output dimensions that were distinct from all of the others in the set. The requirement of distinct input dimensions for the tasks in each set satisfies our definition of a task (see Lesnick et al., 2020); the requirement for distinct output dimensions insures that it was possible in principle to perform the multitask over all stimuli (for example, color naming and word reading would not constitute a legitimate multitasking combination since it is not possible to execute both tasks simultaneously over all possible stimuli, viz. incongruent ones).

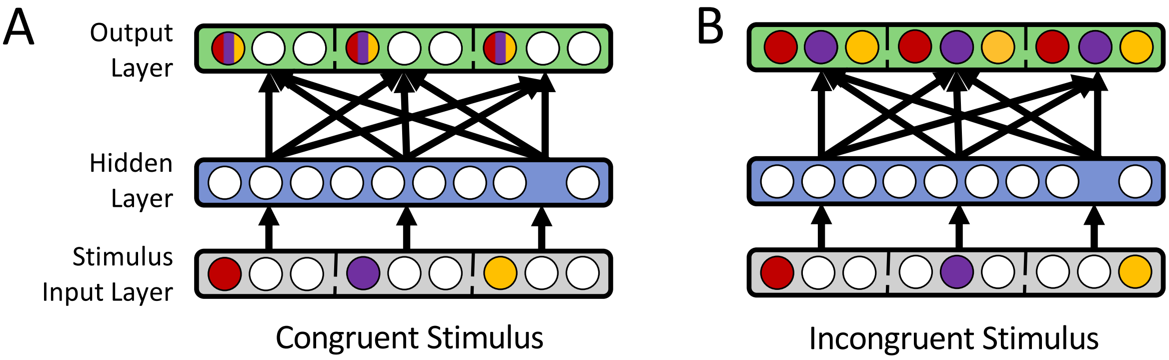

Figure 10. **Congruent and incongruent stimuli.** The network in both panels consists of an input layer, a hidden layer and an output layer (task input layer is not shown). The stimulus input and output layers are grouped into three stimulus and response dimensions, respectively. A task is defined as a mapping from one of three feature units in a given stimulus dimension to one of three output units in a corresponding response dimension. Colored circles in the stimulus input layer indicate the active feature in each stimulus dimension. Colored circles in the output layer indicate the correct response as determined by the task that requires mapping the stimulus feature of the same color. (A) Congruent stimuli require the same response in a given response dimension, irrespective of the task involving that response dimension the network is asked to perform. (B) Incongruent stimuli require a different response in a given response dimension, depending on the task the network is asked to perform.

2.3.2 Simulation Study 1: Predicting Multitasking Capability from Single Task Representations.

In the previous section we introduced graph-theoretic analyses to investigate factors affecting the multitasking capability in simplified network structures. These analyses were based on the assumption that shared representations can induce functional dependence between tasks, constraining the number of tasks a network can perform at the same time. Here, we examine the extent to which these analyses can be applied to more complex models (of biological agents and/or artificial systems) in which tasks representations are learned and distributed across multiple processing units. We describe how neural representations of individual tasks can be used to generate predictions about how many and which combinations of tasks a network can perform in parallel (a space of possibilities that grows combinatorially with the number of tasks, and thus quickly becomes intractable to direct empirical inquiry), based on measurements of single task performance (that grows only linearly in the number of tasks). The purpose of these analyses is to confirm that the constraining effect of shared representations generalizes to more complex network architectures with distributed representations, and to validate the application to such networks of diagnostic tools for assessing multitasking capabilities using measurements made in single task performance – that is, on amounts of data that would practical to acquire in empirical settings.

To assess the accuracy with which the graph-theoretic analyses described above predict the multitasking capability in more complex neural networks, we compared predictions of multitasking performance made by task dependency graphs extracted from 20 separately trained networks with the numerically simulated multitasking performance of those networks. We did so by extracting a bipartite graph from each trained network (using methods described below), and from that a dependency graph. We then used the dependency graphs to make predictions about the networks' multitasking capability, as well as performance of multitasking sets as a function of the number of dependencies between tasks in a set. We first describe specifics of the network architecture and training environment used for these simulations, as well as the procedure for extracting dependency graphs based on learned task representations, followed by a comparison of predictions and results.

Network architecture and processing. The networks in these simulations used five stimulus and five response dimensions ($N = 5$), each of which had three features (i.e., stimuli and responses, respectively). Thus, they supported a total of 25 possible tasks, 1545 multitasking conditions and 243 possible stimulus (and corresponding response) patterns per task (including both task-relevant and task-irrelevant features).

Task environment. As described above, a task was defined as a mapping from the three stimulus features of a task's stimulus dimension to the three corresponding output units of its response dimension, such that only one of the three relevant output units was permitted to be active for a given stimulus input unit. Each network was trained on a different subset of ten randomly sampled tasks (an example training environment is shown in Fig. 11A). Tasks were sampled subject to the constraint that each stimulus dimension and each response dimension was associated with at least one task.

Generating bipartite and dependency graphs from task representations. Network analyses focused on representations (patterns of activity) in the hidden and output layers, insofar as these correspond to data that can be acquired with neuroimaging techniques. Analyses characterized the representations for each task, and how they compared across tasks. We used these measures to construct bipartite and dependency graphs for each network, from which its predicted multitasking capability was computed, and tested against the empirically measured multitasking performance of the network.

The representations (patterns of activity) associated with each task that were learned during training were characterized by calculating, for each unit in the hidden and output layers, the mean of its activity over all of the stimuli for that task.[18] This mean activity pattern at each layer for each task was correlated with the one for each other task to yield a task similarity matrix that was examined separately for the hidden and output layers of the network. Fig. 11B provides an example of such similarity matrices. These were used to assess the extent to which different tasks relied on shared or separated representations within the hidden and output layers of the network, which was used, in turn, to construct a bipartite graph (shown in Fig. 11C). The

[18] A formally equivalent analysis could be carried out using the weight matrix of the network. Here we focus on patterns of activity, as these may serve as useful predictors for patterns of activity observed in empirical data, such as functional magnetic resonance imaging (fMRI) and/or neuronal recordings.

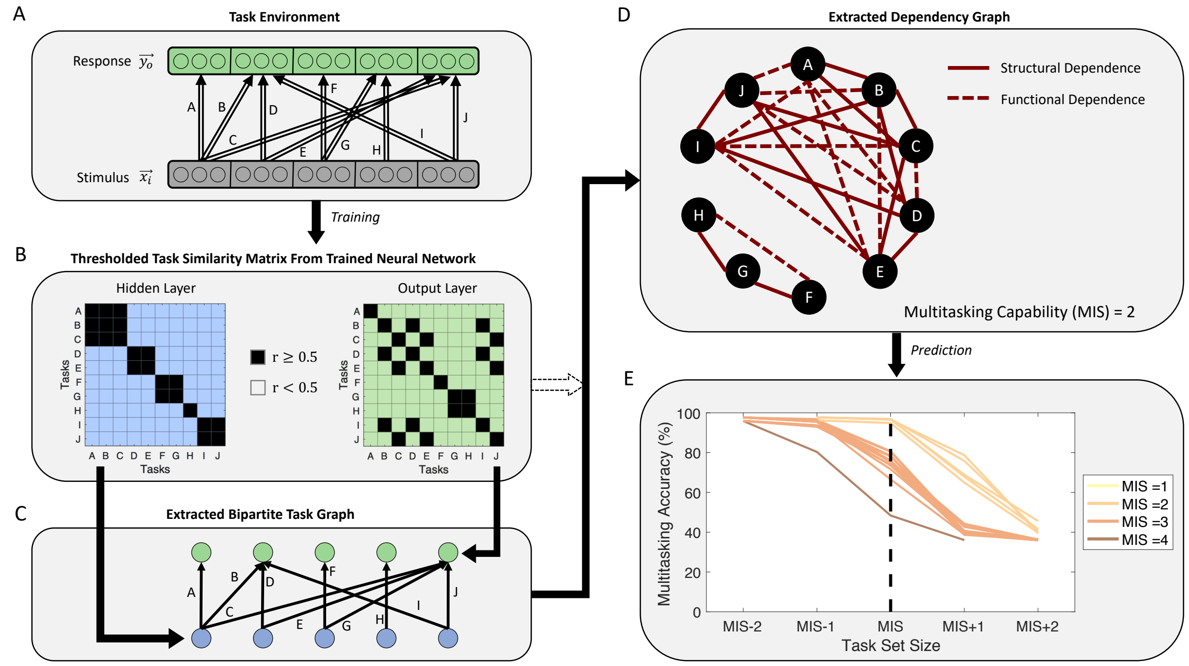

Figure 11. **Prediction of multitasking capability from dependency graph constructed from correlations among single task representations.** (A) A task environment consisting of 10 possible tasks represented as stimulus-response mappings. Each arrow from a stimulus dimension to a response dimension denotes a task. (B) Task similarity matrix computed from correlations among the mean activity patterns learned for each task in the hidden and output layers of a network. Pairs of tasks that exceed a correlation threshold of 0.5 in a given layer are marked in black. The thresholded similarity matrices are used to extract the bipartite (C) and dependency (D) graphs for the tasks (see text). (E) The MIS of the dependency graph is used to predict the multitasking capability of the network. The plot shows the highest multitasking accuracy of a network as a function of the number of tasks it is asked to perform in parallel (multitasking capability curve) and the predicted MIS for that network. Each line corresponds to the multitasking performance of a trained network, whereas the color of each line indicates the predicted MIS for that network. The plot suggests that the multitasking capability curve drops as the set size approaches the predicted MIS.

representations for a pair of tasks within a given layer were considered to be shared if the Pearson correlation coefficient of their mean pattern of activities exceeded 0.5.[19] If a

[19] Thresholding the correlation between task activities was required in order to derive an unweighted dependency graph. However, it is worth noting that some data may be lost when averaging hidden activation patterns across trials; models that operate on unaveraged time series data, by contrast, may offer a more complete measure of sharing and separation. Such models may, for example, attempt to estimate the neural encoding of stimuli while an agent performs each of several tasks, and then

pair of tasks was determined to have a shared representation in the hidden layer, then the two tasks were assigned the same input node in the extracted bipartite task graph. Analogously, if a pair of tasks was determined to have a shared representation in the output layer then both tasks were assigned the same output node. The bipartite graph was then used to generate a dependency graph as described earlier, which was used to examine the multitasking profile of the network.[20] Thus, the dependency graph served as summary of the similarity relationships among tasks, that we used to determine the multitasking capability of the network (i.e., the size of the MIS), as well as the specific combinations of tasks that could and could not be performed concurrently. Fig. 11A-D illustrates this sequence of steps for an example network. It is worth reiterating that the procedure described above requires that the network be examined only on the performance of each task individually, and therefore is substantially more efficient (scaling linearly with number of tasks) than determining the multitasking profile by simulating and examining performance of the network for all combinations of tasks (which scales factorially).

Multitasking capability. To test the extent to which the MIS of the extracted dependency graph for each network predicted its multitasking capability, we compared the analytically-determined MIS with the empirically-observed maximum multitasking performance achievable by each network. We did this by identifying, for each network

compare encoding functions for two different tasks directly, as in Bernardi et al. (2018); U. Cohen, Chung, Lee, and Sompolinsky (2019); Henselman-Petrusek, Segert, Keller, Tepper, and Cohen (2019). It remains a matter for future research to explore how well these measures can be used to predict multitasking capability of network architectures. Nevertheless, all results reported below were qualitatively robust to a wide range of correlation thresholds.

[20] The bipartite graph, and its use in generating the dependency graph, are presented here for clarity and consistency with presentation of the graph-theoretic methods described earlier. However, the dependency graph can also be directly computed from the similarity matrices of the hidden and output layer as follows: An edge is assigned to a pair of tasks in the dependency graph if (1) their correlation exceeds threshold in either of the similarity matrices, or (2) there exists a third task that correlates above threshold with one task in the similarity matrix for the hidden layers and with another in the similarity matrix for the output layers.

and a given number of tasks, the combination of tasks (multitasking set) that yielded the greatest multitasking accuracy. We predicted that the accuracy should remain asymptotically high for multitasking set sizes at or below the analytically-determined MIS, but should decline as a function of set sizes that exceeded it. For example, if the extracted MIS of a trained network was 2, we predicted that the maximum accuracy across multitasking sets would drop for multitasking sets of size of three or more. We refer to the maximum accuracy as a function of multitasking set size as the multitasking capability curve of a network. To statistically evaluate the predictions above, we computed the maximum multitasking capability curve for each network, fit a sigmoid function to each curve[21], and tested the prediction that the inflection point (i.e., offset) of the curve should lie between multitasking set sizes equal to MIS-1 and MIS+1.

Predictions of multitasking accuracy for specific combinations of tasks. We also used the extracted dependency graph to predict how accurately the network could perform particular combinations of tasks, and to characterize the extent to which this was influenced not only by multitasking set size, but also by the number of dependencies between the specific tasks in a given set. For each set size, we computed the multitasking performance for all combinations of that number of tasks. Then, for each set size, we grouped sets based on the number of functional dependencies among the tasks in the set predicted by the dependency graph, and evaluated the effect that this had on multitasking performance across sets. We predicted that multitasking performance for a given set size should drop with the number of dependencies between tasks in the set.

Results. As expected, the dependency graph accurately recovered the task structure imposed during training. That is, it confirmed that the network learned to use similar hidden layer representations for tasks involving the same stimulus dimension (e.g. Tasks A and B in Fig. 11A-B), and that it learned similar output representations

[21] Due to the limited number of data points per curve, we estimated only the slope and offset of the sigmoid function. The maximum and minimum of the sigmoid were fixed to the respective largest and smallest value of the multitasking capability curve.

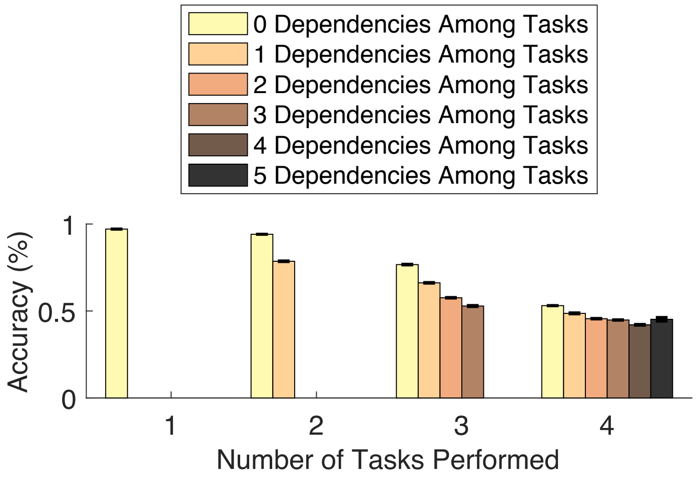

Figure 12. **Network performance for sets of tasks with different numbers of dependencies.** Error bars indicate the standard error of the mean for multitasking conditions of networks trained in different task environments.

for tasks involving the same response dimensions (e.g. Tasks B, D & I in Fig. 11A-B). In Part II of this article (Simulation Study 4), we return to this finding in greater detail and examine the conditions under which the network learns to share representations between tasks. Fig. 11D shows that the predicted multitasking capability (derived from the extracted dependency graph) accurately predicted the maximum number of tasks a network can perform.[22] That is, the inflection point (i.e., offset) of the multitasking capability curve lies significantly above a set size equal to the predicted MIS-1, $t(19) = 3.7810$, $p < 0.001$, and below a set size of MIS+1, $t(19) = -6.6706$, $p < 10^{-5}$. However, as the MIS of a network grows, the analysis begins to overestimate the network's multitasking capability (the drop of the multitasking capability curve occurs before the predicted MIS); that is, the analysis provides a liberal estimate of the constraints imposed by shared representation, which are likely to be even more restrictive in practice (e.g. if only a limited number of tasks are available to perform; see the discussion of *effective multitasking capability* above).

Fig. 12 shows that these analyses also predicted the relative accuracy with which

[22] The prediction is robust to a range of performance metrics, number of hidden units in the network, and choices of correlation threshold (for a robustness analysis, see Petri et al., 2020).

tasks could be performed concurrently that varied by extent of representational sharing. That is, for a given size of a multitasking set, average accuracy decreased reliably as the number of dependencies between tasks predicted by representational sharing increased. Interestingly, in addition to the predicted drop in multitasking performance as a function of dependencies among tasks, we also observed an unpredicted effect: a drop in performance as a function of multitasking set size *irrespective* of how many predicted dependencies there were in the set. This suggests that there were sources of processing interference among tasks other than the dependencies extracted from shared representations at the hidden and output layer, that increased with the number of tasks to be performed. Examination of the networks revealed that a primary source of such cross-task interference is mutual inhibition of output units between tasks. When trained on single tasks, for each task the network learned to suppress irrelevant responses (i.e., associated with the same inputs for other tasks), by developing inhibitory weights for projections from the corresponding task unit in the task input layer to all units in the output layer for task-irrelevant response dimensions. However, this produced cross-task interference when the networks were asked to multitask (something they were not trained to do), an effect that is unrelated to the amount of shared representation between tasks in the hidden and output layer (and thus not captured by the graph theoretic analysis), and scales with the number of tasks to be performed at once, as seen in Fig. 12. This suggests that a similar effect might be observed empirically, for sets of tasks that are predicted to be independent, but for which participants have not be trained to multitask.

2.3.3 Simulation Study 2: Interaction between Representation Sharing and Graded Conflict. The results above offer provisional support for use of graph-theoretic analyses in predicting the effect of shared representations on multitasking performance, subject to the potential for overestimation as the number of tasks grows. However, there is another way in which the analyses presented are limited: they assumed shared processing pathways were of equal strength, and treated the interference associated with the sharing of representations as an all-or-none phenomenon whereas, in actuality, interference can be graded. For example, the relative strength of pathways that share a set of representations can vary by degree of training, that in turn can lead to asymmetric interference effects (e.g., J. D. Cohen et al., 1990; MacLeod & Dunbar, 1988). Thus, graded differences in the relative strength of pathways should be associated with correspondingly graded effects on multitasking performance. Here, we consider the effects of relative differences in connection strengths for pathways that fully share sets of representations. In Part II (Simulation Studies 4-6), we examine the effect of graded degrees of representational sharing.[23]

To illustrate the effects on multitasking of differences in the relative strength of pathways that share representations, consider Tasks A-E shown in Fig. 13. Tasks A, B and C each map a different stimulus dimension to correspondingly distinct response dimension, and thus all are structurally independent of one another. However, if Tasks A and B share representations with Tasks D and E, respectively, then they are functionally dependent. Previously, we considered the connections implementing such tasks to all be of equal strength, and thus functional dependence to be all-or-nothing. However, previous work (J. D. Cohen et al., 1990; Gilbert & Shallice, 2002; MacLeod & Dunbar, 1988) suggests that conflict introduced by Tasks D and E on tasks B and A, respectively, should increase as the strength of pathways for the former increases relative to the latter. That is, progressive training on Tasks D and E should have a graded effect on the ability to multitask A and B (see Fig. 13A), while it should have no

[23] For clarity of exposition, we treat strength of processing (here) and representational sharing (in Part II) as separate factors. However, it should be noted that in network architectures with distributed representations, pathway strength and representational sharing, though potentially dissociable, may also be closely related to one another. For example, in the case of two processing pathways that vary in the strength of their connections to a shared set of processing units, the degree of overlap could be-expressed as the strength of the connections in each pathway to the processing units that are shared. However, at the other extreme, if they are both connected to an equal number of units with equal strengths, then the degree of sharing (number of shared units) can be dissociated from their relative strengths. These are factors that can be determined by learning, and that we consider in greater detail in Part II of this article. Here, we focus on conditions in which varying learning impacts strength but not extent of sharing.

impact on the ability to multitask either of the latter with Task C (see Fig. 13B). Here, we describe simulations of such effects, and confirm expected dependencies between tasks using the graph theoretic methods presented above. We also apply quantitative methods for estimating parallel versus serial processing from reaction time data (Townsend & Wenger, 2004) to determine the amount of interference in the network. To do so, we trained networks on all tasks shown in Fig. 13, varying the amount of training that the network received for Tasks D and E relative to Tasks A, B and C, and evaluating multitasking performance of Task A with Tasks B and C.

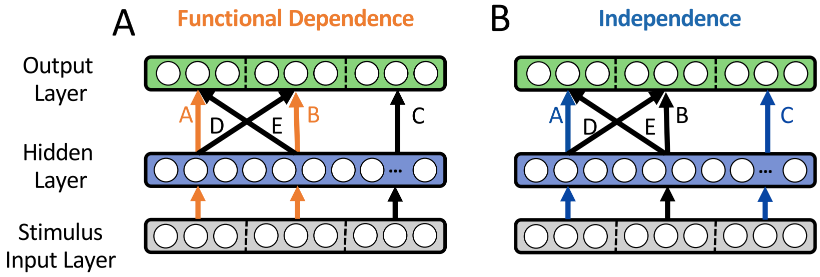

Figure 13. **Task dependencies used in Simulation Studies 2 and 3.** (A) Tasks A and B are assumed to be functionally dependent due to shared representations with Tasks D and E; thus, the ability to multitask A and B should be impacted by the strength of D and E. (B) Tasks A and C are assumed to be independent, and thus multitasking should *not* be affected by the strength of D and E (see text for discussion)

Network architecture and processing. These simulations used a variant of the network architecture described for Simulation Study 1, in this case with just three stimulus dimensions (containing three features per dimension) and three response dimensions (also with three features per dimension). The network was trained on the subset of tasks described below.

Task environment. For each simulation we implemented tasks corresponding to A-E in Fig. 13, such that Tasks A, B and C each mapped different stimulus dimensions to distinct response dimensions; Task D shared a stimulus dimension with Task A and a response dimension with Task B; and, conversely, Task E shared a stimulus dimension with Task B and a response dimension with Task A.

Training. We initialized 20 networks for each training condition with small random weights. For each training condition, we sampled 100 patterns for each of the three Tasks A, B and C per training epoch. For Tasks D and E, however, we varied the number of patterns sample across conditions from none (0% task strength) to 150 (150% task strength relative to Tasks A, B and C). Every network was trained until it reached the same performance criterion for Tasks A, B and C.

Functional dependencies between tasks. To confirm assumptions about functional dependencies between tasks, we assessed the similarity of the learned representations for each pair of tasks after training, and then applied the graph-theoretic methods described above to determine dependencies between tasks. In Simulation Study 1, we focused on analyzing average patterns of activity for each task, to demonstrate how graph-theoretic methods can be applied to neuroimaging data (e.g. fMRI). Here, we quantify representation sharing by calculating the Pearson correlation of their weight vectors to the hidden layer, as these provide a more direct measure of representational overlap, i.e. the degree to which two task units project to the same hidden units.[24] We then applied the same graph-theoretic analysis to extract a functional dependencies between tasks from the correlations.

Intermediate Results: functional dependencies and multitasking accuracy. Analyses of the learned representational similarity between tasks confirms that Tasks A and B are functionally dependent whereas Tasks A and C are independent. Fig. 14 shows the correlation between learned task representations in the hidden layer of the network, averaged across all networks and training conditions. As expected, Tasks A and D learned shared representations in the hidden layer, as did Tasks B and E, since each pair relied on the same set of stimulus features. As a result, Tasks A and B were found to functionally dependent on one another, whereas they were independent of Task C. We assessed the multitasking accuracy for performing Tasks A and B, and similarly for Tasks A and C, as well as the single task accuracy for Tasks D and E as a function of training on Tasks D and E(Fig. 14C). Multitasking performance for Tasks A and B

[24] Prior simulations (not reported) suggest that weight vectors yield more accurate predictions of multitasking performance than averaged patterns of activity.

decreased with the amount of training on Tasks D and E, while performance for Tasks A and C was virtually unaffected by the training condition. Even small amounts (30%) of training on Tasks D and E, sufficient to improve their performance, came at the expense of impaired multitasking performance of Tasks A & B. This suggests that detriments in multitasking performance scale with the degree of interference induced by shared representations. In other words, shared representations alone may not be sufficient to impair multitasking performance, but they do so if the processing strength of these other tasks induces a sufficient amount of interference.

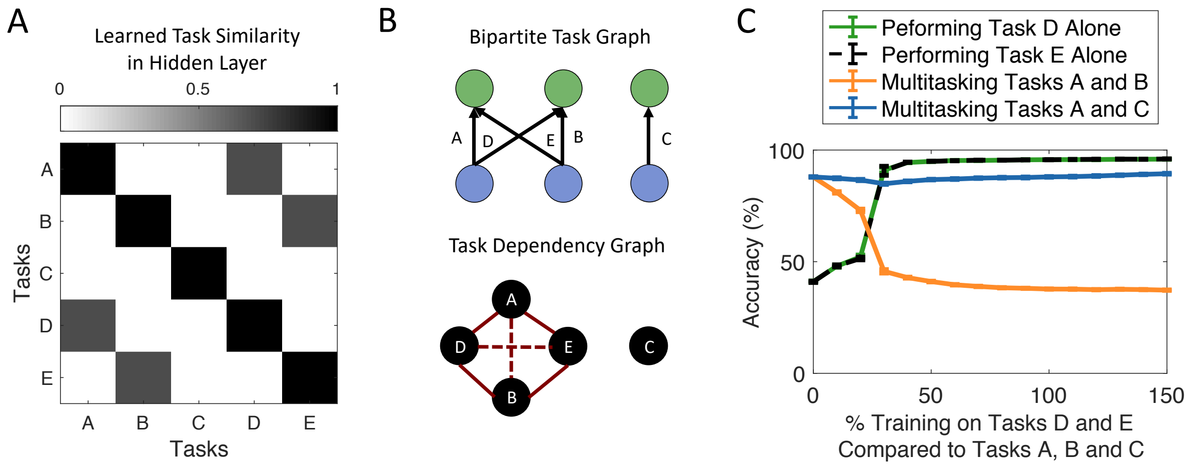

Figure 14. **Effects of shared representation and graded interference on multitasking accuracy.** (A) Average correlations between learned task representations in the hidden layer. (B) Bipartite Task Graph and Task Dependency Graph extracted from the similarity between task representations at the hidden and output layers. Solid lines in the dependency graph indicate structural dependence whereas dashed lines indicate functional dependence. (C) Single tasking performance of Tasks E and D, as well as multitasking performance for Tasks A & B and Tasks A & C as a function of training on Tasks D and E (cf. Fig. 13). Error bars indicate the standard error of the mean across 20 simulated networks.

Response time series after single task training. The results above focused on the effects of shared representation in networks with non-linear processing units, and evaluated in terms of multitasking accuracy. This complements a separate, but closely related line of work pursued by Townsend and colleagues (Townsend, Ashby, et al., 1983; Townsend, Ashby, Castellan, & Restle, 1978; Townsend & Wenger, 2004), developing mathematical methods for inferring the extent of parallel processing involved in task performance from measures of cumulative reaction time (RT) distributions. These methods assume that task performance relies on linear integration processes. Here, we examine the extent to which these methods can be used to infer parallel processing (and hence multitasking capability) in networks composed of nonlinear processing mechanisms. In particular, we evaluate their sensitivity to shared representations, and whether this aligns with the results described above using measures that infer multitasking capability from network representations rather than performance.

Specifically, Townsend and Wenger (2004) showed that the cumulative RT distribution for two non-interacting (i.e., parallelizable) linear integration processes T_A and T_B both reaching a fixed threshold before time step t lies within the bounds formulated by Colonius and Vorberg (1994):

$$\begin{aligned} &P_A(T_A \leq t) + P_B(T_B \leq t) - 1 \\ &\qquad \leq P_{AB}(T_A \leq t \text{ AND } T_B \leq t) \leq \\ &\qquad\qquad min[P_A(T_A \leq t), P_B(T_B \leq t)] \end{aligned} \tag{5}$$

where $P_A(T_A \leq t)$ and $P_B(T_B \leq t)$ are the probabilities of each task reaching its threshold, respectively, conditioned on having a feature present in the stimulus dimension relevant to each task and the responses being the correct ones for those stimuli. Conversely, interactions between two processes (i.e., cross-talk) should lead to violations of these bounds (Townsend & Wenger, 2004). Here, we tested whether similar properties are observed for performance in networks with non-linear processing units and distributed representations; that is, whether tasks implemented in such networks that are functionally independent obey the inequalities above, while ones that are functionally dependent violate it, and the extent to which this is sensitive to the relative strength of the pathways involved. To do so, we assessed $P_{AB}(T_A \leq t \text{ AND } T_B \leq t)$ for Tasks A and B, as well as $P_{AC}(T_A \leq t \text{ AND } T_C \leq t)$ for Tasks A and C in the networks described above, as a function of the strength of Tasks D and E, where t corresponded to the time taken by the LCA to reach threshold.[25]

The results indicate that, while multitasking both pairs of tasks (A & B, and A & C) strictly violated the inequality, this effect was distinctively greater for Tasks A and B when the tasks that mediated the functional interference between them – Tasks D and E – were strong (i.e., fully trained) compared to the other conditions (see upper right panel of Fig. 15). In that case, Tasks A and B crossed the lower bound of the cumulative RT distributions for independent processing channels at a much later point than in the other conditions, indicating that it took more time for both tasks to reach a response presumably due to functional interference. Thus, the degree of inequality violation appears to clearly reflect the degree of functional dependence. The observation that the inequality was also violated in the other conditions (though to a much less degree) is consistent with an effect discussed earlier: Training on single tasks can lead the network to learn to directly inhibit output representations that are not relevant to the current task, causing multitasking interference at the output layer (see Simulation Study 1).

Response time series after multitasking training. While the discrepancy between the analysis of RT distributions and the graph-theoretic analysis across conditions may, as just noted, reflect the effects of learning, it is possible that this could also be due to the non-linearity of processing and/or presence of distributed representations in the network, both of which deviate from assumptions made by the RT analysis of Townsend and Wenger (2004). To evaluate this, we sought to eliminate any effects of cross-task interference by training the network explicitly on multitasking for Tasks A & B as well as for Tasks A & C, and then evaluating its performance using the analysis of the RT distributions. If this eliminated the violations of the inequalities, it would suggest that those were due to the effects of cross-task interference that arose from single task training, whereas if the violations persisted it would suggest that they were due to deviations of the network architecture from assumptions made by the analysis.

In this simulation, 20 networks were trained until criterion on all five tasks as

[25] We conformed to the same assumption used by Townsend and Wenger (2004), restricting our analysis to only correct responses.

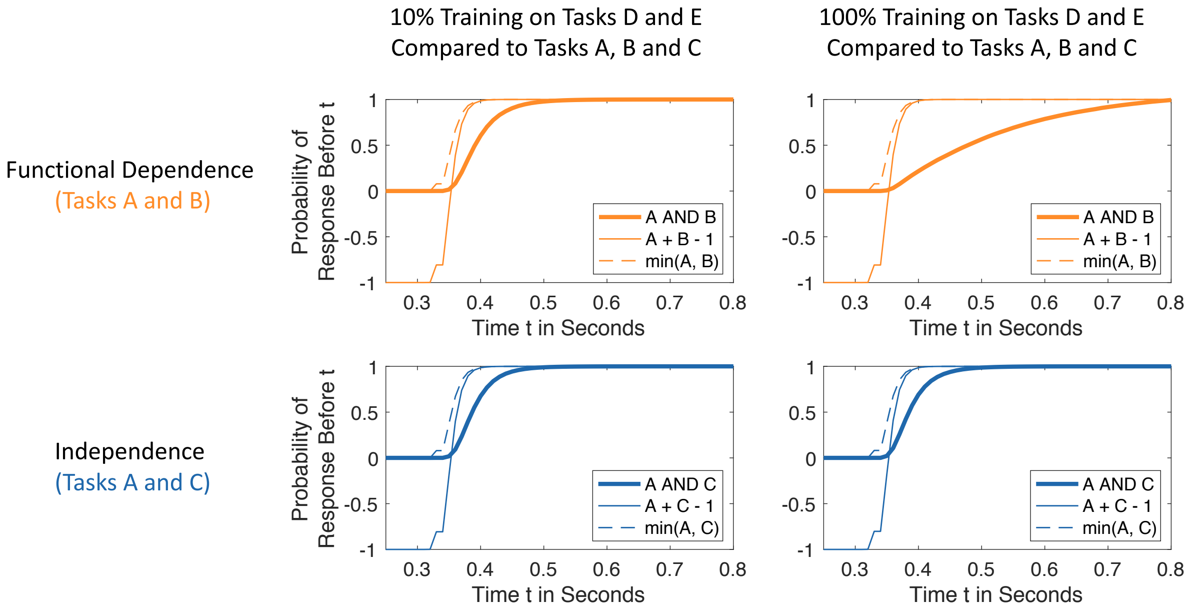

Figure 15. **Cumulative RT distributions as a function of dependence (shared representation) and training on interfering tasks (graded interference).** Each plot shows the lower $(A + X - 1)$ and upper $(min(A, X))$ bounds (thin solid and dashed lines, respectively) for the cumulative RT distribution of multitasking Tasks A and X (thick solid lines), where X is either Task B (upper panels) or Task C (lower panels); see Fig. 13 for task configurations. Cumulative RT distributions are shown for either 10% (left panels) or 100% (right panels) of training on Tasks D and E, relative to the other tasks (as a manipulation of the strength of those pathways). Note that, whereas the cumulative RT distribution evolves to fall below the lower bounds in all conditions, it does so to a considerably greater degree for Tasks A and B when Tasks D and E strong (100% training condition; upper right panel) compared to the other conditions; see text for discussion.

described above (with 100% training on Tasks E and D). In addition, each training epoch included 100 patterns of multitasking Tasks A & B, as well as 100 patterns for multitasking Tasks A & C. After training, the representational similarity between all tasks, as well as the cumulative RT distribution for both multitasking conditions was assessed as described above.

Multitasking training virtually eliminated representational sharing between tasks that relied on a common stimulus dimension (Tasks A and D, as well as Tasks B and E; see Fig. 16A), and thus eliminated the functional interference between Tasks A and B, which was required to achieve criterion in training on multitasking performance. We will consider these effects of multitask training on shared representations in greater detail in Part II (Simulation Study 5), Here, we note that the analysis of RT distributions accurately reflects this effect, now showing strict adherence to the inequalities indicative of full parallel processing (Fig. 16B). These results suggest that the methods described by Townsend and Wenger (2004) can be extended to the analysis of non-linear systems (at least those implemented in the networks described above), and that measurements using these methods align with an assessment of parallelism in such networks based on the graph theoretic analysis as well as the accuracy of multitasking performance of such network evaluated directly in simulations. These results also suggest that for the simulations involving single task training reported above, the analysis of RT distributions was able to detect interactions between tasks that arose during learning, but were not predicted by graph theoretical analysis of representations at the hidden and output layers (see Simulation 1 for a discussion).

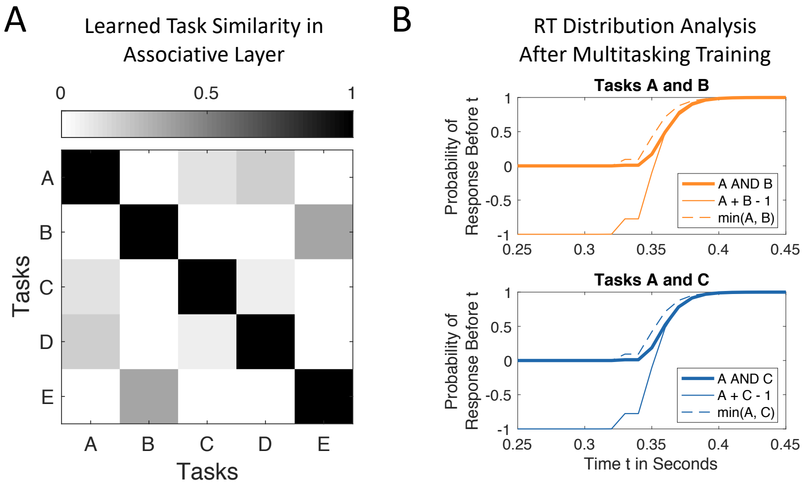

Figure 16. **Representational task similarity and cumulative RT distributions after multitasking training.** (A) Average correlations between learned task representations in the hidden layer (cf. Fig. 14). (B) Each plot shows the lower ($A + X - 1$) and upper ($min(A, X)$) bound for the cumulative RT distribution of multitasking Tasks A and X, where X corresponds to either Task B (upper panel) or Task C (lower panel); see text and Fig. 15 for explanation of bounds; and Fig. 13 for task configurations.

2.3.4 Simulation Study 3: Interaction Between Shared Representation and Persistence in Multitasking.

While training can be used to overcome multitasking interference due to functional dependence – a topic to which we will return at length in Part II – it is of course also possible to overcome such interference by executing the individual tasks in series. However, a large body of evidence suggests that, for humans, serial execution of tasks is also associated with costs. Serial task execution has been studied in a number of experimental paradigms, the two most prominent of which are the PRP procedure (Telford, 1931) and the task switching paradigm (Allport et al., 1994; R. D. Rogers & Monsell, 1995). Interestingly, however, little work has addressed the relationship of effects between these; that is, between dual-task interference in the PRP paradigm and switch costs associated with task-switching (Koch et al., 2018). Furthermore, a neural mechanism underlying both effects remains elusive. Here, we suggest that both reflect interference arising from the same underlying mechanism: an interaction between shared representations and the persistence characteristics of representations in neural architectures.

In the PRP procedure, participants are asked to respond as quickly as possible to two tasks within the same trial. Each trial begins with the presentation of a stimulus relevant to the first task (Task 1), followed by an experimentally manipulated delay (the stimulus onset asynchrony; SOA) and then the stimulus for the second task (Task 2; Fig. 17). Participants tend to respond more slowly to the second stimulus as the SOA is reduced (Telford, 1931). The additional amount of time that it takes to respond to the second task in the presence of a short SOA is referred to as the PRP. If the two tasks could be performed fully in parallel, then participants should execute Task 2 as soon as the relevant stimulus is available, and there should be no PRP. Therefore, observation of a PRP has been interpreted as evidence for a serial processing architecture, in which both tasks rely on a central, limited capacity control mechanism. The latter is assumed to impose a bottleneck on processing, which delays the execution of Task 2 while Task 1 is still being executed (e.g., Welford, 1952; Broadbent, 1957, 1958; Pashler, 1984, 1994). Alternatively, production system models closer in spirit to the multiple resource theory have suggested the PRP effect can be explained by bottlenecks that arise within more local resources (e.g., perceptual or motor processes) shared by the particular tasks that are competing for execution, rather than a “central executive” (Byrne & Anderson, 2001; Kieras & Meyer, 1997; Meyer & Kieras, 1997b; Salvucci & Taatgen, 2008). However, those models do not explain *why* such bottlenecks exist; nor, to our knowledge, have they used the same mechanisms to explain effects in task switching paradigms.[26]

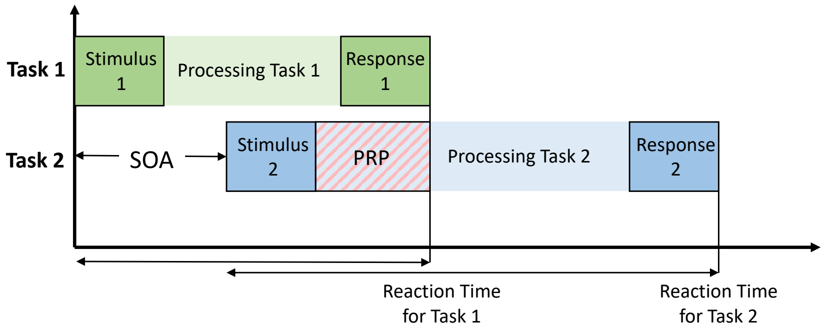

Figure 17. **Psychological refractory period (PRP) procedure (Telford, 1931).** See text for description.

In task switching experiments participants are required to respond to only one task per trial, but must switch periodically between tasks across trials. A large literature reports a number of effects consistently observed in such experiments (for a review, see Kiesel et al., 2010). Here we focus on the explicit task-cuing procedure, in which each trial is preceded by a task cue indicating the next task to be performed (Meiran, 1996; Sudevan & Taylor, 1987). Task switch trials require the participant to perform a different task relative to the previous trial, whereas task repetition trials require the participant to perform the same task again. Participants reliably exhibit a *switch cost* on task switch trials; that is, they respond more slowly and/or less accurately on task switch relative to task repetition trials. Switch costs have been attributed to various cognitive processes. Some have suggested that switch costs reflect an active process of task-set reconfiguration (Mayr & Kliegl, 2000; Meiran, 1996; R. D. Rogers & Monsell, 1995; Rubinstein, Meyer, & Evans, 2001) that relies on a control mechanism. Others have suggested that switch costs arise from passive processes, such as: proactive interference (sometimes referred to as “task-set inertia”)

[26] In Part II we return to the question of why such bottlenecks might arise, providing an account in terms of the value of shared representations during learning. Here we focus on an explanation of PRP and task switching effects in terms of common underling mechanisms.

from the previous task-set (Allport et al., 1994); inhibition of the previously executed task-set (Altmann, 2007; Mayr & Keele, 2000); repetition priming of the task cue (Logan & Bundesen, 2003; Anderson & Lebiere, 2014); or repetition priming of stimulus features (Waszak, Hommel, & Allport, 2004; Wylie & Allport, 2000). Note that *all* of these accounts assume some form of persistence of information encoded about the previous task. In a neural network architecture, this is naturally interpreted as the persistence of the activity patterns used to represent such information.

The persistence of activity is a common computational feature of neural network architectures, that enables integration of information over time. Persistence characteristics have been used to account for a variety of cognitive phenomena, including sequential processing of stimuli (Braver, Barch, & Cohen, 1999; Elman, 1990), working memory (Engle, Kane, & Tuholski, 1999), integration of sensory input in perceptual decision-making (Curtis & Lee, 2010; Major & Tank, 2004; Mazurek, Roitman, Ditterich, & Shadlen, 2003; Shadlen & Newsome, 2001; Usher & McClelland, 2001), temporal credit assignment in reinforcement learning (O'Reilly & Frank, 2006), and the evolution of context representations proposed to underlie event segmentation and temporal encoding in episodic memory (Hasson, Chen, & Honey, 2015; Lerner, Honey, Silbert, & Hasson, 2011). Persistence of activity also suggests that the effects of shared representation on multitasking performance may extend to the sequential execution of two tasks: the more that a representation of a previously executed task persists in time, the more it can interfere with a subsequent task that shares the same set of representations. Here, we show that such an interaction between persistence of activity and shared representations can explain interference effects associated with the sequential execution of tasks, both in the context of PRP experiments as a function of SOA, and task switching experiments as a function of response set overlap and stimulus congruency.

Network architecture, processing and task environment. Using the same neural network model and task environment as described in the previous section, we trained 20 networks on Tasks A-E (see Fig. 13) until each network reached the performance criterion across all tasks. After training, we introduced persistence in the computation of the net input of a unit i in the hidden and output layers,

$$\overline{net_i}^T = (1-p) \cdot net_i^T + p \cdot \overline{net_i}^{T-1}, \tag{6}$$

where $\overline{net_i}^{T-1}$ corresponds to the time averaged net input from the previous time step, net_i^T corresponds to the instantaneous net input, and p determines how much the time averaged net input of the current time step $\overline{net_i}^T$ depends on the time averaged net input from the previous time step.[27] Thus, the higher the value of p, the longer activity persists in a given state over time. For each network, we considered different values of $p \in \{0, 0.5, 0.8, 0.9\}$.

PRP after single task training. We simulated the PRP paradigm for Tasks A and B, as well as Tasks A and C. As demonstrated in the previous section, after single task training, Tasks A and B were functionally dependent and interfered with each other when executed simultaneously, whereas Tasks A & C were independent and interfered less (cf. Fig. 14). Here, we examined the effects of sequentially executing each pair of tasks, with Task A always executed second. Thus, we first presented the network with a feature from the stimulus dimension relevant to Task 1 (Task B or Task C), by activating the corresponding unit in the stimulus input layer and by keeping all other stimulus input units inactivated. After a number of time steps (determined by the SOA), we presented the network with a feature from the stimulus dimension relevant to Task 2 (Task A) by activating a unit in the stimulus dimension relevant to that task while the stimulus feature for Task 1 (Task B or Task C) was still present. PRP studies

[27] This implementation of persistence by integrating (“time-averaging”) the net input to each unit follows similar implementations (e.g., Cohen et al., 1990), though it can also be achieved through recurrent excitatory connections (e.g., Usher & McClelland, 2001). For efficiency of simulation, training occurred without integration so that, after training, integration during processing causes activity patterns to asymptote on the learned patterns. Similar results were shown to apply when integration is applied throughout training (Herd et al., 2014), so long as sufficient time is afforded during each training trial for the activity of the network to approach asymptote.

commonly instruct participants to prioritize Task 1 (Meyer & Kieras, 1997b). We therefore activated the task input layer unit for Task 1 at the beginning of each trial, and deactivated it as soon the network had responded to that task. For Task 2 we assumed that participants sought to optimize the outcome of performance by choosing to initiate execution at a time that maximized reward rate. Accordingly, we determined the optimal onset of the task unit for Task 2 such that the joint reward rate for both tasks was maximized, with

$$\text{Reward Rate} = \frac{P(\text{correct})_{\text{Task 1}} P(\text{correct})_{\text{Task 2}}}{(\text{ITI} + \text{RT}_{total})} \tag{7}$$

where $P(\text{correct})_{\text{Task 1}} and P(\text{correct})_{\text{Task 2}}$ correspond to the accuracies of Task 1 and Task 2, respectively; ITI corresponds to an inter-trial interval of 0.5s;[28] and RT_{total} is the RT that was determined by the time of the response to the last task to be executed, measured from the onset of the trial. We then assessed RTs for Task 1 (Task B or Task C) and Task 2 (Task A) as a function of SOA, by varying the SOA from 1s to 8s in steps of 1s (with each simulation step amounting to 0.1s).

PRP after dual-task training. A number of studies have demonstrated that the PRP can be eliminated after a sufficient amount of dual-task training (Allport et al., 1972; Hazeltine, Teague, & Ivry, 2002; Liepelt et al., 2011; Schumacher et al., 2001; Wickens, 1976), yielding “virtually perfect time sharing.” Accordingly, we tested whether the PRP remained if the network was trained on dual-tasking Tasks A and B, as well as on Tasks A and C. To do so, we trained 20 networks to criterion on all five tasks as described above (with 100% training on Tasks E and D, to allow for the possibility that shared representations and functional interference would develop between Tasks A and B). In addition, each training epoch included 100 patterns of dual-tasking for Tasks A and B (to determine whether any PRP effects that occurred following single task training were eliminated by dual-task training), as well as 100 patterns for dual-tasking Tasks A and C. After training, we measured the PRP as a

[28] The duration of the ITI varies across PRP studies. Here, we choose an ITI of 0.5, similar to Halvorson et al. (2013).

function of SOA, as well as the the amount of representation sharing that developed between tasks (see Simulation Study 2).

Results: PRP after single- and dual-task training. Simulation results validated the expected effect that higher persistence prolonged RT for both Task 1 and Task 2, due to slower rates of integration (Fig. 18). Critically, following single task training, the model exhibited a PRP effect for all non-zero values of persistence, showing a delay of Task 2 as a function of SOA (Fig. 18B). This effect was greater when Task 2 (always Task A) followed Task B versus C, indicating that Task B interfered more with the subsequently executed Task A. This is consistent with persistence of shared representations between Tasks A & D, as well as Tasks B & E, that produced functional interference between Tasks A and B but not A and C, and therefore that the effects of functional interference can be mediated by persistence in shared representations, even when tasks are executed serially. Interestingly, a PRP effect, albeit smaller, was also observed when Task A followed Task C. This is consistent with the results of Simulation Studies 1 and 2 suggesting, once again, that interference between tasks can arise through suppression of responses at the output layer acquired during single task training (see Simulation 1 for a discussion). The results here suggest that persistence can amplify this effect, and produce a PRP even for tasks that are functionally independent according to the graphic theoretic analysis. It is also worth noting that, in line with prior observations (Marill, 1957; Pashler, 1994), the RT of Task 1 remained unaffected by the SOA, irrespective of whether Task 1 was functionally dependent or independent of Task 2 (Fig. 18A). That is, a potentially early execution of Task 2 did not interfere with an ongoing execution of Task 1. This reflects the instructed strategy of the model to prioritize Task 1, by activating the task unit for Task 1 before the task unit for Task 2. This strategy allowed the model to elicit a response for Task 1 before the activity of the task unit for Task 2 became high enough to cause interference.

Finally, Fig. 19 shows that dual-task training, which greatly diminished representational sharing (Fig. 19A), all but eliminated the PRP effect; it is now observed only at the highest levels of persistence ($p \geq 0.8$, Fig. 19B).

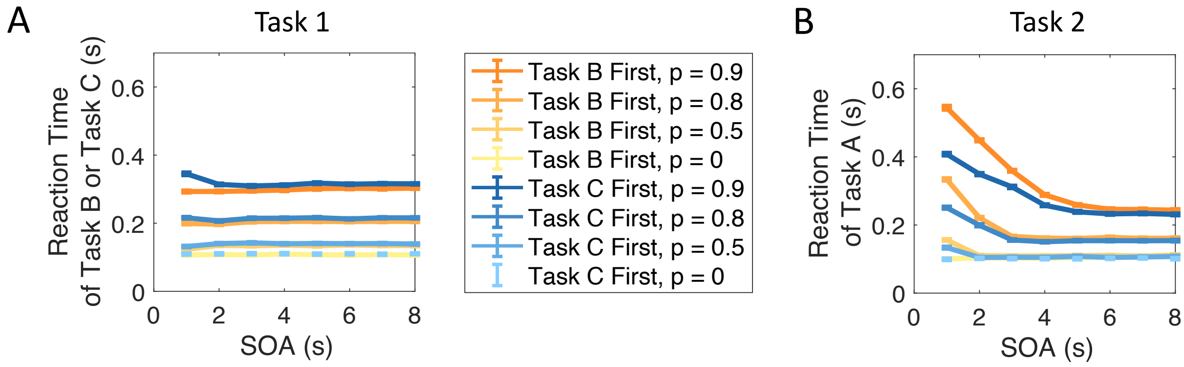

Figure 18. **Simulated PRP after single task training.** RTs for (A) Task 1 and (B) Task 2 in the PRP procedure as a function of persistence p, as well as Task 1 (B or C). Error bars show the standard error of the mean across 20 simulated networks trained only on single tasks.

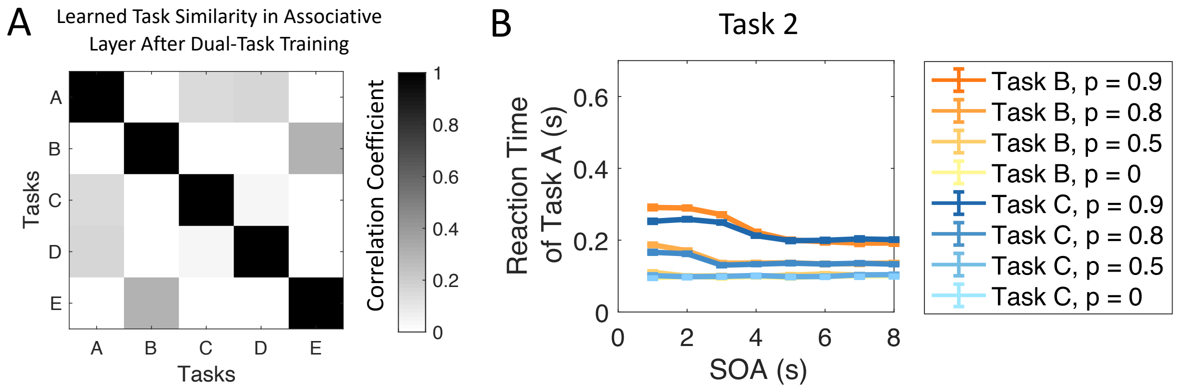

Figure 19. **Simulated PRP after multitasking training.** (A) Average correlations between learned task representations in the hidden layer. (B) RT of Task 2 in the PRP paradigm as a function of persistence p and task. Error bars show the standard error of the mean across 20 simulated networks.

Task switching. A large number of empirical studies have shown that switch costs can vary, depending on whether the pairs of tasks involved share the same set of (bivalent) responses or whether they use different (univalent) sets of responses. Our analysis of task dependence suggests a refinement of this distinction, such that task pairs with with bivalent responses are structurally dependent (e.g. Task A and Task E), whereas task pairs with univalent responses may be either functionally dependent (e.g. Task A and Task B) or independent (e.g. Task A and Task C). This, in turn, suggests more refined predictions concerning switches between tasks that have univalent responses: ones that are functionally dependent should exhibit switch costs, whereas ones that are independent should not. We tested these predictions in the same networks trained for the PRP simulations, by comparing performance in three task-switch sequences (see upper panels of Fig. 20): Task E to Task A (structural dependence), Task B to Task A (functional dependence), and Task C to Task A (independence), and computing the switch cost of each relative to a repetition sequence (Task A twice in a row).

Each task in each sequence was simulated by setting its unit in the task input layer to 1 and all others to 0; randomly selecting a stimulus pattern (either congruent or incongruent, cf. Fig. 10) for the stimulus input layer (with one unit active in each stimulus dimension)[29]; and allowing the network to process the input until it reached a response. Task 1 was either Task E, Task B or Task C in task switch sequences, and Task A in task repetition sequences. It was followed by the presentation of Task 2 (always Task A). We measured switch costs as the difference in RT between the switch and repeat conditions, averaged over 100 randomly sampled congruent stimuli and, separately, averaged over 100 randomly sampled incongruent stimuli, calculated separately for the three switch scenarios. As in Simulation Studies 1 and 2, the RT of the network was determined by the response threshold that maximized reward rate for a given combination of task and stimulus inputs. Note that the model did not implement any mechanism by which the RT was explicitly delayed on task switches as opposed to task repetitions. Thus, a slower RT on task switch trials relative to task repetition trials would reflect a normative strategy of raising the response threshold to maximize reward rate.

Results: task switching. Fig. 20 shows the RTs for Task A in all three switch sequences and congruency conditions, compared to those for the repeat condition. The network exhibited switch costs (i.e., a higher RT for task switches, Allport et al., 1994; R. D. Rogers & Monsell, 1995) compared to task repetitions for all three sequence types. The results also indicate that switch costs for structurally dependent tasks (Task

[29] Stimuli for which the features of the stimulus dimensions for both tasks are present are commonly referred to as “bivalent” stimuli as they afford the application of a competing task.

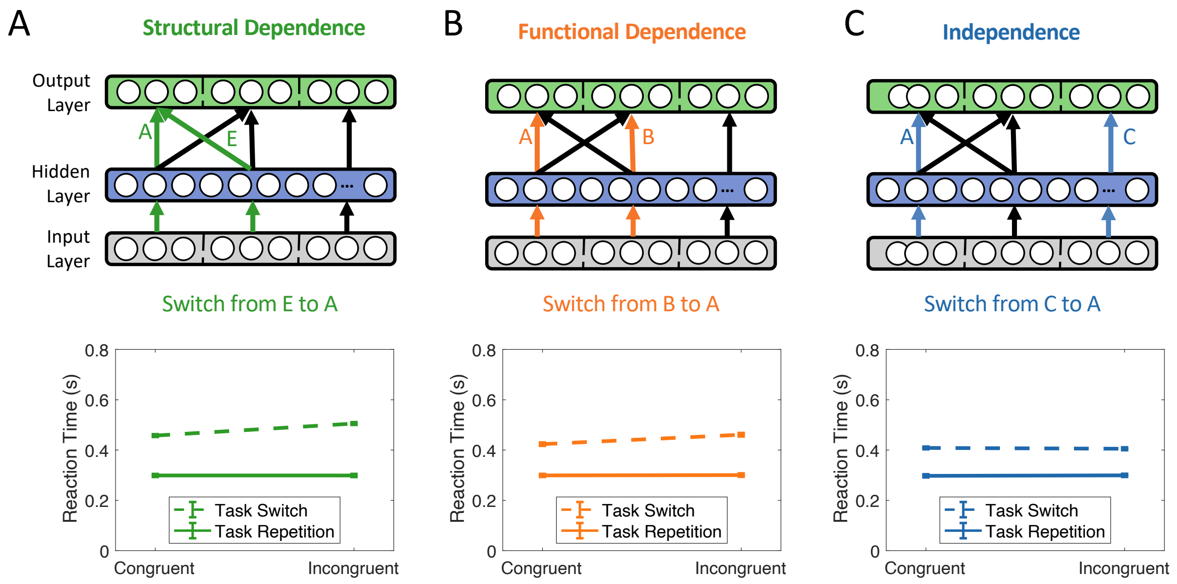

Figure 20. **Effects of shared representations on task switching.** The three upper panels show task pairs used in simulations of each of the three switch sequences. Lower panels show corresponding RTs for Task 2 (always Task A) in each of the three switch sequences (dashed lines) compared to the repetition sequence (solid lines), for congruent and incongruent stimuli, and a persistence of $p = 0.9$. Error bars show the standard error of the mean across 10 simulated networks.

A and Task E) and functionally dependent tasks (Task A and Task B) were higher for incongruent stimuli compared to congruent stimuli. Such an interaction between task transition and stimulus congruency has frequently been reported for structurally dependent tasks (using "bivalent" responses; e.g. Fagot, 1995; Goschke, 2000; Meiran, Chorev, & Sapir, 2000; R. D. Rogers & Monsell, 1995; Wendt & Kiesel, 2008). Previous accounts have suggested that higher switch costs for incongruent stimuli reflect an increase in "proactive interference" (Kiesel et al., 2010). In our simulations, persistence of shared representations from the previously executed task mediated this effect, and the longer RTs observed for incongruent trials reflect the effects of such interference. As expected, we did not observe this effect for independent tasks (Task A and Task C) although persistence of activity from the to be repeated task facilitated task repetitions relative to task switches. Note, however, that this makes the novel prediction that switch costs for pairs of tasks with univalent responses (i.e., that involve different response dimensions) should nevertheless differ, based on whether they are functionally dependent tasks (such as Tasks A and B) or independent (such as Tasks A and E). To our knowledge, this is an effect that was not yet examined in the literature.

Finally, Fig. 21 illustrates the effect of persistence on the switch costs, averaged across all stimuli, for each of the three sequence types. Switch costs increase with persistence in all three though, over most of the range, switch costs are greater for structurally dependent tasks than functionally dependent and independent tasks, mirroring the empirical observation that switch costs for tasks with bivalent responses are higher compared to tasks with univalent responses (Brass et al., 2003; Meiran et al., 2000). Again, however, the model makes the novel prediction of a distinction among univalent tasks, that can be empirically tested.

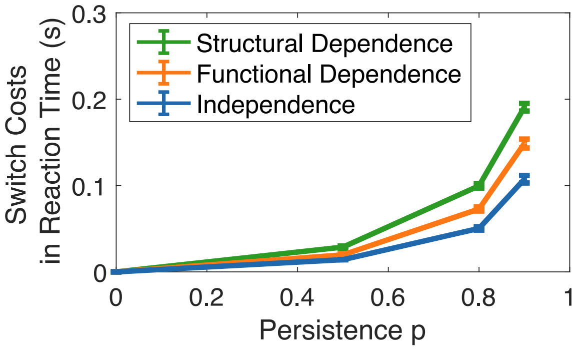

Figure 21. **RT switch costs for incongruent stimuli in all task switching scenarios (see text) as a function of persistence.**

2.4 Summary, Discussion and Conclusions for Part I

2.4.1 Summary. We introduced a graph-theoretic approach to compute the multitasking capability of feed-forward, single-layer, non-linear networks from task-related patterns of activity over their hidden and output layers, and used this to predict network performance for different multitasking conditions. This involved representing the network as a bipartite graph, and using that to generate a task dependency graph that provides a compact representation of its multitasking capability. Determining the MIS in the dependency graph identifies the maximum number of concurrent tasks that can be executed without performance loss. The dependency graph can also be used to identify all combinations of tasks that can be performed in parallel.

Building on this formalism, we conducted a quantitative analysis of the multiple resource theory, demonstrating that the multitasking capability of the network drops drastically with the sharing of representations in the network. Furthermore, we showed that the sharing of representations interacts with the strength of processing pathways and the persistence characteristics of network representations, to define a continuum in the dependence on control, and a commensurate one between parallel and serial processing for given combinations of tasks. Finally, we showed how these factors can provide a mechanistic account of widely observed interference effects between tasks, including the PRP and task switch costs, and generate new predictions concerning these phenomena as a function of the persistence characteristics and sharing of representations between tasks.

Below, we review implications of the analytical results for the multiple resource theory, and discuss how the underlying graph-theoretic framework can be applied to predict multitasking performance from neural correlates. We then describe the relationship between estimations of multitasking capability based on neural measures, on the one hand, and behavioral measures on the other (Townsend & Altieri, 2012; Townsend & Wenger, 2004). Finally, we consider some broader implications that viewing task performance and control dependence through the lens of shared representations has for the interpretation of classic phenomena, such as the PRP, task switching, and cognitive control more broadly.

2.4.2 A Quantitative Approach to the Multiple Resource Theory. As noted above, the graph-theoretic framework permits a quantitative analysis of the multiple resource theory, according to which parallel processing limitations can arise due to local processing bottlenecks of shared task representations rather than a central capacity limitation of the control system itself (Allport et al., 1972; Allport, 1980; Navon & Gopher, 1979; Wickens, 1991). Analytical investigations of the multitasking capability of two-layer networks confirmed previous numerical results (Feng et al., 2014), showing that small increases in the average number of tasks that share a representation lead to dramatic constraints on the number of tasks that can be executed simultaneously without cross-talk.

One may argue that the constraints that shared representations impose on multitasking are negligibly small in a processing system as large as the human brain. The structural capacity of a network may grow both with the number of nodes per processing layer and the number of processing layers. Our analytical results suggest that the multitasking capability of a two-layer network increases in a dramatically sub-linear way with the number of nodes in a processing layer, yielding diminishing returns. That is, the limitations imposed by shared representations may not be easily circumvented by increasing the number of nodes per processing layer in a network. Furthermore, although exact analysis of networks quickly becomes intractable as the size of the network grows, a probabilistic approach to the analysis of deep networks reveals that multitasking capability decreases even further as the number of processing layers in a network increases, since the two layers with the smallest multitasking capability constitute a bottleneck for the entire network (Alon et al., 2017). Note that the detrimental effect of depth on multitasking capability stands in contrast to the benefit of depth for the learning of complex functions (Goodfellow et al., 2016; Simonyan & Zisserman, 2014; Telgarsky, 2016). Altogether, these analyses suggest that high amounts of representation sharing between tasks, paired with a high number of processing layers may be sufficient to yield significant limitations in multitasking capability, even in neural architectures with high a structural and representational capacity such as the brain.

An potential appeal of using neural network architectures to understand constraints on processing is that, in principle, they can be tested by directly examining brain function. Unfortunately, in practice, though both representational mapping and connectomics have become important areas of progress in neuroscientific research, current methods are not sufficient to provide a precise measure of the constructs corresponding to those used in our analyses. That is, it is still not possible to reliably distinguish pools of units responsible for each dimension of processing in a task, for multiple tasks, and at the same time the patterns of synaptic connectivity among them.

Nevertheless, suggestive lines of evidence are beginning to appear.

For example, analyses of functional networks of the macaque cortex, that treat distinct brain areas as nodes and inter-cortical tracts connecting them as edges, yield node degrees ranging from 20 to 40 (Sporns, Honey, & Kötter, 2007; see also Rubinov & Sporns, 2010; Young, 1993). If different brain areas are assumed to represent different forms of information, and the tracts between them correspond to processing pathways used for task execution, then the estimated node degrees are in a range for which we observed asymptotically low multitasking capabilities. Of course, as noted, such findings are at an extremely coarse grain of analysis, and allow for the obvious possibilities that a given brain area may support multiple distinct pools of representations, and that connections among them could remain distinct within intracortical tracts. More detailed studies are needed to directly quantify structural overlap between task pathways, including ones of the human brain. An important factor to consider in such studies is the *distribution* of node degrees, as the analyses we report suggest that multitasking limitations are sensitive not only to the density, but also to the entropy of connectivity in a network. It will, of course, be equally important to relate such factors to task performance, as considered below and in Part II of this article.

2.4.3 Application of Analytic Methods to Prediction of Multitasking Capability. The results of Simulation Study 1 indicate that it is possible to estimate the multitasking capability, and predicted multitasking performance of a network based solely on measures of similarity among representations associated with individual tasks. These methods are of a form that it may also be possible to apply them to the analysis of brain activity, to predict multitasking performance in humans and perhaps even other species. For example, if patterns of neural activity (measured using direct neuronal recordings and/or fMRI) can be identified for a set of individual tasks, then the analyses described above can be used to predict multitasking performance for all combinations of those tasks. This might be impossible to determine directly (i.e., by measuring performance for all task combinations individually), as the number of combinations grows factorially with the number of tasks (for example, with just five input and five output dimensions, from which 25 tasks can be formed, the number of multitasking combinations is over 1500). In contrast, the methods we have described require measuring only the pattern of activity associated with each task individually, which grows linearly with the number of tasks. That is, these analyses may be particularly useful in situations in which exhaustively assessing the entire space of task combinations is empirically intractable (e.g. combinations of tasks that can be performed in a pilot cockpit).

The application of graph-theoretic methods to analyze connectionist models in particular, and neural systems more broadly, is still early in its development, and requires making simplifications. An important simplification in our analyses, that could be relevant to its use in empirical applications, is the thresholding of real-valued correlations among task representations in order to construct the binary bipartite and dependency task graphs used to determine multitasking capability. As we noted above, simulation results suggest that the methods are robust across a wide range of thresholds and learned task representations (see Petri et al., 2020). Nevertheless, generalizing the methods to address graded interference effects (e.g., using weighted graphs) is an important avenue for future research. More generally, it will be important to explore the extent to which these methods can be extended to networks with more complex and realistic architectures (e.g., recurrent networks, or ones subject to more complex dynamics such as gating).

2.4.4 Relationship to Response Time Methodology. As discussed above, sophisticated mathematical methods have been developed for using measurements of response time distributions to infer the extent to which performance of a task relies on parallel processing (e.g. Townsend & Altieri, 2012; Townsend & Wenger, 2004), based on Systems Factorial Design Technology (Townsend & Nozawa, 1995) and theoretical results concerning RT inequalities for independent information channels (Colonius & Vorberg, 1994; Grice, Canham, & Boroughs, 1984; Grice, Canham, & Gwynne, 1984; J. Miller, 1982). Applications of these methods to paradigms such as short-term memory search (Townsend & Fifić, 2004), visual search (Fifić, Townsend, & Eidels,

2008) and the Stroop task (Eidels, Townsend, & Algom, 2010) have generated insights into the extent to which mental processes rely on parallel vs. serial processing. The approach presented here complements this work in several ways. Like these behavioral measures, the approach presented here provides a means for estimating multitasking capability when the underlying task structure is not known. However, here we suggest how this can be done by measuring internal representations engaged by the individual tasks rather than behavior. We have demonstrated this in artificial neural networks, and suggested how it might be applied empirically (e.g., using patterns of activity measured from direct neuronal recordings or neuroimaging methods such as fMRI). Second, while the analysis of RT distributions requires measurements for *every combination* of the tasks of interest — which, as noted above, can rapidly become impractical for even modest numbers of tasks — the methods we have described can be used to predict multitasking capability and performance from measurements made of *each task individually*, which may be more practical in realistically complex task settings. Third, our application of the methods used to analyze response time distributions to neural network simulations shows that, although the derivation of those methods was based on assumptions of linear processing, they appear to apply reasonably well to non-linear processing mechanisms and distributed representations commonly used in neural network models, comporting both with predictions made by our graph theoretic methods and direct measures of multitasking accuracy. Finally, and perhaps most importantly, while the two approaches offer complementary ways to infer multitasking capability from empirical data, the simulation studies presented here also sought to identify and examine the influence of a causal factor — shared representations — that determines the multitasking capability of a system. In this respect, we hope that our findings contribute to providing a "*linkage of quantitative concepts [. . .] with neural mechanisms*" (Townsend & Wenger, 2004, p. 1016).

One apparent disparity between the degree of parallelism estimated from RT distributions and predicted from the analysis of shared representations was the observation, in Simulation Study 2, that the RT bounds of the independent channels model (Colonius & Vorberg, 1994) were violated for tasks that did *not* appear to share representations in the hidden or output layer of the network. While this might be taken to suggest that the RT bounds for a linear independent channels model may not generalize fully to non-linear systems, we observed that after multitasking training, cumulative RT distributions fell within the predicted bounds of the independent channels model. This suggests that methods based on this model can be usefully applied to the behavior of network with nonlinear processing units, at least of the sort used in our simulations and, furthermore, that it is sensitive to sources of cross-task interference that can arise between tasks that are not detected graph theoretic analysis of shared representations at the hidden and output layers. One source for such interference is interactions mutual inhibition of response dimensions among tasks that arises during single-task training, but is diminished with multitask training (see Simulation Study 1 for a discussion). While the graph theoretic analyses we described here were not sensitive to this, it is possible that those can be extended, and or other similar measures developed that are able to detect such interactions from internal representations (Bernardi et al., 2018; Henselman-Petrusek et al., 2019; Chung, Lee, & Sompolinsky, 2018). It is important to note that, irrespective of methods of analysis, such interference at the output layer is consistent with the general proposition that limitations in multitasking performance, and the concomitant need for control, reflect local competition among task-specific representations (in this case, at the output layer of the network) rather than a limitation in the capacity for control *itself.*

2.4.5 Dual-Task Interference and the PRP. A large body of empirical work on dual-task interference suggests that limitations in multitasking can extend to situations in which two tasks are executed in sequence (Koch et al., 2018; Pashler, 1994; Salvucci et al., 2009). One of the hall-marks of dual-task interference is the PRP, a period during which processing of a second task is delayed because a first task is still being processed (Telford, 1931). The PRP was an explanandum for some of the earliest theories of modern cognitive psychology, in which the processing delay for the second task was interpreted as evidence of a central information processing bottleneck that limits processing to only one task at time (Broadbent, 1957, 1958; Welford, 1952). Pashler (1994) introduced a refinement of this theory, suggesting that the central bottleneck occurs at an intermediate processing stage that excludes stimulus perception and motor response production, referred to as "response selection." Kieras and Meyer (1997, p. 4) offered a definition of this as a *"process that converts the stimulus code to an abstract symbolic code for a physical response based on some set of innate or previously learned stimulus-response associations."* Note that the two-layer network used in Simulation Study 1-3 (see Fig. 9) implements this process, by mapping stimulus codes in the stimulus input layer, through an internal, distributed representation in the hidden layer, to a representation in the output layer. Thus, the processes encoded in the 3-layer network follow a perceptual processing stage and precede a motor production stage, and can be summarized as an intermediate, "response selection" stage. However, unlike the response-selection bottleneck account (Pashler, 1994), the neural network model allows that: (1) structurally, the process of response selection can occur in parallel for two or more tasks, albeit with the potential for interference (i.e., a shared representation may integrate incongruent information from the task processes); responses to the second task may be strategically delayed to avoid interference from the first task, mimicking a bottleneck. Accordingly, the PRP depends on the amount of interference induced by shared representation. That said, this does not preclude the possibility that dual-task interference can arise at other points in processing or for other reasons; for example competition among task representations, a possibility to which we will return below.

More importantly, the original suggestion that a bottleneck in the response selection stage of processing is responsible for the PRP assumed that this was modality-general, and thus closely related to if not identical to the idea that the PRP reflects a constraint in a centralized processing mechanisms. In contrast, the neural network models described here align more closely with accounts that build on the multiple resource theory, suggesting that processing bottlenecks responsible for the PRP lie in local, task-specific resources (Byrne & Anderson, 2001; Meyer & Kieras, 1997b; Navon & Gopher, 1979; Salvucci & Taatgen, 2008). However, previous theories have generally implemented these resources in production system (symbol processing) architectures as discrete, predefined sets of processing modules. Here, we did so using neural network models based on the parallel distributed processing framework (McClelland et al., 1986), in which task-specific resources are representations that can learned, engaged in a graded way (based on the strengths of connections in the network), distributed across multiple processing units that permit varying degrees of overlap, and have persistence characteristics that can also cause processes to overlap in time — features that are generally thought to be characteristics of computation in the brain. These can explain a number of effects that have not been — and might not be easily — addressed using strictly symbolic approaches. For example, while some studies have observed that the PRP effect at an SOA of 0 matches the RT of the first task (e.g., Welford, 1952), as predicted by central bottleneck models, others have reported a smaller-than-predicted PRP (Karlin & Kestenbaum, 1968). Simulation Study 3 showed that the PRP can match the RT of the first task if it and the second task are functionally dependent and there is a high amount of persistence in the network. However, the PRP can be lower if the tasks are only partially dependent or if persistence is low (see Fig. 18B). Conversely, longer persistence of shared representations can explain a PRP (delayed execution of a second task) that exceeds the RT for the first task (Welford, 1952; Marill, 1957). That is, the response to the second task can be slowed even if the stimulus for the second task is presented after response to the first task; something that discrete, symbolic processing mechanisms might find difficult to explain (Pashler, 1994).

The neural network implementation also provides a natural and quantitative account of how the number of tasks that a system can perform may impact its multitasking abilities, as well as the effects of practice. We discussed the quantitative analyses of multitasking capability above. With regard to practice, Simulation Study 3 replicated the finding that the PRP can be eliminated with sufficient practice on dual-tasking (Hazeltine et al., 2002; Liepelt et al., 2011; Schumacher et al., 2001). Central bottleneck models have proposed that this reflects a reduction in preparatory demands for both tasks (Pashler, 1994), and/or shortens the central processing stage (Ruthruff, Johnston, Van Selst, Whitsell, & Remington, 2003). Neural network models offer potential mechanisms for these effects of practice; for example, increasing the strength of each processing pathway could reduce integration times and thus the effects of persistence, and/or accelerate their engagement by control. Here, however, we have focused on a qualitatively different effect of dual task practice, that is specific to network architectures and more closely related to the multiple resources account: that this can lead to the separation of representations between tasks — an effect to which we will return in detail in Part II.

The graded nature of representations in neural network architectures, and their potential for overlap in both space and time, also provides a mechanistic grounding for other accounts of dual-task interference in terms of "dimensional" (Liepelt et al., 2011; Hazeltine et al., 2006) or "representational" (Göthe et al., 2016) overlap. Here, "dimensional" or "representational" overlap can be defined in terms of the degree to which tasks share representations that may induce structural or functional dependence, and the interactions that this has with the persistence characteristics of those representations. These factors also make a number of novel predictions. For example, they predict that functionally dependent pairs of tasks should be associated with a longer PRP compared to independent pairs of tasks. They also predict a longer PRP for tasks that rely on representations with longer persistence characteristics, such as tasks that require integration of information over longer periods of time (Hasson et al., 2015).

Despite the implementational differences between our approach and ones using symbolic processing mechanisms to implement the multiple resource theory (see Section "Relationship to Existing Theories of Dual-Task Limitations" in the General Discussion), these approaches agree in at least two fundamental ways: (1) that PRP effects are driven by the potential for local conflicts in processing, and (2) that these are avoided by strategically delaying the second task to prevent interference from first. This was first described as strategic response deferment (SRD) within the EPIC framework by Kieras and Meyer (1997); Meyer and Kieras (1997b), in which a response to the second task could be deferred by an executive (control) mechanism until after sufficient progress had occurred on the first task. Similarly, in Simulation Study 3, response to the second task was deferred by increasing the response threshold of the LCA for that task, to circumvent persisting interference from the first task. In our simulations we further assumed that these adjustments were made in a normative fashion, in order to optimize the joint reward rate for both tasks. More sophisticated algorithms for making such normative adjustments, and neural mechanism that implement them, are the focus of several ongoing lines of work (Lieder et al., 2018; Simen et al., 2009; Shenhav et al., 2013; Westbrook et al., 2020). Such normative adjustments could, of course, also be added to a symbolic processing architectures such as EPIC (Meyer & Kieras, 1997a, 1997b) or threaded cognition (Salvucci & Taatgen, 2008). However once again, such mechanisms are likely to be constrained to making discrete adjustments, whereas their implementation in a neural architecture would permit graded adjustments, and allow these to be learned.

Finally, there is at least one set of observations from the PRP paradigm that the models we have described do not directly address: that performance of the first task can, under certain conditions, be affected by features of the second. For example, Hommel (1998) demonstrated that the RT of the first task can vary as function of compatibility between the response to the first task and the response to the second task. In that study, participants responded to the color (red or green) of a letter stimulus with a button press (left or right; Task 1) before responding to the identity of the letter ("H" or "S") with a verbal response ("left" or "right"; Task 2). Processing of the first task was delayed if the response to the second task (e.g. say "left") was incompatible with the response to the first task (e.g. press the right button). In a different PRP study, Logan and Schulkind (2000) presented participants with two digits. Both tasks required categorizing a digit by its magnitude (i.e. judging whether the digit was larger or smaller than 5). RTs for the first task were faster if the both digits belonged to the same category. Logan and Gordon (2001) proposed a computational model that explains these effects in terms of category-level cross-talk:

The outcome of any categorization process (this may involve categorizations of stimulus features for the first and the second tasks, both of which may occur in parallel) is attributed to the object that is currently given priority (the digit relevant to the first task), leading to a speed-up in processing the first task if the categories for both tasks are compatible. These, and other studies lend support to the claim that the two tasks are being processed in parallel rather than in serial (Ellenbogen & Meiran, 2008; Fischer, Gottschalk, & Dreisbach, 2014; Hommel, 1998; Logan & Schulkind, 2000; Schubert, Fischer, & Stelzel, 2008). The effect described by Hommel (1998) may arise in a neural network model that learns a shared representation between location of the stimulus for Task 1 (left or right) and the verbal response for Task 2 ("left" or "right") in the same (hidden) layer. This may be achieved by training the network to represent the general concept of left and right. Alternatively, feedback connections from the representation of the verbal response in the output layer to the representation of the stimulus location in the hidden layer could introduce to cross-talk from the response for the (second) location-verbal task to the (first) color-manual task. While these possibilities are compatible with extensions of the models we described here, those extensions remain to be implemented and tested in future work.

2.4.6 Performance Costs Associated with Task Switching. The simulations we reported showed that the same mechanisms used to account for the PRP can also explain effects observed in task switching paradigms. Costs associated with task switching — one of the most robust findings in the cognitive literature — have previously been considered in isolation of, and in different terms than the PRP (Koch et al., 2018). One prominent account of switch costs is the task-set inertia hypothesis, according to which the task-set of a previously executed task carries over to the next (Allport et al., 1994). Simulation Study 3 provides a mechanistic interpretation of this hypothesis, in which the task-set is represented as patterns of activity over the hidden and output layers of the neural network,[30] its inertia corresponds to the persistence of

[30] In connectionist systems, a task-set can be defined as the "internal state of the network at a given time that biases it to respond to a multivalent stimulus configuration" (Grange & Houghton, 2014, p.

those representations, and switch costs arise as a consequence of the interaction between the extent to which the patterns of activity are shared with the next task to be performed, and persist during its performance. This suggests that switch costs should scale with (1) the amount of shared representation between tasks and (2) with their persistence in the network. Simulation Study 3 demonstrated that these effects provide a mechanistic account for a number of widely replicated findings in the task switching literature, such as greater costs associated with incongruent stimuli on a switch between tasks that use the same (bivalent) responses (e.g., Fagot, 1995; Goschke, 2000; Meiran et al., 2000; R. D. Rogers & Monsell, 1995; Wendt & Kiesel, 2008), as compared to tasks using distinct (univalent) responses (e.g. Brass et al., 2003; Meiran et al., 2000; R. D. Rogers & Monsell, 1995).

The model also makes novel predictions with respect to switch costs for tasks with univalent responses. The simulation results indicated that: (1) tasks with univalent responses should exhibit greater switch costs if they are functionally dependent relative to independent tasks; and that (2) tasks with univalent responses may be sensitive to response congruency. For instance, in the extended Stroop task (see Fig. 5A), color naming is predicted to be functionally dependent on word mapping, but not on word reading. Thus, switching from word mapping to color naming may require more time than switching from word reading to color naming.[31]. Moreover, when switching from word mapping to color naming, the model predicts a higher cost of switching for incongruent Stroop stimuli compared to congruent Stroop stimuli, since incongruent stimuli would lead to stronger functional interference.

The assumption that task representations persist in time, and that the persistence of a previously activated task-set leads to a benefit of task repetitions over task switches is certainly not unique to neural network models. Symbolic models, for example ones based on ACT-R (Anderson & Lebiere, 2014), explain a portion of switch costs in terms of repetition priming of task-relevant information in declarative memory: Recently

180-181).

[31] This assumes that word reading and word mapping are comparable in performance

activated task-sets[32] are more likely to be retrieved from a declarative memory buffer, leading to a facilitation of task repetitions (Altmann & Gray, 2008; Sohn & Anderson, 2001). One important difference between the effects of persistence in symbolic and neural network models is that, in the latter, persistence characteristics can interact with distributed representations, and thus have graded effects determined by the degree of representational sharing — a characteristic that is ripe for investigation in domains where distributed representations have played a critical explanatory role, such as semantics (we will return to this in the General Discussion). However, even within the scope of neural network models, there are conceptual differences with regard to where persistence may occur. For example, some neural network models of task switching assume that task-sets persist in the form of stimulus-response associations that are updated each trial (Brown et al., 2007; Gilbert & Shallice, 2002), while others attribute this to the persistence of task-related activity patterns (e.g., Herd et al., 2014, and the model we report here). It remains an important avenue for future research to tease apart the different types of persistence, and their contribution to performance costs in task switching.

Finally, we note that the models described above were not intended to address a number of other important task switching phenomena, such as repetition priming effects of task cues (Altmann & Gray, 2008; Logan & Bundesen, 2003; Sohn & Anderson, 2001). We suspect that adding the elements to the model necessary to address such effects (e.g. processing units that represent task cues), coupled with the features we have described (such as persistence characteristics), may be sufficient to address such phenomena. Nevertheless, these too remain as targets for future work.

2.4.7 Broader Implications. Altogether, Simulation Studies 1-3 suggest that an interaction between (1) the potential for conflict introduced by shared use of representation between tasks, and (2) the persistence of task representations over time, define a continuum in the extent to which a set of tasks be executed in parallel (i.e.,

[32] In symbolic architectures, a task-set often corresponds to task-relevant chunks (e.g. chunks that map stimuli to particular responses) in declarative memory.

concurrently multitasked), permit rapid switching, or require full sequential execution (i.e., “serial processing”). Furthermore, insofar as control mechanisms are responsible for regulating the execution of a task in order to mitigate the conflict that can arise from parallel or overly rapid serial execution, then these factors also define a continuum in the extent to which a task must rely on control (i.e,. its “automaticity”), as a function of the context (i.e., the other tasks in contention) in which it must be executed. We have shown that this perspective can provide a quantitative grounding of the multiple resource theory, including the influence that the number of tasks that share representations in a network has on its multitasking capability; as well as a unifying account of two sets of phenomena classically associated with control-dependent processing, but previously considered largely independently of one another: the PRP and task switching costs.

Intriguingly, this perspective predicts that there should be a relationship between the performance costs associated with dual-tasking (such as the PRP) and those associated with task switching, as a function of the extent to which the tasks involved share representations (i.e., are structurally or functionally dependent). Although, to our knowledge, there has not yet been a direct empirical test of this prediction, modality-specific effects in both dual-task and task switching paradigms suggest such a relationship (Stephan Koch, 2010). For example, several studies have reported smaller dual-task interference for pairs of tasks with compatible stimulus response mappings (e.g. a visual-manual task paired with an auditory-vocal task) compared to tasks with incompatible stimulus-response mappings (e.g. a visual-vocal task paired with an auditory-manual task; Greenwald, 1970; Greenwald & Shulman, 1973; Göthe et al., 2016; Halvorson et al., 2013; Hazeltine et al., 2006; Liepelt et al., 2011; Shaffer, 1975). Similarly, Stephan and Koch (2010) found that participants can switch faster between pairs of tasks with compatible stimulus-response mappings relative to pairs of tasks with incompatible stimulus-response mappings, and that this effect diminishes as the time between the last response and next stimulus increases, suggesting that the interference induced by modality compatibility ceases to persist in time.

Finally, it is worth noting that approach taken here may resolve a longstanding puzzle concerning the relationship of empirical evidence for a response-selection bottleneck in dual-tasks experiments (e.g., the PRP) with the classic interference effect observed for color naming of incongruent stimuli in the Stroop task. Keele (1973) pointed out that the latter is difficult to reconcile with evidence for a response selection bottleneck in dual-tasking: If the responses for two tasks cannot be selected at the same time in dual-tasking scenarios, how could the color naming response be influenced by the response associated with the word stimulus in the Stroop task? Pashler (1994, p. 237) addressed this paradox, suggesting that *"[. . .] recent investigations of neural networks suggest some possible ways of reconciling the two lines of evidence. Consider, for example, so-called "pattern completion networks" composed of simple units connected with variable strengths. Selection of one response may involve a particular pattern of activity emerging in some subset of the units, whereas selection of a different response involves producing a different pattern in the same units. Putting different inputs into such a network might involve activating different subsets of units. The network could not select two different responses at the same time simply because the output units could not settle into two different states at the same time. On the other hand, different input units could be activated at the same time [. . .]. If the irrelevant input was associated with a different response than the relevant one, it could retard the process of settling into a final output state".*

The neural network models described above provide a mechanistic implementation of this account: Shared representations in the hidden layer pose the risk of cross-talk between tasks, leading to the simultaneous activation of competing output states for those tasks. Resolving this competition results in a delayed response, providing an explanation for Stroop interference, as well as the PRP in dual-tasking scenarios. Critically, Pashler (1994, p. 237) pointed out that such an account would rely on assumptions about the nature of task representations: *"One unattractive feature of this explanation is that there is no independent motivation for supposing that different outputs would be represented in the same units and different inputs would be represented* *in different units"*.

In Part II we directly address this concern, showing that interactions between the task environment and learning can provide a normative motivation for the sharing or separation of representations between tasks.

3 Part II: Shared vs. Separated Representations and Learning Efficacy vs. Processing Efficiency

The findings reported in Part I support the proposition of multiple resource theory: that limitations associated with control-dependent processing reflect cross-talk that arises from the sharing of representations between task processing pathways — cross-talk that control mechanisms are responsible for managing. However, the assumption of shared resources poses an explanatory gap, as pointed out by Kieras and Meyer (1997, p. 11): *"One [. . .] [concern] is that the concept of multiple resources lacks sufficient principled constraints. In the absence of such constraints, there is a temptation to hypothesize new sets of resources whenever additional problematic data are collected. This could lead ultimately to an amorphous potpourri of theoretical concepts without parsimony or predictive power"*.

To address this explanatory gap, Wickens (1991) derived a taxonomy of resources from empirical data, building on the assumption that dual-task interference arises when two tasks share a common set of resources. For instance, it was observed that dual-task interference is higher if two tasks share the same perceptual modality (McLeod, 1977). These and other findings lead Wickens (1991) to conclude that each perceptual modality is associated with a separate, dedicated processing resource. A similar proposal has been made with respect to motor modalities (e.g., Glucksberg, 1963; Treisman & Gelade, 1980; Treisman & Davies, 1973). More generally, Wickens (1991) proposed that task processing resources can be distinguished along four dimensions: processing stage (perceptual vs. central vs. response-related), processing code (verbal vs. spatial), input modality (verbal vs. auditory), and response modality (manual vs. vocal). Similarly, McCracken and Aldrich (1984) proposed a segmentation of resources into visual, auditory, cognitive and psychomotor components, each representing a local resource that may be shared with other tasks.

Computational implementations of multiple-resource theories, such as the EPIC framework (Meyer & Kieras, 1997a, 1997b) and threaded cognition (Salvucci & Taatgen, 2008), adapted the resource taxonomy by Wickens (1991) and others to define shared resources. For example, EPIC assumes distinct processors for auditory and visual inputs, as well as vocal and verbal outputs. Kieras and Meyer (1997) argued that perceptual and motor resources are constrained to be operating in serial (i.e. being able to handle only one task process at a time), whereas other, cognitive resources, such as working memory, can be used for multiple tasks in parallel. The theory of threaded cognition assumes the same set of perceptual and motor processes in addition to two cognitive resources, one declarative resource for memory encoding and retrieval, and one procedural resource for coordinating goal-directed behavior (Salvucci & Taatgen, 2008). Similar to Wickens (1991), these instantiations of multiple resource theory motivate their set of resources based on the type of behavioral data that they seek to explain (e.g. a shared resource for visual processing is motivated by the observation that participants fail to perform two visual tasks in parallel).

While the resource taxonomies used by multiple-resource theories enabled initial mechanistic insights into multitasking phenomena, they neither account for the circumstances under which shared representations arise, nor do they provide a rationale for why shared representations, that introduce the need for control, be favored over dedicated representations that render a task independent of others and capable of automatic processing. Addressing this concern requires an understanding of *when* and *why* a cognitive system would develop shared as opposed to separated representations between tasks. As noted by Meyer and Kieras (1997a, p. 68), models such as EPIC *"have chosen to embody [their] theoretical ideas in an architectural production system and symbolic computation, rather than in hypothetical [...] neural mechanisms, simply because the former level of representation is perhaps most appropriate for initially characterizing functional aspects of executive cognitive processes and multiple-task* *performance"*. Here, we suggest that addressing the neural mechanisms — or at least taking account of their computational properties — may in fact be useful in helping to characterize multitasking performance, by providing insights into the processes that give rise to shared vs. task-dedicated representations, and thus reliance on cognitive control versus the development of automaticity. As in many other applications of neural network modeling, such insights derive from considering how learning shapes representations.

It is well established the types of representations learned in neural networks are heavily influenced by the statistics of the environment in which they are trained. In particular, networks are likely to acquire shared representations for different tasks if those tasks share similar statistics (e.g., they involve similar input and/or output representations). This has been studied heavily in the context of semantic tasks, in which the network is presented with physical features of objects and trained to report their functional properties and/or category memberships. Networks develop representations that are shared between semantic concepts if those concepts are statistically related (e.g., Caruana, 1997; Bengio, Courville, & Vincent, 2013; Higgins et al., 2018; Hinton et al., 1986; McClelland & Rogers, 2003; A. M. Saxe et al., 2019). For example, A. M. Saxe et al. (2019) have shown, in formal analyses of network learning, that multi-layer linear networks learn overlapping representations for objects that share features relevant for categorization (e.g. salmon and sunfish) compared to objects that don't share category-relevant features (e.g. salmon and canary). In psychological research, this has been used to explain empirical phenomena (such as semantic priming and similarity judgements; T. T. Rogers & McClelland, 2004), and in machine learning it has been exploited to promote generalization (as we will discuss further below). Interestingly, however, such work has focused almost exclusively on conditions in which the network is required to perform only one task at a time (i.e., is presented with a single stimulus to which it must respond), with almost no attention paid to conditions in which the system is expected to perform more than one task at a time. The results presented in Part I suggest that the propensity to develop shared representations with single task training presents an impediment to multitasking performance which, in turn, incurs reliance on control; they also showed that this can be overcome by training explicitly on multitasking, which enforces the development of separated, task-dedicated representations that permit parallel execution. These observations pose an interesting, and fundamental question: Why should a system favor shared representations, at the expense of a seriality constraint in processing and an attendant dependence on control, over task-dedicated representations that afford the efficiency of multitasking capability.

There are several reasons why *why* a neural system might favor the learning of shared representations. An obvious one is that shared representations are more efficient with respect to storage capacity. While this is certainly a possibility, this seems unlikely to be a strong constraint, considering the enormous representational resources of the brain, and the cost of seriality and dependence on control. A more compelling reason is that shared representations permit transfer to novel tasks; that is, more effective learning through generalization (Baxter, 1995; Caruana, 1997; Bengio et al., 2013). In the domain of machine learning, this often discussed in the context of "multi-task learning," which refers to the ability of an agent to learn multiple different tasks from experience with only limited exposure to those tasks during training. Note that this is distinct from "learning to multitask": the latter requires the agent to perform two or more tasks *simultaneously* (as elaborated in Part I), whereas in multi-task learning the agent is trained to perform a set of auxiliary tasks, one at a time, in addition to a target task. If the auxiliary tasks share similarities with the target task, then exploiting this to learn shared, more general learned representations has been shown to improve acquisition of the target task. These benefits of shared representation are strongly linked to the ability of the network to learn and process representations by simultaneously taking into account a large number of interrelated and interacting constraints among them (McClelland et al., 1986). This allows them to detect and encode complex forms of similarity (e.g., high order correlations among features).

In Part I we showed that shared representations incur the risk of conflict, and thus reliance on control to impose seriality of processing that limits multitasking capability.

In this part of the article, we investigate how these limitations weigh against the benefits of shared representation. Specifically, we describe a set of computational, mathematical and behavioral studies that examine the tension between the efficiacy of learning afforded by shared representations (i.e., transfer to novel tasks) and the efficiency of processing afforded by separated representations (i.e., automaticity and the capability for multitasking) . We begin by examining circumstances that promote shared representation and attendant multitasking constraints in neural networks. Specifically, we use computational simulations to investigate the learning of shared representations between tasks as a function of (1) shared structure between tasks in the environment, and (2) the effects of of training on single task (serial) vs. multitasking (parallel) performance. We then describe a combination of mathematical analysis and computational simulation to further characterize the tension between between the learning of shared versus separate representations, and report the results from a behavioral experiment that tests predictions of these theoretical studies. Finally, we discuss a normative theory of multitasking, which suggests that constraints on multitasking may reflect a preference for learning efficacy (i.e., transfer) over performance efficiency (i.e., multitasking).

3.1 Learning Biases for Shared vs. Separated Representations

There are various external factors that can bias a neural system towards shared vs. separated representations. Here, we use computational simulations to demonstrate that shared representations are function of both the task environment, as well as the training regime. In the first simulation study, we examine the effect of structural overlap between task-relevant stimulus features on the learning of overlapping representations between tasks. In the second study, we investigate the differential effects of single vs. multitasking training on the learning of shared vs. separate task representations.

3.1.1 Simulation Study 4: Impact of the Task Environment on the Development of Shared Representations. A key feature of neural network architectures is their ability to discover latent structure in the training environment, exploiting similarity between stimulus features in the form of shared representations (Bengio et al., 2013; Caruana, 1997; Hinton et al., 1986; A. M. Saxe et al., 2019). This has been clearly shown to benefit learning efficacy through transfer (Baxter, 1995; Caruana, 1997). Furthermore, formal analyses of learning in networks with linear processing units have characterized interactions between the statistical structure of the training environment and the emergence of representations, showing that the most widely shared features (e.g., corresponding to the highest level, or broadest categories) are learned faster than features shared more narrowly (corresponding to lower level or more specific categories (A. M. Saxe, McClelland, & Ganguli, 2013; A. M. Saxe et al., 2019). For example, in learning about living things, the distinction between plants and animals is learned more quickly than the distinction between different kinds of plants or animals.

While this work focused mostly on inference (e.g., object categorization and the learning of semantics), the network architectures and learning mechanisms involved are homologous to those used in Part I to address tasks involving actions, and thus the same principles should apply. Interesting, however, work in the domain of inference did not consider how the acquisition of shared representations effects performance efficiency (i.e., multitasking; e.g., the ability to recognize more than one object at the same time). Here, we replicate the findings concerning the acquisition of shared representations referred to above, using the neural network model described in Part I, and then extend this work to directly examine the impact that shared structure between tasks and the development of shared representation has on both the speed of learning and on multitasking performance.

Network architecture. We used a variant of the network architecture described in Part I, that allowed us to examine a graded range of similarity structure of the stimuli within subsets of tasks. The input layer consisted of 54 units, nine of which were used to represent the current task, and 45 to represent the current stimulus. As before, tasks were coded as binary "one-hot" vectors: a single unit was assigned to each task, with the unit for the current task assigned a value of 1, and all other units assigned 0.

Similarly, a different set of five stimulus features were assigned as being relevant to each of the nine tasks, corresponding to the stimulus dimensions assigned to each task in the models described in Part I. However, whereas in Part I we simulated environments made up of three subsets of three tasks each, in which the same set of features was relevant to all three tasks within a given subset, here we simulated environments that differed in the degree to which tasks within a given subset shared features (see below). To implement the degree of sharing in a continuous manner, input patterns were continuous valued (rather than binary, “one-hot”) vectors, each unit of which was assigned a value between 0 and 1. These input patterns were used to define different tasks, as described below. The remainder of the network was configured in a manner similar to those described in Part I: the hidden layer consisted of 100 units; and the output layer consisted of 15 units organized into three response dimensions of five units each, in which each response was coded as binary, “one-hot” value.

Task environment. Each task was implemented as a rule that determined how a pattern of activity over five stimulus input units assigned to that task should be mapped to one of the five output units in the response dimension assigned to the task (see Fig. 22). The task rules were randomly generated. Each task rule assigned a pattern over the stimulus input units relevant for that task to a distinct response within the dimension used for that task.[33] Using this procedure, we generated six environments that varied the similarity among tasks within each subset. Similarity was defined by stimulus feature overlap; that is, the number of stimulus input units shared between a pair of tasks within a subset that were associated with different response dimensions. At one extreme (full overlap), resembling the environment used in Part I, the nine tasks were divided into three subsets, with all of the tasks within a subset sharing the same stimulus input units (Fig. 22A and upper row of Fig. 23A); at the other extreme (no overlap), every task was assigned a separate pool of stimulus input units (Fig. 22 B and bottom row of Fig. 23A). In addition, four environments with intermediate levels of similarity were generated by varying the number stimulus input

[33] Task rules were generated such that every output unit was equally likely to be required for execution.

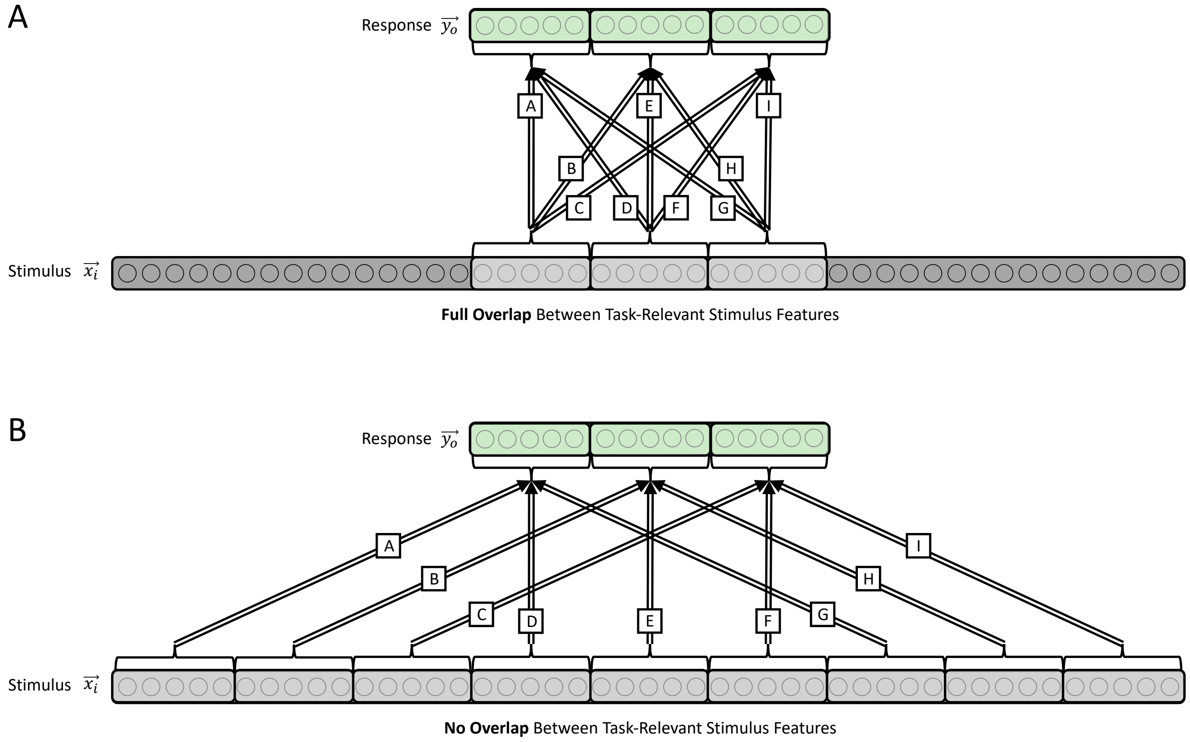
A
Response $\vec{y_o}$
A
E
I
B
H
C
D
F
G
Stimulus $\vec{x_i}$
Full Overlap Between Task-Relevant Stimulus Features
B
Response $\vec{y_o}$
A
I
B
H
C
D
E
F
G
Stimulus $\vec{x_i}$
No Overlap Between Task-Relevant Stimulus Features

Figure 22. **Task environments of varying feature overlap.** Figure illustrates relationships between stimulus features and responses on which the network was trained (i.e., *not* the network itself, which included hidden units and for which connections were learned). For each task, the network was trained to map a subset of five stimulus features onto a subset of five responses. The two panels show examples of extremes of overlap within each set of three tasks (e.g., Tasks A-C). (A) Complete overlap, in which the stimulus features are the same for all the three tasks in each set. (B) No overlap, in which each task within a set uses a distinct set of features.

units shared from 1 to 4 while ensuring that all tasks involved the same number of "relevant" input units (see intermediate rows of Fig. 23A).[34] Note that, despite the sharing of stimulus input units, tasks within a set were structurally independent of one

[34] The task structures defined by these schemes allow tasks to be implemented that do not necessarily align with naturally defined stimulus dimensions (such as shape, size, color, etc.). This accords with the more general, formal definition of a task described in Lesnick et al. (2020), in which a task is defined as a mapping from any set of input features to a set of output features; and allows us to examine how the variation in the similarity structure among inputs — which may be a characteristic of real world tasks, such as semantic ones that involve more complex combinations of features — impacts the structure of the representations learned by a network and, in turn, how that impacts its ability to perform those tasks in parallel.

another insofar each was associated with a distinct response dimension.

Training and analysis. We trained 100 networks using the backpropagation learning algorithm (Linnainmaa, 1970; Rumelhart et al., 1986; Werbos, 1982) in each of the six different task environments described above. The networks were initialized with a set of small random weights and then trained on all nine tasks with the same set of 50 stimulus samples (selected as described above) until the network achieved criterial performance (MSE of 0.01). For each training trial, an input pattern was generated by selecting a task (i.e., activating one of the nine task units), and assigning an activity to each stimulus unit by randomly sampling from a uniform distribution $U[0,1]$. Note that, although the activity of stimulus input units was assigned randomly, the procedure for generating tasks insured that there was a mapping from any arbitrary input pattern in the stimulus dimension for a given task to one of the five output units in the response dimension for that task (see above). Thus, every pattern of activity over the set of stimulus units in the input layer was associated with a fully specified response for each task at the output layer; and, given the procedure for generating these mappings, random sampling of input values insured an equal likelihood of sampling (and generating a corresponding error signal) for each response during training.

Based on previous work reviewed above, we hypothesized that the amount of stimulus feature overlap between two tasks would affect how similar the two tasks would be represented in the hidden layer of the network after training; and, based on the results reported in Part I, this would impact the multitasking capability of the network. As in Simulation Studies 1 and 2 in Part I, we focused our analysis on the weights from each task unit to the hidden layer (see Footnote 24), by computing the Pearson correlation between weight vectors from the two task units to the hidden layer for each pair of tasks. This analysis was restricted to pairs of tasks that are structurally dependent based on some amount of overlap with respect a stimulus dimension but mapped to a different response dimension (e.g. Tasks A and B in Fig. 22), in order to evaluate the extent to which the development of shared representations in the hidden layer could be attributed to similarity structure in the input. Also as in Part 1, we measured multitasking accuracy for the corresponding pairs of tasks by activating the two corresponding task units and evaluating the concurrent processing performance in the response dimensions for the two tasks. Finally, as a measure of learning efficacy, we assessed the average number of training iterations it took to train a network to criterion on all 9 tasks for each environment.

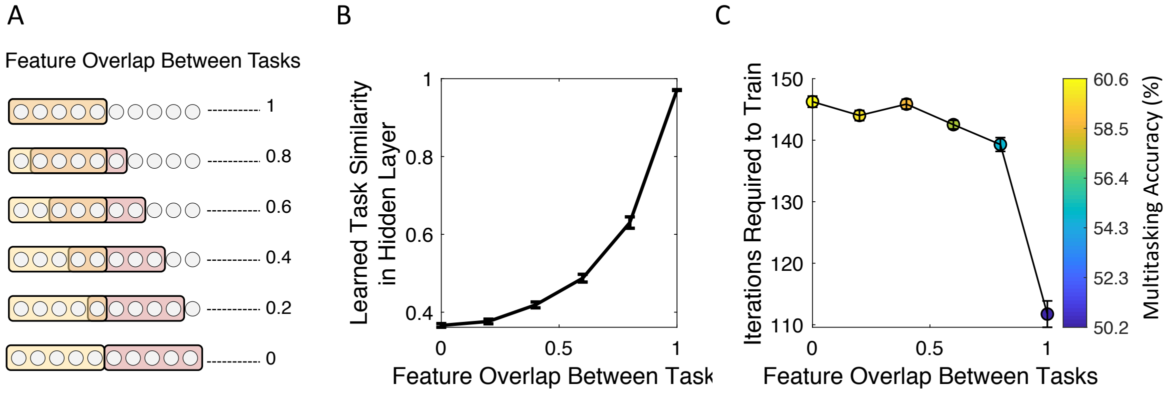

Figure 23. **Effects of task similarity.** (A) Networks were trained in task environments that differed by the number of features shared by subsets of tasks in their stimulus dimensions ("feature overlap"). Yellow and pink shades designate task-relevant stimulus features for each of two tasks within a subset, with orange designating features shared between two tasks (see text). The effects of feature overlap are shown with respect to: (B) average similarity of the learned representations at the hidden layer; and (C) average number of iterations required to train the network to criterion (colors of each data point in Panel C indicate the multitasking accuracy). Vertical bars in (B) and (C) indicate the standard error of the mean across networks.

Results. The simulation results confirm the well-characterized behavior of neural networks trained with backpropagation (Hinton et al., 1986; McClelland & Rogers, 2003; Rumelhart, Todd, et al., 1993); viz., that similarities in the input are encoded as similarities among learned internal representations. This is shown in Fig. 23B, in which greater overlap among stimulus features between tasks within a subset was associated with higher correlation between the vector of weights from the task unit for each task to units in the hidden layer. Critically, greater overlap among stimulus features also promoted faster learning of all tasks, as shared structure between tasks can be exploited in the form of shared representations (Fig. 23C). Interestingly, there is a non-linear relationship between stimulus feature overlap and learning speed, with a substantially greater improvement in the efficacy of learning at the highest levels of overlap. As predicted by the analyses in Part I, we also found that the learning of shared representations progressively degraded multitasking accuracy (colors of dots in Fig. 23C). Thus, this simulation clearly illustrates that similarity in the input among a set of tasks not only shapes the similarity among the internal (hidden) representations learned by a network, favoring the development of shared representations; but, critically, the acquisition of such shared representations has a direct, and graded impact on the network's multitasking accuracy.

3.1.2 Simulation Study 5: Impact of Training Regime on the Development of Shared Representations. The previous simulation showed that, when tasks share similar inputs and the network is trained on tasks one at a time, there is strong bias toward the development of shared representations and concomitant limitations in multitasking capability. However, as discussed in Part I, empirical studies involving dual-task training indicate that participants can overcome such limitations through multitasking training (Hazeltine et al., 2002; Liepelt et al., 2011; Schumacher et al., 2001), an effect that we captured qualitatively in Simulations 2 and 3. These observations indicate that multitasking capability does not depend only on similarity between the tasks, but also on the nature of the training itself. However, the details of this effect have not been well studied, either empirically or in neural networks. This former was noted by Schumacher et al. (2001), after observing that not all participants achieved interference-free multitasking performance after dual-task training: *"Why do some but not all people readily achieve virtually perfect time sharing? Would practice eventually enable everyone to time-share perfectly? Can special training regimens promote this perfection?"* (p. 107). Furthermore, some have suggested that multitasking performance can improve through single task practice alone (Ruthruff, Van Selst, Johnston, & Remington, 2006), while others have argued that multitasking training combined with single task training leads to greater improvements in multitasking performance as compared to single task training alone. For instance, Liepelt et al. (2011) assessed multitasking performance for a verbal-manual task and an auditory-vocal task for two groups of participants. The first group was trained to perform a mixture of single task and multitasking trials over seven sessions (hybrid practice group) and the second group received practice on only single task trials over the same number of sessions (single task group). Multitasking performance, assessed in a final eighth session, was higher for the hybrid practice group compared to the single task group. However, while these studies have provided evidence for the benefits of multitask training on multitask performance (unsurprising in itself), they do not address the mechanisms involved. For example, while some have argued that such benefits reflect improvements in the efficacy of control mechanisms, the results presented in Part I of this article suggest that they result from the learning of separated, task-dedicated representations.

A recent neuroimaging study provided evidence of an association between improvements in multitasking performance and representational separation between tasks (Garner & Dux, 2015). In their fMRI study, Garner & Dux described two training groups. In the experimental group, participants were trained to perform two single tasks in isolation, as well both tasks simultaneously. In the control group, participants were trained to execute a visual search task instead. The authors observed that multitasking training in the experimental group lead to a higher separation of neural representations associated with the two individual tasks compared to the control group. However, the study leaves open the question of which aspects of the training procedure were responsible for the observed effects. For example, the observation of representational separation and concurrent improvements in multitasking, may have been due to the practice on single task executions, training on concurrent processing of both tasks, or both.

Here, we report the results of simulations that characterize: (1) the degree to which the relative amount of multitasking versus single task training has on the development of separated, task-dedicated representations; (2) the shape of its influence on multitasking performance; and (3) which aspects of multitasking training lead to most effective separation of task representations, in particular the extent to which the potential for interference between tasks drove the development of separated representations. We did this by comparing single task training to variable amounts of multitasking training in each of two types of multitasking training regimes: training to execute groups of tasks simultaneously in response to congruent stimuli; and training to execute groups of tasks simultaneously in response to incongruent stimuli. In addition to providing a more detailed characterization of the effects of training on the development of shared representations and multitasking capability, our goal was to generate predictions concerning the dynamics of acquisition that can be tested in future empirical studies.

Network architecture and training environment. The network architecture and processing were the same as those reported in Part I, with the following exception. The number of units in the input and output layers was adjusted to accommodate a task environment with three stimulus dimensions and three response dimensions, and with three features in each dimension. Thus, the stimulus input and output layers each had nine units, and the network could support a total of $3 * 3 = 9$ possible tasks.

Training and analysis. 100 instances of the network were implemented and initialized. We then generated nine copies of each initialized network and applied different training regimes to each. All regimes involved training the network on 500 patterns per training iteration. The nine training regimes were divided into three types: single task (one), multitask congruent (four), and multitask incongruent (four). As in Part I, for the "congruent" conditions, stimuli were chosen such that, for structurally dependent tasks (that is, ones that shared the same response dimension), they were associated with the same response across those tasks (see Fig. 10 in Part I); whereas in the "incongruent" conditions, stimuli were chosen that were associated with competing responses. In the single task regime, all of the training patterns in every iteration were sampled with replacement from the set of all single task training patterns. In the multitask congruent regimes, a proportion of the training patterns was sampled with replacement from all multitasking patterns that involved executing three tasks at the same time using congruent stimuli (either 20%, 40%, 60% or 80%), whereas the remaining proportion was sampled from all single task patterns. In the multitask incongruent regimes, a proportion of the training patterns was sampled with replacement from patterns that involved performing three tasks at the same time using incongruent stimuli (either 20%, 40%, 60% or 80%). Each regime was executed for 1000 training trials.

For tasks trained in each of the three types of regime, we assessed: (1) the average number of training iterations it took to reach an MSE of 0.01 on all single tasks (in the single task regime); (2) multitasking accuracy at the end of training (after 1000 training trials); and (3) how similarity of the hidden layer representations between tasks changed over the course of training (using the similarity measure described for Simulation Study 4). We focused the similarity analysis on task pairs that used the same stimulus dimension since, as suggested by the results of Simulation Study 4, the network should have developed shared representations for those tasks pairs when only trained on single tasks. Similarity was assessed at the end of each training procedure for each of the 100 networks trained using a given regime, and then averaged over all 100 networks for a given training trial and training regime. We visualized the relationship between task representations learned under each training regime using multi-dimensional scaling (MDS). This involved measuring the hidden representation for performing each of the nine tasks alone, and projecting all nine single task representations to a 2-dimensional plane. The projection was performed such that the Euclidean distance between the single task representations was preserved.

Results. As in previous simulations, networks trained only on single tasks yielded poor multitasking performance (Fig. 24A). However, networks trained on single tasking were able to acquire all single tasks much faster than the networks trained on multitasking (Fig. 24B). As expected, an increase in multitasking training also yielded better multitasking performance at the expense of slower acquisition of single tasks.[35]

[35] Note that neither multitasking training on congruent stimuli alone, nor multitasking training on incongruent stimuli alone yields perfect multitasking performance, as the multitasking performance is assessed across the set of all congruent and incongruent stimuli.

Critically, all three effects were stronger when multitasking training was performed with incongruent stimuli as opposed to congruent stimuli. The effects of the different training regimes on the learning of shared representations is clearly observed in the MDS projections of the patterns of activity for the hidden layer of each network (Fig. 25). For single task training (upper left panel), the representations project perfectly into three points, one corresponding to each stimulus dimension, confirming that tasks that shared a stimulus dimension developed extremely similar hidden unit representations (as was observed in the correlations reported for previous simulations). As the proportion of multitasking increased, representations for different tasks showed progressively more separation; however, this effect was considerably less for the congruent than the incongruent conditions. The persistence of clustering by stimulus dimension in the congruent condition even at the highest levels of multitasking training, and a similar trend even in the incongruent condition, indicates a strong bias toward shared representation. Nevertheless, at the highest levels of multitasking training with incongruent stimuli, the network develops fully separated representations, indicated by distances among them that are roughly equivalent for tasks associated with the same and different stimulus dimensions.

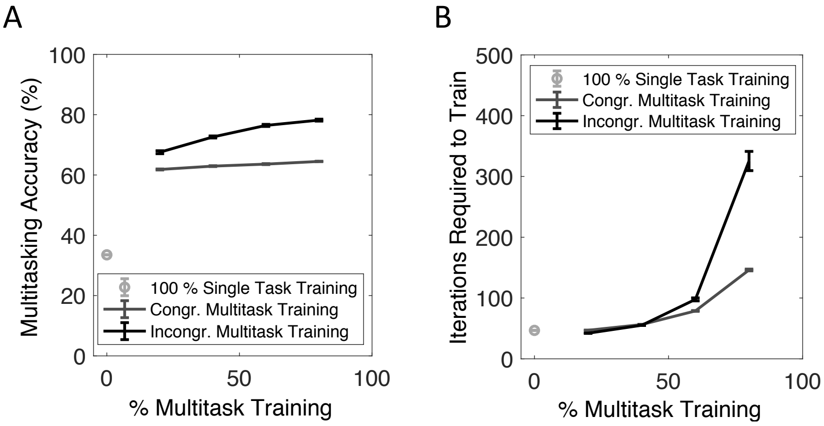

Figure 24. **Effects of training regime on performance.** (A) Average multitasking accuracy and (B) iterations of training required for networks to achieve criterial single-task performance (MSE = 0.01 across all tasks individually) as a function of the proportion of multitasking training (abscissa) for each of the three training regimes (shades of gray — see legend, and see text for explanation of regime types). Vertical bars indicate standard errors of the mean across networks.

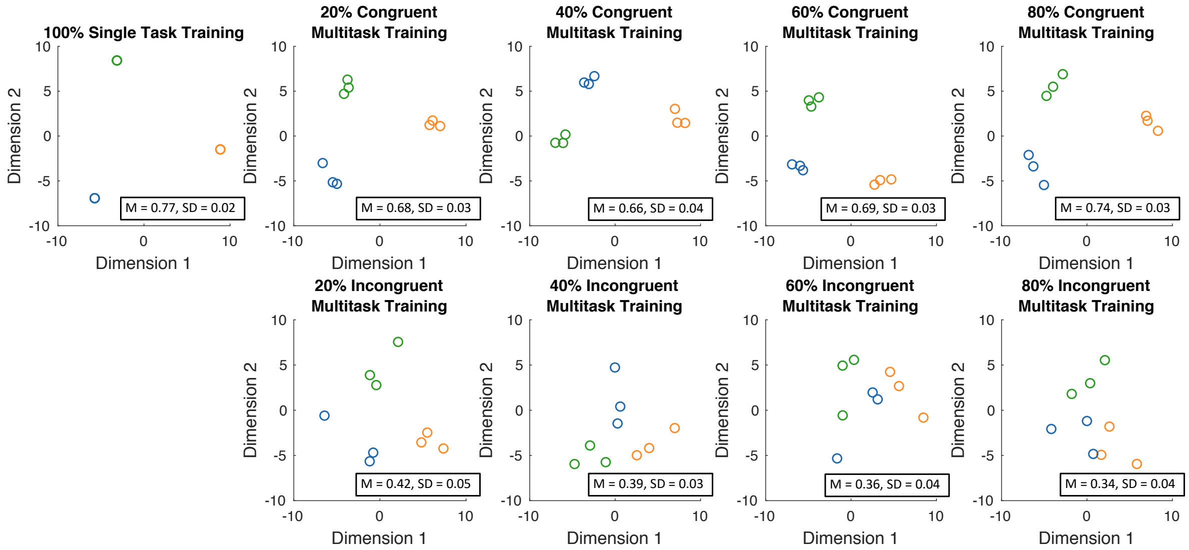

Figure 25. **Effects of training regime on representational separation.** Projections of hidden representations for each task in example networks trained with varying proportions of multitasking. For each network, MDS was used to make projections onto a 2-dimensional plane while maintaining Euclidean distances between its hidden unit representations for each task. Each panels depicts the projections for an example network trained with each of the nine training regimes; each point depicts the hidden unit representation for each of the nine tasks in a regime; and colors depict representations for tasks associated with the same stimulus dimension. Note that in the 100% Single Task Training regime there are in fact 9 dots, but all three for each input dimension are fully overlapping, indicating fully shared representations. Insets correspond to the mean (M) and standard deviation (SD) of the average Pearson-correlation between the hidden unit representations of tasks that are associated with the same stimulus dimension.

Shared vs. Separated Representations and the Tradeoff between Learning and Processing Efficiency

In the preceding section we investigated the conditions under which networks favor the development of shared versus separated representations, showing that shared representations are learned more quickly and that there is a bias toward doing so even under conditions of modest exposure to multitasking training. Here, we turn to detailed analyses of how this impacts the trade-off between the efficacy of learning efficacy provided by shared representations and the efficiency of processing provided by separated representations. We begin by presenting a mathematical analysis that builds on exact solutions to learning dynamics in deep *linear* networks (A. M. Saxe et al., 2013), that we apply to the tradeoff between learning efficacy and processing efficiency in such networks. We follow this, in Simulation Study 6, with a validation of the results of that analysis in simulations involving *non-linear* networks. Then, in Simulation 7, we illustrate how shared representations of learned tasks can facilitate transfer to novel, related tasks. Finally, we report results from an empirical study using the extended Stroop paradigm that tests predictions from these analyses.

3.1.3 Mathematical Analysis: Tradeoff Between Learning Efficacy vs. Processing Efficiency in Linear Networks. To analyze the tradeoff between shared and separated representations in neural networks, we introduce a simplified version of the networks considered in the previous sections, that uses linear processing units. As part of this simplification, task units and their projections to the hidden and output layers are replaced with “gating signals” that regulate the activity of units in the hidden and output layers (as described below). With these simplifications, the dynamics of learning for the mapping of stimuli to responses for sets of tasks can be solved exactly using methods developed by A. M. Saxe et al. (2013).

The simplified model, shown in Fig. 26, consists of stimulus and response dimensions in the input and output layers, respectively. As in the models described in Part I (cf. Fig. 3), units in the hidden layer are separated into sets corresponding to stimulus dimensions, and sets in the output layer corresponding to each response dimension. We analyze two types of such model: one with full sharing of stimulus input representations in the hidden layer (i.e., the minimal basis set representation, Fig. 26A), and one with full separation (i.e. the tensor product representation, Fig. 26B). Unlike the models described above, hidden and output units use linear rather than non-linear activation functions. Furthermore, tasks are specified by gating the activity in sets of hidden and output units corresponding to task-relevant dimensions. Specifically, the activity is zeroed for all units in all sets at the hidden layer except those that receive input from the task-relevant stimulus dimension(s); similarly, activity is zeroed for all units in all sets at the output layer except those corresponding to the task-relevant response dimension(s). The activity of units in sets corresponding to task-relevant relevant dimensions is allowed to "pass through."

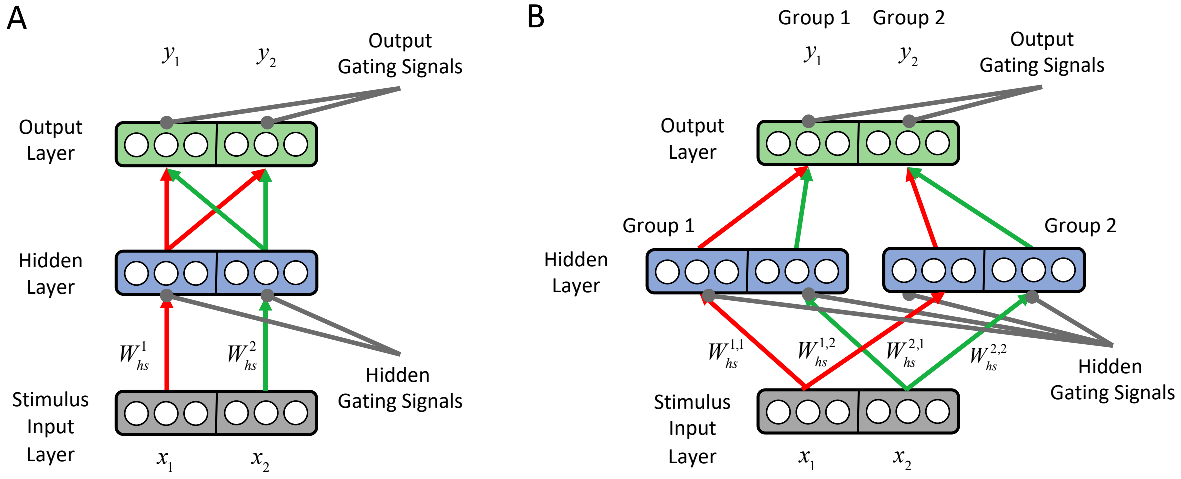

Figure 26. **Gating model used for mathematical analysis of the tradeoff between the learning efficacy vs. processing efficiency.** (A) Network with shared representations in the hidden layer for tasks associated with the same stimulus dimension (minimal basis set representation). Since the same input-to-hidden weights are used for the M different tasks associated with a given stimulus dimension, this increases learning speed by a factor $\sqrt{M}$ relative to learning the tasks with separated representations as shown in (B) (see text). However, in this configuration, functional dependence prevents two tasks that rely on different stimulus dimensions to be performed at the same time, due to crosstalk at the output layer (convergent red and green arrows). (B) Network with separated representations, grouped by output representations (tensor product representation). As elaborated in Part I, dedicating separate hidden units to each individual task allows tasks associated with different stimulus dimensions to be performed simultaneously, as long as they also don't share a response dimension (also see Fig. 3 and Fig. 5); here, tasks are grouped by those sharing a response dimension, so that one from each group can be performed at the same time. However, only tasks within a group share weights from the input to the hidden layer, yielding a learning speed of $\sqrt{M/Q}$, where Q is the number of groups (see text).

Crucially, with this implementation, the output of the network is a linear function of units in the task-relevant dimensions (i.e., that are not zeroed). This, coupled with the gating scheme, permits closed-form analysis of the learning dynamics, which amounts to the aggregation of a set of linear solutions across training examples. To illustrate the effects of the gating scheme, consider the network with a minimal basis set representation, in which the input-to-hidden weights for one stimulus dimension are shared by all tasks that rely on that stimulus dimension, and a task that maps the first stimulus dimension to the first response dimension (see Fig. 26A, red). This will rely on the weights W^1_{hs}, mapping stimulus dimension x_1 to response dimension y_1. In a linear network without gating to the hidden layer, the output y_1 could be corrupted by the other stimulus dimension x_2 (Fig. 26A, green), as information from other stimulus dimensions would simply pass through the network with impunity. Furthermore, without gating to the output layer, the network would produce a response in the irrelevant response dimension y_2. As in non-linear networks, we assume that control mechanisms manage such cross talk. To implement a comparable mechanism in the linear network, a gating signal is configured such that irrelevant stimulus dimensions (x_2) are gated off in the hidden layer and the irrelevant response dimensions are gated off in the output layer (y_2), allowing only information from the task-relevant stimulus dimension x_1 to pass to the task-relevant response dimension y_1. The gating scheme can be configured to perform all other tasks in an analogous manner, if these tasks are performed alone. In the minimal basis set representation, gating allows each input-to-hidden weight matrix to be shared across the tasks corresponding to different response dimensions. This leads to a factor $\sqrt{M}$ speedup in learning speed relative to learning the tasks with separated representations (see Appendix C).

While the sharing of representations in the network speeds learning, it impedes multitasking as in non-linear networks. For example, in the minimal basis set configuration shown in Fig. 26A, gating more than one task through to the output will lead to interference due to functional dependence between tasks. As discussed in Part I, this can be mitigated by separating hidden unit representations into sets dedicated to each individual tasks (i.e., tensor product representations), as shown in Fig. 26B (cf. Panel B of Fig. 3). This allows a maximum of Q tasks (i.e., the number of output dimensions) to be performed simultaneously; however, the number of shared weights projecting from the input to the hidden layer is reduced across tasks by a factor Q, which slows learning. These effects can be formalized, providing an analytic expression of the tradeoff between learning speed and multitasking ability as follows:

$$t^2 \propto kQ/M \tag{8}$$

where t is the number of iterations required to learn all tasks, Q is the maximum number of concurrently executable tasks, M is the number of tasks sharing the same stimulus dimension, and k is a proportionality constant that summarizes the statistical strength of the stimulus-response associations for each task, the learning rate, and the performance criterion used to decide when learning is complete (see Appendix C for the derivation and complete form of this expression).

A key observation from this expression is that, as noted above, learning speed increases in proportion to $\sqrt{M}$ — that is, the number of tasks that share the same stimulus dimension. In full nonlinear networks of the sort described in Part I (and used in the simulations below), random initial weights from task units to the hidden and output layers can be thought of as implementing a random sampling of (weak) gating schemes. Equation 8 indicates that gating schemes that can exploit shared representations at the hidden layer will learn more quickly. This should bias networks in which the weights from the task units to the hidden output layers are learned, to develop task weights that induce shared representations at the hidden layer for tasks that share similar inputs. In the section that follows, we test the link between speed of learning and multitasking performance through causal manipulation of representation sharing in non-linear networks.

3.1.4 Simulation Study 6: Tradeoff Between Learning Efficacy vs. Processing Efficiency in Non-Linear Networks. The mathematical analysis of linear networks presented above suggests that the presence of shared representation should result in (1) faster learning of single tasks and (2) decrements (at least initially) in multitasking performance. Simulation Studies 4 and 5 exhibited effects that suggest that these relationships generalize to non-linear networks as well, showing that single task training on tasks with shared structure was associated with the acquisition of shared representation, and that this was accompanied by faster learning and poorer multitasking performance. However, those simulations did not establish a *causal* relationship between the presence of shared representation and the consequences for learning and processing in those networks. That is, faster learning and poor multitasking performance could have resulted from the task environment and training regime alone, irrespective of whether the network learns shared representations for tasks.

To test whether the learning of shared representations is a cause of faster learning in non-linear networks, we biased the network toward learning either shared or separate representations through weight initialization. Architectural biases in artificial systems, such as weight initialization, may correspond to innate constraints of biological neural systems. Thus, studying the effects of architectural biases toward shared representation may yield insights into *why* neural systems like the human brain would prefer representation sharing over representational separation, and may provide a rationale for resulting limitations in multitasking.

Network architecture and task environment. We used the same network architecture and task environments as described in Simulation Study 4. However, here we restricted simulations to three environments, in which tasks were divided into subsets that shared either 100%, 80% or 0% of their stimulus features (see Simulation Study 4). We also added a manipulation of initial task weights as described below.

Training and analysis. To manipulate sharing, we initialized the weights from the task input units to units in the hidden layer ("task weights"), as these determine the amount of overlap between task representations at the hidden layer. Specifically, for each subset of tasks that shared input features, we initialized the task weights within the subset such they had a correlation of r. The weight vectors for tasks of non-overlapping stimulus dimensions were constrained to be uncorrelated. For each of the three task environments described above, we constructed a separate set of networks that varied r from 0 to 0.975 in steps of 0.025. Finally, all task weights to the hidden layer were scaled to be on average five times higher than the weights between other layers in the network.[36]. 100 networks were trained per initialization condition, using the same values for all other parameters as those reported for Simulation Study 4. For every pair of

[36] This was done to enhance the effects of different initial task similarities on learning.

tasks that mapped to different response dimensions, we assessed the similarity between the task weights learned for the two tasks, and the networks' multitasking performance for that pair (see Simulation Study 4). In addition, we assessed the number of learning iterations required to reach training criterion (MSE = 0.01) across all single tasks.

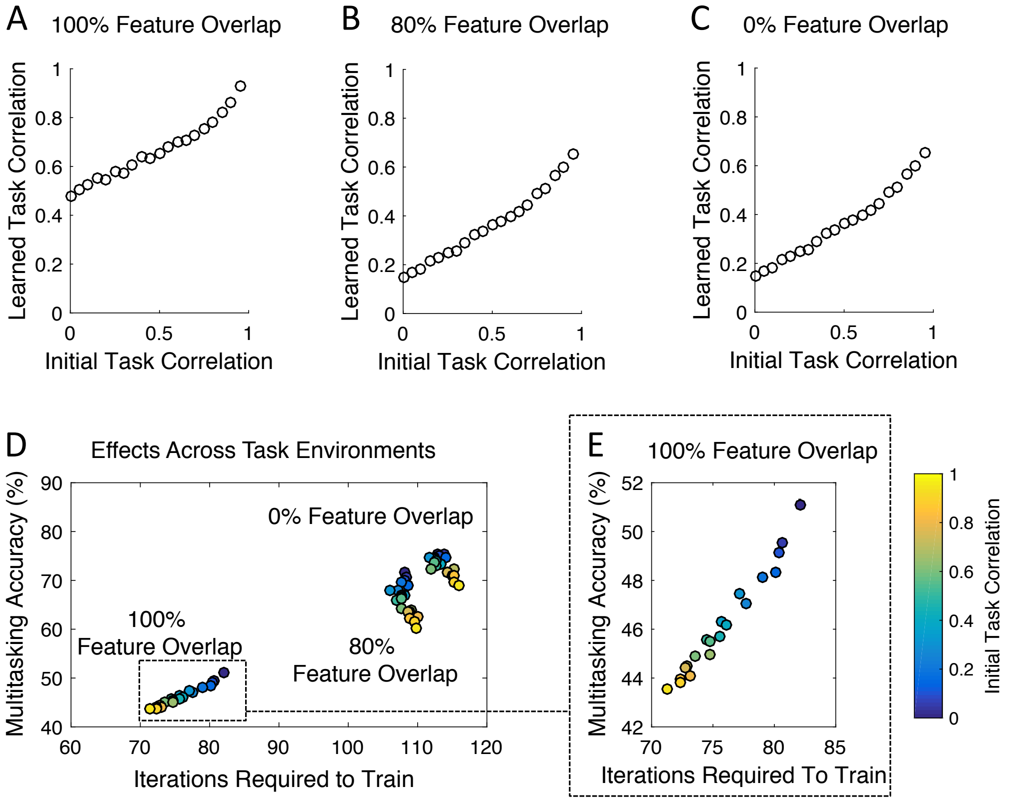

Figure 27. **Effects of bias toward sharing in weight initialization.** (A-C) The average similarity in task weights, after learning, between pairs of tasks in the same subset associated with different response dimensions, as a function of the initial similarity in their weights, for environments with (A) 100%, (B) 80% and (C) 0 % stimulus feature overlap among tasks within the subset. (D) Mean multitasking accuracy (averaged over pairs of tasks within a subset associated with different response dimensions) plotted against the mean number of iterations required to train the network to a fixed criterion on all single tasks (MSE=0.01). All data points represent the mean measures across networks initialized with the same task similarity for tasks in the same subset and same environment. (E) Enlarged view of 100% feature overlap condition showing that, unlike in the other conditions, initial bias toward sharing was positively correlated with faster learning and negatively correlated with multitasking accuracy.

Results. As might be expected, networks with a higher initial bias toward sharing (i.e., higher correlation of the task weights between pairs within a set) developed more similar representations at the hidden layer for those tasks (in terms of the final correlations between task weight vectors; Fig. 27A-C). Furthermore, as observed in Simulation Study 4, shared structure in the task environment influenced the correlation between learned task representations, with higher stimulus feature overlap between tasks within a set leading to higher correlations between the representations of those tasks. Critically, in environments with high feature overlap between tasks, stronger initial biases toward shared representation lead to increased learning speed (i.e. fewer iterations required to achieve a given level of single-task performance), as similarities between tasks could be exploited by means of shared representations (Fig. 27D-E). That is, biases toward shared representation amplified learning benefits from shared structure between tasks, suggesting a direct link between the presence of shared representation and learning efficacy. However, this came at the cost of multitasking performance. Networks that learned faster (due to biases toward shared representation) showed lower performance in multitasking, at least for environments with high amount of feature overlap (Fig. 27E). Not surprisingly, learning benefits from shared representations were less prevalent in environments with less feature overlap between tasks (in fact, there was a trend toward the opposite effect). Nevertheless, the effects of shared representation on impairments in multitasking remained (Fig. 27D). These results suggest that, to the extent it is advantageous for an agent to be able to respond to the same set of stimuli in more than one way (e.g., point to an object such as a ball or a rock, pick it up, or kick it) then an “inductive bias” (such as small, random initial weights) that favors the development of shared representations may be valuable insofar as it ensures faster learning of different responses to those objects (i.e., tasks), even though they will be dependent on control and risk multitasking interference if several of those objects must be processed in different ways at the same time. That is, systems that must function flexibly in rich environments may, at least by default, favor efficacy of learning over the efficiency of parallel processing. We address this tradeoff more directly in the section titled “A Normative Theory of Automaticity: Meta-control and the Tradeoff between Shared and Separated Representations”.

3.1.5 Simulation Study 7: Cognitive Flexibility and Transfer to Novel Tasks. In addition to more rapid learning, shared representations have been associated with improved transfer; that is, facilitated acquisition of novel tasks that share structure with those on which the network was already trained (Bengio et al., 2013; Caruana, 1997; Collobert & Weston, 2008; Zamir et al., 2018). The ability to flexibly acquire novel tasks is often attributed to cognitive control (Diamond, 2013; Goschke, 2000; Kriete et al., 2013; Shiffrin & Schneider, 1977; Verguts, 2017). That is, cognitive control is thought to support rapid learning of novel tasks, allowing organisms to flexibly adapt to changing demands. Some have suggested that the brain can achieve this flexibility by leveraging existing representations for novel tasks (Kriete et al., 2013; Verguts, 2017). For instance, Verguts (2017) suggests that the rapid acquisition of novel stimulus-response mapping tasks can be accomplished by synchronizing existing representations for stimuli and responses that are needed to perform the task. However, Vergut's study did not illuminate how flexible task learning depends on the presence of existing representations.

In machine learning, the learned representations of pre-trained tasks are found to improve the generalization performance on a primary, related task (Baxter, 1995; Bengio et al., 2013; Caruana, 1997; Collobert & Weston, 2008; Zamir et al., 2018). Similarly, prior learning of simple task-related information was shown to facilitate the transfer to novel, more complex tasks (Bengio, Louradour, Collobert, & Weston, 2009; Chang, Gupta, Levine, & Griffiths, 2018; Elman, 1993; Krueger & Dayan, 2009; Rohde & Plaut, 1999). Such transfer effects are often studied in the context of "multi-task learning" paradigms (Caruana, 1997), in which an agent is be pre-trained on a set of auxiliary tasks before it is trained on a primary (target) task.

While research in machine learning has primarily related the effects of pre-training to improvements in performance on a primary task, we adopt the multi-task learning paradigm to demonstrate that shared representations give rise to the computational benefits of cognitive control in terms of the ability to rapidly acquire novel tasks. We test this hypothesis in the non-linear networks used above by studying the learning performance of a set of target tasks as a function of the number of tasks that a network is pre-trained on. Specifically, we investigate whether learned representations for stimulus dimensions in the hidden layer of a network facilitate the learning of tasks that are associated with the same stimulus dimensions.

Network architecture and task environment. The network architecture and processing used in this simulation were the same as those reported in Simulation Study 6. However, features in each stimulus dimension were coded as one-hot vectors, as in Simulation Studies 1-3. In addition, the number of units in the input and output layers was adjusted to represent a task environment with three stimulus dimensions and six response dimensions, and with three features in each dimension. Thus, the stimulus input layer had nine units and the output layer had 18 units, so that the network could support a total of $3 * 6 = 18$ possible tasks. However, as described below, the network was trained initially on only a subset of those tasks, and then tested on how quickly it could acquire others.

Training and analysis. 80 instances of the network were implemented and divided equally into four groups, in which the networks were pre-trained either on no auxiliary tasks, or one, two or three auxiliary tasks (see Fig. 28A, auxiliary tasks are depicted as thin, dashed arrows). Networks in all groups were trained until they reached an MSE criterion of 0.001. Each of the auxiliary tasks was associated with different stimulus and response dimensions. After their initial training (in the groups that received pre-training), networks in all four groups were trained on the same set of three target tasks, each of which was (like the auxiliary tasks) associated with different stimulus and response dimensions. Critically, target tasks shared the same relevant stimulus dimensions as the pre-trained auxiliary tasks, whereas they were associated with a different set of response dimensions. The networks were trained on all target tasks until they reached an MSE criterion of 0.001. For each group of tasks, we assessed transfer performance: the number of training iterations required to reach criterion on all target tasks. In order to visualize the the similarity between the hidden representations of auxiliary tasks and target tasks, we used MDS to project the single task patterns for all nine tasks in the hidden layer on a 2-dimensional plane, such that the Euclidean distances between task representations were preserved (see Simulation Study 5, cf. Fig. 25).

Results. Fig. 28B shows the MDS projections of the hidden layer patterns of activity for the auxiliary tasks (shown as thin circles) and target tasks (shown as thick circles) from an example network in each group. In each example, the representations of the tasks cluster into three groups, one for each of the stimulus dimensions. Furthermore, for networks that were pre-trained on auxiliary tasks, the representations for the target task were close to those for the auxiliary task that shared the same stimulus dimension. This suggests that target tasks re-use the representations for the stimulus dimension that they share with a pre-trained auxiliary task. The average learning curve for each group is shown in Fig. 28C. The learning curves indicate that target tasks are acquired faster if the network is pre-trained on more auxiliary tasks. Without any pre-training, networks required on average 27.55 ($SD = 0.76$) training iterations. When pre-trained on one, two or three auxiliary tasks, networks learned all target tasks on average in 22.10 (SD = 1.37), 15.35 ($SD = 1.18$) and 11.10 ($SD = 0.55$) training iterations, respectively.

These results support the conjecture that shared representation do not just give rise to serial processing constraints, as explored in previous sections, but do also facilitate rapid acquisition of novel tasks, i.e. cognitive flexibility. This suggests that representation sharing enables tasks to be learned quickly at the expense of inducing structural and functional dependencies with other tasks, forcing novel tasks to be processed in serial. In the next section, we empirically test this prediction in a modified version of the Stroop paradigm.

3.1.6 Empirical Study: Learning, Shared Representations and Functional Dependence. The mathematical analysis and simulation studies above make clear the consequences of the tradeoff between shared and separated representations for learning efficacy vs. processing efficiency. These make three predictions with regard to human performance: (1) learning a new task involving a

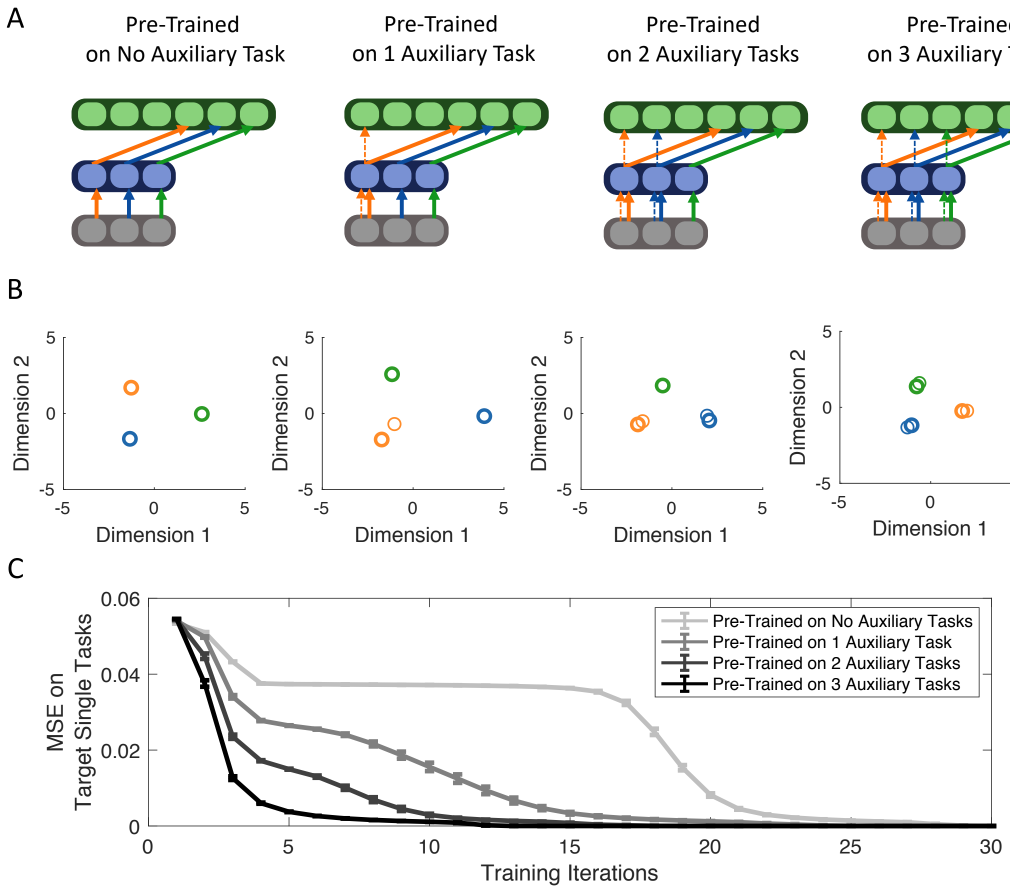

Figure 28. **Effects of pre-training on the acquisition of novel tasks.** (A) Pre-training conditions. Pre-training was performed in a network with three stimulus dimensions in the input layer (shown in grey) and six response dimensions in the output layer (shown in green). The hidden layer is shown in blue and depicts hypothesized learned representations of each stimulus dimension. Networks were pre-trained on no, one, two, or three auxiliary tasks (thin, dashed arrows) before they were trained on three target tasks (thick, solid arrows). (B) Projections of hidden representations for each task in a trained example network onto a 2-dimensional plane while maintaining Euclidean distances between the representations using MDS. Each plot in (B) corresponds to the pre-training condition shown above in (A). Projections of auxiliary tasks are shown as thin circles and projections of target tasks are shown as thick circles. Circles with the same color correspond to projections of tasks that share the same stimulus dimension. (C) Mean squared error on the target tasks as a function of training iterations for different pre-training conditions. Vertical bars represent standard errors of the mean across different networks.

stimulus dimension for which there are already representations (i.e., that is used by other familiar tasks) should be associated with rapid acquisition (by exploiting the shared use of those representations); (2) it should not initially be possible to perform that task simultaneously with others that rely on that input representation; however, (3) extensive practice on such multitasking should make it possible to perform them simultaneously. The idea that performance of a novel task may be control-dependent, but that extensive practice can lead to automaticity (and associated multitasking capability) has of course been demonstrated by a number of classic studies (e.g., Logan, 1988; MacLeod & Dunbar, 1988; Shiffrin & Schneider, 1977). However, neither those studies nor any others of which we are aware explicitly addressed the role of shared representations in mediating the observed effects. To do so, we conducted an empirical study using a modified version of the Stroop paradigm (cf. Fig. 5), and analyzed both overt performance (i.e., RT and accuracy) as well the extent to which multitasking performance reflected serial parallel vs. parallel processing, using the measures discussed above (Townsend & Wenger, 2004, see Simulation Study 2).

In the classical Stroop (1935) paradigm (described in Part I, in the section titled “A Simple Neural Network Model”), the canonical observation of poorer performance for color naming of incongruent stimuli (e.g., responding “red” to the word GREEN displayed in red) is widely considered to reflect response interference (Glaser & Glaser, 1982; Morton & Chambers, 1973; Roelofs, 2003) arising from shared phonological representations (see Fig. 2). This represents an instance of structural interference, as we defined it in Part I (see Section “Definitions”). This not only precludes multitasking but, as the Stroop effect demonstrates, can even degrade single task performance in the case that it involves one that is weaker than those with which it shares representations (as in the case of the color naming versus word reading; J. D. Cohen et al. (1990)). Here, we use an extended version of the Stroop task to address *functional* interference, and test the first and second predictions enumerated above; viz., that learning an new task preferentially relies, when possible on the use of existing representations, and that doing so can lead to functional interference that impairs multitasking performance.

The study involved three single task conditions and two dual-task conditions, all of which used the same Stroop stimuli. In all conditions, a trial consisted of presenting a Stroop stimulus (color word displayed in a congruent or incongruent color) at one of four eccentric locations on a computer screen.

Single task conditions. In the single task conditions, participants were asked either to say the color of the stimulus out loud (*color naming*, CN), to map the location of the stimulus to a key press (*location mapping*, LM), or to map the color word to a key press (*word mapping*, WM). Note that location mapping and word mapping are considered novel tasks in the sense that participants were required to learn arbitrary associations between color words and locations (as stimuli) and keys (as responses). Trials in which the ink color and the color word matched were considered to be *congruent* trials whereas trials in which they did not match were considered to be *incongruent* trials.

As discussed in Section "Graph-Theoretic Analyses" of Part I, there are at least two ways participants could learn to perform the word mapping task: They could exploit existing orthographic representations (i.e., those used for word reading), and learn to map these to manual responses (see Fig. 29A); alternatively, they could learn a new set of orthographic representations dedicated to mapping words to manual responses (see Fig. 29B). The former involves the sharing of existing representations (e.g., between word reading and word mapping) that is predicted to be relatively quick, but should lead to control-dependence of the word mapping task and, in particular, the inability to multitasking it with color naming; while the latter involves the development of new representations dedicated to the word mapping task, that are separate from those used for word reading, which should take longer but allow the word mapping task to be multitasked with color naming. The multitasking conditions of the experiment were designed to test predictions made by each of these possibilities.

Multitasking conditions. In the first multitasking condition, participants were asked to perform the color naming task concurrently with the word mapping task (CN+WM). If participants learned to perform the word mapping task using shared orthographic representations (Fig. 29A), then performance in this multitasking condition should be subject to considerable interference. This is because it would require the allocation of control to the hidden representations for words, which are

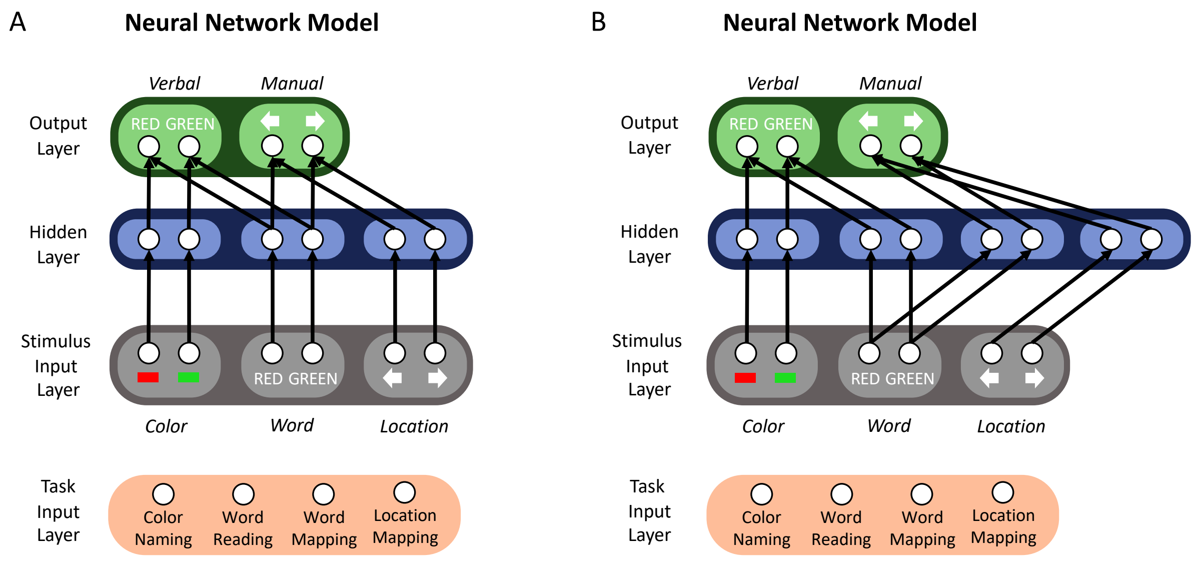

Figure 29. **Two Neural Network Models of the Extended Stroop Paradigm.** Each network implements simplified examples of the four tasks (using only two features for each stimulus and response dimension): color naming (CN), word reading (WR), word mapping (WM) from a word to a key press, and location mapping (LM) from a location to a key press. Both networks are capable of performing color naming and location mapping at the same time because both tasks are independent of one another (i.e., they do not share any representations). However, the two networks show different performance when asked to multitask color naming and word mapping. (A) In the first network, the word mapping task shares a representation for words with the word reading task at the hidden layer, introducing functional dependence between the word mapping task and the color naming task. As a consequence, the network is not able to accurately perform color naming and word mapping at the same time. (B) In the second network, the word mapping task has a separate representation for words. As a consequence, these are independent, and the network can perform color naming and word mapping simultaneously.

shared with the word reading process. This would implicitly engage the word reading process, which interferes with color naming, thus producing functional dependence between word mapping and color naming. Such functional dependence would induce greater interference for incongruent Stroop stimuli compared to congruent Stroop stimuli. In contrast, if participants learned a set of orthographic representations dedicated to the word mapping task (Fig. 29B), then this multitasking condition would not introduce functional dependence and any attendant interference, and therefore performance should be unaffected by congruency. Thus, the use of shared vs. separated representations for word reading vs. word mapping make different predictions regarding performance for multitasking color naming and word mapping, which can be used to adjudicate between the two possibilities. Based on the formal analyses above, we predicted that learning of the word mapping task in the second single task condition should favor the exploitation of shared representations (i.e, use of existing orthographic representations for word reading), which should not only produce an impairment of multitasking performance for color naming and word mapping but, critically make this sensitive to congruency.

In the second multitasking condition participants were asked to multitask color naming and location mapping (CN+LM). This served as a control for the effects predicted above. According to the network model depicted in Fig. 29, these tasks are fully independent, and thus it should be possible to perform them concurrently without interference, by allocating control to the hidden representations that map the two stimulus dimensions (color and location) to the response dimensions associated with each task (verbal and manual, respectively). For the same reasons, performance in this condition should be unaffected by stimulus congruency.

Below, we present additional details of the experimental procedure, simulations using the neural network model presented in Part I that formalize our predictions, and empirical data regarding human performance that test these predictions.

Experiment procedure. The experiment consisted of the three single task conditions and two multitask conditions described above. Participants first performed the three single task conditions (CN, WM, LM) in the fixed order described above, and then performed the two multitask conditions (CN+WM, CN+LM). The order of the multitask conditions was counter-balanced across participants.

In all conditions, a trial began with a grey screen and a fixation cross at its center for an inter-trial interval (ITI) of 500ms. After the fixation cross, a Stroop stimulus was presented for 850ms. Each Stroop stimulus consisted of one of four color words (“RED”, “GREEN”, “BLUE”, “BROWN”) displayed in one of four colors (red, green, blue, brown) at one of four locations (left, top, bottom, right). The color, word, and location of the stimulus was fully counterbalanced across conditions. Thus, each condition contained one block of 64 trials (reflecting a fully crossed 4 x 4 x 4 design involving the three factors (color, word and location) with four levels each. All single task conditions were performed before multitask conditions, and before each of the single task conditions participants performed five practice trials of the task for that condition.

In each condition, participants were instructed to indicate their response(s) while the stimulus was on the screen. In the CN condition, participants responded to the color of the stimulus with their voice, by naming the color out loud. In the LM condition, participants were instructed to respond to the left, top, bottom and right position of the stimulus with the keys “1”, “2”, “3” and “4” respectively. In the WM condition, participants were asked to respond with the same set of keys to the four color words, with specific assignments counterbalanced across participants. In each of the single task conditions, participants were instructed to ignore the two task-irrelevant stimulus dimensions (e.g. during the CN condition, participants were told to ignore the word and location of the stimulus). In the two multitask conditions, participants were instructed to respond to the two task-relevant stimulus dimensions simultaneously, using the same response mappings as in the single task conditions, while ignoring the third stimulus dimension irrelevant to both tasks. Thus, in the CN+WM condition, participants were instructed to name the color of the stimulus *at the same time* as pressing the key corresponding to the word learned for the WM condition, while ignoring the location of the stimulus; whereas in the CN+LM condition, they were instructed to name the color in which the stimulus was displayed while simultaneously pressing the key corresponding to the location of the stimulus relative to the center dot, while ignoring the word.

Sample. Thirty individuals were initially enrolled to participate, but three were disqualified based on technical malfunctions or misunderstanding of instructions. We excluded another 6 participants whose accuracy was below chance (25%) in at least one of the single task conditions, yielding 21 participants (14 female) ages 18 to 34 years (M = 21.52 years) who were included in data analysis. All participants gave written informed consent and were debriefed about the purpose of the study after the experiment. The study was approved by the Institutional Review Board of Princeton University.

Data analysis. The response time (RT) and accuracy for each task in each trial was recorded. Reaction times for verbal responses were determined by plotting the waveform for the audio response for each trial and having graders manually select the time of speech onset. Manual grading was necessary to ensure that random acoustic signals, such as coughing or deep breaths, were not counted as speech onset. The graders were blind as to which trials came from which conditions. Mean RT and accuracy was computed separately for congruent and incongruent trials in each single task condition for each participant. For the multitask conditions, we computed accuracy by considering a trial to be correct if the response for both tasks was correct. The RT of a multitasking condition corresponded to the slower of the two responses and was conditioned on correct trials only. As with the single task conditions, multitasking accuracy and RTs were computed separately for congruent and incongruent trials.

We first conducted one-tailed t-tests for each multitasking condition to determine whether accuracy was above chance level for each condition. In order to investigate the effects of multitasking condition (CN+WM vs. CN+LM) and stimulus congruency (color-word congruent/incongruent), we used two linear mixed effects regression models: (1) a generalized linear mixed effect regression of multitasking accuracy, assuming binomial distribution of response variables with a logit link function and (2) a mixed effect linear regression of multitasking RT. In the first model, accuracy (as defined above) was the dependent measure, with fixed effects estimated for multitasking condition, stimulus congruency, and the interaction between multitasking condition and congruency. In the second model, RTs were used as the dependent measure, with the same fixed effects as the first model. Both models also included a random effect of participant to account for individual differences.

Previous work has shown that accuracy and RT measures are insufficient indicators of parallel versus serial processing (Townsend, 1972, 1990). Moreover, accuracy or RT differences between multitasking conditions may be the result of performance differences in the single tasks. Thus, it is difficult to infer whether participants operated more or less parallel in one multitasking condition versus the other when investigating multitasking accuracy and RT alone. To overcome for these limitations, we computed a metric of parallel processing capacity proposed by Townsend and Wenger (2004) for both multitasking conditions. In their work, Townsend Wenger introduce a *load capacity coefficient* C(t) that assesses the degree two which two task processes operate in parallel at time point t, by assessing the distribution of RTs for each individual task, and comparing it to the distribution of RTs at which participants respond to multiple tasks simultaneously. The capacity coefficients can be used to assess the degree of interaction between two tasks, taking into account performance for each single task.

For each participant, the capacity coefficient in the CN+WM condition was defined as

$$C_{CN+WM}(t) = \frac{log(P(T_{CN} \leq t)) + log(P(T_{WM} \leq t))}{log(P(T_{CN} \leq t \text{ AND } T_{WM} \leq t))} \tag{9}$$

where $P(T_{CN} \leq t)$ corresponds to the probability that the participant responded to the color naming task before time t in the CN condition, $P(T_{WM} \leq t)$ corresponds to the probability that the participant responded to the word mapping task before time t in the WM condition, and $P(T_{CN} \leq t \text{ AND } T_{WM} \leq t)$ corresponds to the probability that the participant responded to both tasks before time t in the CN+WM condition. The capacity coefficient for the CN+LM condition, $C_{CN+LM}(t)$, was defined an analogous manner. We computed the capacity coefficients in both multitasking conditions across all stimuli and separately for each participant. Similar to Townsend Wenger (2004), we conditioned these measures on correct trials.[37] A capacity coefficient of 1 would indicate that the two tasks were executed in parallel at time point t, suggesting that the underlying task processes are independent. A capacity coefficient larger than 1 would

[37] Townsend and Altieri (2012) propose similar metrics taking into account multitasking accuracy. However, our experiment did not yield sufficient numbers of trials for both correct and incorrect responses to compute those metrics.

indicate that the two task processes facilitate each other when executed in parallel (yielding faster RTs for both tasks compared to when each task is executed alone). Conversely, a capacity coefficient smaller than 1 would indicate that the two task processes interfere with one another. We predicted that $C_{CN+WM}(t) < C_{CN+LM}(t)$ at any time t if the color naming and word mapping task are functionally dependent by means of a shared representation between word reading and word mapping.

Neural network simulation. We simulated the experiment using the same general neural network architecture and learning parameters as described in Simulation Study 2.[38] The stimulus input layer was comprised of three stimulus dimensions (representing color, word and location) with four input units per dimension. The output layer was compromised of two response dimensions (verbal and manual), with four output units per dimension. The task input layer was comprised of four task units, one each for the color naming, word reading, word mapping and location mapping tasks.

We trained 21 networks on each of the four individual tasks using the entire set of Stroop stimuli used in the experiment. As in Simulation Study 2, we sampled 100 patterns for each of the single tasks (CN, WM and LM) per epoch. We also trained the network on twice as many patterns for the word reading task to simulate prior training on WR (cf. J. D. Cohen et al., 1990). The network was trained until it reached an average MSE of 0.001 over all three relevant single tasks.

After training, we used the procedure described in Simulation Study 1 to extract a task dependency graph based on the single task representations in the network. To assess the similarity between the learned representations for each task in the hidden layer of the network, we projected each task representation onto a 2-dimensional plane as described in Simulation Study 5. We also computed the average accuracy across all networks for all single tasks (CN, WM and LM), as well as for both multitasking conditions (CN+WM, CN+LM), separately for congruent and incongruent stimuli.

[38] Note that the network was not fit to experiment data. Instead, we used the same parameters as in previous simulations to derive qualitative predictions about the network's performance in the extended Stroop paradigm.

Finally, we investigated the effects of multitasking condition (CN+WM vs. CN+LM) and stimulus congruency (color/word congruent/incongruent) in a mixed effect linear regression. We modeled multitasking accuracy as a function of multitasking condition, stimulus congruency, as well as their interaction. Differently initialized networks were treated as a random effect.

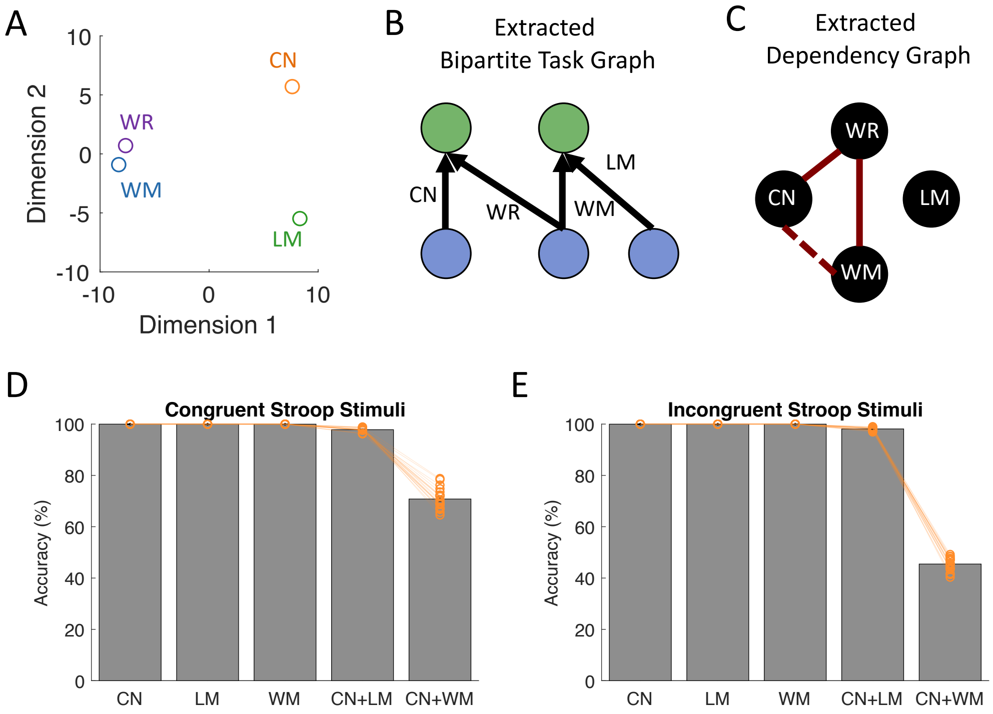

Figure 30. **Neural network simulation of modified Stroop paradigm.** (A) Hidden unit representations in an example of a trained network for color naming (CN), word reading (WR), word mapping (WM) and location mapping (LM) projected onto a 2-dimensional plane while maintaining Euclidean distances between the representations using MDS. Each circle corresponds to a projection for a given single task (see Fig. 25 for additional details). (B) The bipartite task graph extracted from representations in the hidden and output layers of the example network. (C) The corresponding task dependency graph, with structural dependencies shown as solid lines and functional dependencies as dashed lines. (D, E) Accuracies for single tasks and multitasking conditions after network training for (D) congruent and (E) incongruent Stroop stimuli, averaged across all networks. Each dot corresponds to performance of a single network in a given condition.

Results: neural network simulation. Fig. 30A shows projections of the patterns of activity in the hidden layer for the four single tasks after training in an example network. The representations for word reading and word mapping form a cluster, suggesting that the neural network exploits structural similarity between the two tasks by learning a shared representation. As a consequence, both tasks share an input node in the extracted bipartite task graph (Fig. 30B). Thus, the corresponding task dependency graph predicts functional dependence between the color naming and word mapping tasks (Fig. 30B-C). However, neither structural nor functional dependence is predicted between the color naming and location mapping tasks. The performance overall all networks was consistent with this prediction: they were more accurate in multitasking color naming and location mapping (CN+LM) than color naming and word mapping (CN+WM), ($\beta = -0.2701,\ SEM = 0.0070,\ p < 10^{-52}$). Notably, multitasking performance in the CN+LM condition was comparable to the high overall performance on all single tasks, and stimulus congruence showed no main effect on multitasking accuracy ($\beta = 0.0030,\ SEM = 0.0070,\ p = 0.6683$). However, the mixed effect regression revealed a significant interaction between multitasking condition and stimulus congruency ($\beta = -0.2564,\ SEM = 0.0100,\ p < 10^{-40}$), suggesting that incongruent stimuli had a detrimental effect on accuracy when multitasking CN+WM but not when multitasking CN+LM (Fig. 30D-E).

Results: human performance. Table 1 lists accuracies and RTs for all experiment conditions. Performance dropped for multitasking CN+LM, but participants' error rate was still above chance (multitasking chance performance = 6.25%) for congruent trials ($M = 76.02\%,\ SD = 33.83\%$), $t(20) = 9.4520,\ p < .0001$, and incongruent trials ($M = 71.54\%,\ SD = 28.67\%$), $t(20) = 10.4357,\ p < .0001$. Note that human performance in the CN+LM condition was notably lower compared to the neural networks' performance in this condition. This suggests that there may be factors over and above functional dependence that contributed to impaired multitasking performance (see Summary and Conclusions for Part II). However, as predicted by the simulation results, performance for CN+WM was much lower, despite the fact that participants could perform each of these tasks on their own extremely well (see Fig. 31A-B). The error rate for congruent CN+WM trials was still above chance ($M = 28.03\%,\ SD = 33.83\%$), $t(20) = 3.7092,\ p < .001$. For incongruent CN+WM trials, where the color and word were in conflict, accuracy was also above chance ($M = 11.47\%$, $SD = 28.67\%$), $t(20) = 2.2564$, $p < .05$.

Condition	Accuracy in % (M ± SD)		RT in s (M ± SD)	
	Congruent	Incongruent	Congruent	Inongruent
Single Tasking				
CN	100.00 ± 0.00	96.49 ± 4.56	0.641 ± 0.086	0.696 ± 0.074
LM	96.63 ± 7.41	97.10 ± 4.75	0.498 ± 0.088	0.502 ± 0.083
WM	88.35 ± 16.57	89.68 ± 8.26	0.720 ± 0.101	0.775 ± 0.088
Multitasking				
CN+LM	85.71 ± 35.86	80.95 ± 40.24	0.971 ± 0.087	0.991 ± 0.074
CN+WM	33.33 ± 48.30	9.52 ± 30.08	0.883 ± 0.151	0.964 ± 0.124

Table 1

Accuracies and RTs for extended Stroop task. *M and SD correspond to the mean and standard deviation across participants, respectively. Results are reported for single task conditions color naming (CN), location mapping (LM), word mapping (CM) and multitasking conditions color naming + location mapping (CN+LM), as well as color naming + word mapping (CN+WM).*

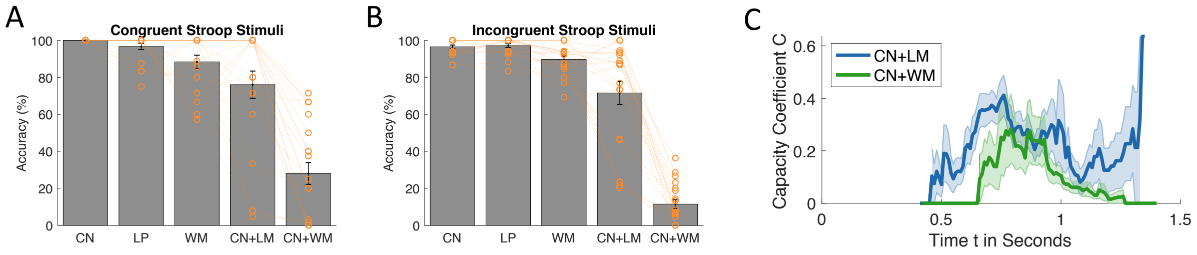

Figure 31. **Behavioral results for human participants in modified Stroop paradigm.** (A, B) Accuracies for single tasks (color naming, CN; location mapping, LM; word mapping WM) and multitasking conditions for (A) congruent and (B) incongruent Stroop stimuli averaged across all participants. Each dot corresponds to performance of a single participant in a given condition. (C) Capacity coefficient for both multitasking conditions as a function of time (see text) averaged across all participants (solid lines). Shaded area around each line indicates standard error of the mean across participants.

The linear mixed effects regression models further illustrate the differences between multitasking condition and stimulus congruency; accuracy was significantly lower on CN+WM trials compared to CN+LM trials ($\beta = -1.9630,\ SEM = 0.2813,\ p < .0001$), and RTs significantly slower ($\beta = 0.1834,\ SEM = 0.0317,\ p < .0001$). As expected, RTs were overall slower on incongruent compared to congruent trials ($\beta = 0.0641,\ SEM = 0.0202,\ p < 0.01$). However, accuracy was overall higher on incongruent compared to congruent trials ($\beta = 0.5660,\ SEM = 0.2076,\ p < .01$). A post-hoc analysis revealed that incongruent trials were associated with higher accuracy than congruent trials on in the CN+LM condition ($\beta = 0.5172,\ SEM = 0.2175,\ p < 0.05$); as predicted by functional dependence of CN and WM, participants performed worse on incongruent trials relative to congruent trials in the CN+WM condition ($\beta = -0.9072,\ SEM = 0.2672,\ p < .001$). As a consequence, accuracies yielded a significant interaction between multitasking condition and congruency ($\beta = -1.7031,\ SEM = 0.3421,\ p < .0001$), while there was no significant interaction between multitasking condition and congruency for RTs ($\beta = 0.0650,\ SEM = 0.0421,\ p = 0.1237$).

Fig. 31C shows the capacity coefficient for both multitasking conditions as a function of time within trial. The capacity coefficient stayed below 1 across all participants for both multitasking conditions, suggesting that in both multitasking conditions overall the two tasks interfered with one another. For short response times (< 0.74s), the capacity coefficient was significantly lower in the CN+WM condition compared to the CN+LM condition, suggesting a greater degree of interference at early stages of processing (note that the capacity coefficient ensures a fair comparison by taking into account the RT of each single task). For longer response times, the two multitask conditions were comparable in terms of their capacity coefficient.

Overall, these results indicate that human participants performed poorly in the CN+WM condition relative to the CN+LM condition, as predicted by the network model. This supports the conjecture that participants leveraged existing representations (e.g. for WR) when acquiring a novel task (WM), leading to functional interference between CN and WM. This is further supported by the observation that performance decrements in multitasking CN+WM are greater for both the network model and human participants when stimuli were incongruent as opposed to congruent. One could argue that participants shouldn't exhibit this behavior if they had learned separated instead of shared representations for WM and WR. This begs the question: Why would humans prefer learning shared over separated representations, at the cost of limitations in multitasking capability? In the next section, we review a normative theory of the tradeoff between shared and separated representations to explore this question.

3.1.7 A Normative Theory of Automaticity: Meta-control and the Tradeoff between Shared and Separated Representations. The tradeoff between shared versus separated representations, and its consequences for learning efficacy versus processing efficiency, raises a higher level question about strategic decision making, and the allocation of cognitive control. This involves an intertemporal choice between the more immediate value of acquiring a skill quickly using shared representations, but at the expense of control-dependence and the inefficiency of serial processing (e.g., playing the piano with one finger at a time), vs. the potentially greater value of processing efficiency afforded by separated representations, but that is deferred due to the additional time (as well as effort, and possibly even expense) required to acquire task dedicated representations (e.g., playing chords with several fingers at the same time).[39] This can be framed as an optimization, or bounded rationality problem (Gigerenzer, 2008; Simon, 1957; Todd & Gigerenzer, 2012), along the lines of recently proposed theories of cognitive control (Shenhav et al., 2013, 2017) by taking into account both the rewards and costs associated with each option. Here we present work that pursues such an approach as applied to the choice between learning shared versus separated representations.

3.1.8 Model Structure. Sagiv, Musslick, Niv, and Cohen (2018) constructed an abstract model of the tradeoff between shared and separated representations. This modeled an agent that learns to perform a set of tasks involving a common set of

[39] This is consistent with the proposition that intertemporal choice is a fundamental feature of all decisions about the allocation of control (J. D. Cohen, 2017).

stimulus and response dimensions (along the lines of those described in Fig. 3), by progressively sampling and modifying through learning the efficacy of one or both of two ways of performing the task: one with the properties of a minimal basis set representation, that was learned more quickly but imposed a serialization cost that scaled with the number of tasks that had to be performed in a given trial; and another with the properties of a full tensor product representation, that took longer to learn but permitted full multitasking (i.e., concurrent performance) of any number of tasks thus averting any serialization costs. On each trial, the agent was required to perform some number of tasks (ranging from 1-5), and could choose which of the two ways to perform them. The agent was rewarded for the performance of each task independently based on the accuracy for that task, and the goal of the agent was to optimize overall future-discounted reward rate (discussed further below). Critically, if the method corresponding to a minimal basis set was used, each task had to be performed one at a time, whereas for the one corresponding to the tensor product all tasks were performed at once. Thus, for equal levels of performance, using the former yielded a reward rate that was a fraction of the other (that is, it scaled inversely with the number of tasks performed in a given trial).

Both methods were initialized to generate poor performance, but each time one was chosen (once per trial), performance using that form improved according to a pre-specified (logistic-shaped) learning curve. Thus, initially improvement was slow, but then accelerated, and eventually asymptoted at maximal performance. While the slope of the two learning functions was the same, the offset for the minimal basis method was less than for the tensor product method, thus implementing faster learning for the former compared to the latter. The agent was initialized with high uncertainty around the true values of the offset and slope (learning rate) of the learning curves, but then observed and learned about the rate at which each method improved with use — by observing the outcome of performance and using this to improve its estimate of the offset and slope parameters for each learning function. The agent used Bayesian inference to estimate both the learning rate for each learning function as well as the probability of receiving task-sets of different sizes (i.e., opportunities for multitasking) in the environment. These estimates were then used to compute, on each trial, the expected discounted future value of each method, and choose which to execute for that trial. Thus, the model implemented a normative solution to the question of whether to more quickly achieve acceptable levels of performance using the method corresponding to a minimal basis set representation, at the expense of the serial costs and lower achievable reward rate when more than one task had to be performed; or to invest in learning the method corresponding to a tensor product representation, that took longer but yielded a higher asymptotic reward rate once it was learned.

3.1.9 Results. The model was simulated for a range of serialization costs associated with the minimal basis set method, differences in learning rates between the two methods, and the discount factor used to estimate the cumulative future reward associated with each.[40] The results exhibited robust and clearly separated ranges of parameters that favored each method. This numerical result was complemented by a closed form analysis of a simplified version of the task, that characterized the conditions under which it was optimal to choose one method versus the other. That is, it defined the serialization cost and frequency of multitasking opportunities below which it was optimal to learn the minimal basis method, and above which it was optimal to learn the tensor product method. This analysis indicated that the minimal basis set method is more advantageous if: (1) shared representations lead to high improvements in learning speed; (2) there is a low cost associated with executing tasks sequentially; and (3) the agent's time horizon is finite (i.e. it has a limited amount of time to learn and perform the tasks). In general, the results suggest that it is optimal for the agent to choose the minimal basis set method over the tensor product method for a wide range of parameters. This provides the outline of a normative account for why, under many conditions, it is advantageous to favor control-dependent processing during initial acquisition of a task, and only invest the effort required for automatization under

[40] Alternatively, this can be thought of as different durations over which the agent expected to be performing these tasks (i.e., horizons over which future discounted value was computed).

conditions in which it is evident that this is worthwhile.

3.2 Summary and Discussion of Part II

In Part II of this article, we addressed the question of why a neural system would favor shared over separated task representations, given the reliance on control and constraints that this imposes on processing efficiency — that is, the multitasking capability of a network. We framed this in terms of a tension between shared versus separated representations, according to which the former affords more effective learning (and better transfer), while the latter affords greater efficiency of processing (i.e., multitasking capability). In the first two simulation studies of Part II, we showed that neural networks are likely to develop shared representations between tasks if they rely on similar stimulus features, and if the networks are trained to execute one task at a time. Conversely, training the networks on unrelated tasks, or training the network to perform multiple tasks at the same time lead to the acquisition of separated, task-dedicated representations.

We then investigated the computational tradeoff between these types of representations. We began with a formal analysis of linear networks, that revealed a fundamental dilemma faced by neural network architectures: Increasing the number of shared representations between similar tasks increases the speed with which the network can learn those tasks, but decreases the number of tasks that the network can ultimately perform at the same time without interference. We then showed that this tradeoff also applies to non-linear networks, by using weight initialization to bias such networks towards more or less shared representations. Furthermore, we showed that a bias toward shared representation arises “naturally” when a network is trained on multiple tasks that have shared input structure, and that such shared representations promote cognitive flexibility by facilitating transfer to novel tasks. Predictions about human performance made by these simulations were confirmed in a behavioral study using an extended version of the Stroop task, consistent with the hypothesis that human participants rely on shared representation of prior tasks (e.g. word reading) when learning a new task (e.g. word mapping) at the expense of multitasking performance (e.g. naming the color while pointing according to a word). Finally, we described a normative treatment of the tradeoff between shared and separated representations, showing that shared representations — and attendant limitations in multitasking — may be an optimal choice under a wide range of circumstances, providing an explanation for why performance of novel tasks often relies on control-dependent processing, and providing a formal framework for examining conditions under which the choice may be made to pursue automaticity.

Here we consider how the framework we have described relates to treatments of multiple resource theory and, more generally, how it relates to mechanisms described for learning and representation in the domain of semantic cognition, as well as machine learning.

3.2.1 Shared Representations and Multiple Resource Theory. One of the major criticisms of the original multiple resource theory (Allport et al., 1972; Navon & Gopher, 1979; Wickens, 1991), and more recent, computational implementations of it (Meyer & Kieras, 1997b; Salvucci & Taatgen, 2008), concerns the lack of specificity with regard to its core assumption; that is, which resources are shared between two tasks, and the extent to which they are shared. This explanatory gap allows arbitrary sets of resources to be proposed to account for any particular set of data (Hirst & Kalmar, 1987; Meyer & Kieras, 1997b). To address this explanatory gap, we studied the circumstances under which shared task representations emerge in neural network architectures.

Our simulation results suggest that statistical regularities between task-relevant stimulus features may help rationalize and constrain future instances of multiple resource theory. Simulation Studies 4 and 5 demonstrated that a subset of stimulus features that is statistically independent of other stimulus features is likely dedicated a separate resource (representation) in a neural network whereas statistically correlated features may be dedicated a common resource (representation) (see Lesnick et al. (2020) for a more formal definition of a stimulus and response dimension). That is, neural networks develop variable amounts of shared representation as a function of structural similarity between tasks: The more task-relevant stimulus features are shared between two tasks, the more a neural network is likely to learn shared representations for those two tasks. This observation reflects a fundamental and well-recognized characteristic of neural network architectures and learning algorithms: that they encode similarity structure of the environment and exploit this in learning in a graded manner, as functions of both degree of similarity and training (Hinton et al., 1986; A. M. Saxe et al., 2019; Rumelhart et al., 1993). This characteristic provides a rationale, and a quantitative grounding for the core assumption of multiple resource theory: In addition to perceptual similarity, at a finer scale, if the structure of information *within* a modality is shared across tasks, then those tasks will like rely on shared representations of that structure. Conversely, Simulation Study 4 showed that a neural system may learn different representations for tasks, even if they rely on the same perceptual modality, if the stimulus features on which they rely are uncorrelated. For instance, colors and words are both visual inputs but may be regarded as separate stimulus dimensions if they are statistically unrelated. Results from Simulation Study 4 are in line with findings of P. Lindsay, Taylor, and Forbes (1968), showing that even if two tasks rely on the same sensory modality (e.g. for visual inputs), they may not interfere with one another if they rely on representations for different sets of task-relevant features[41].

Results from Stimulation Study 4 are also in line with insights gained from the study of semantic knowledge acquisition, showing that neural networks develop shared representations for stimuli that share similar semantic features (Hinton et al., 1986; McClelland et al., 1995; Quinn & Johnson, 1997; Rumelhart et al., 1993). This has received empirical support from fMRI studies, which suggest that stimuli with similar semantic features overlap in terms of their neural patterns of activity, both within and across individuals (Kriegeskorte & Kievit, 2013; Carlson, Simmons, Kriegeskorte, & Slevc, 2014; Connolly, Gobbini, & Haxby, 2012). A recent mathematical analysis of

[41] Note that a lack of interference requires the two tasks are also functionally, and not just structurally independent (see Section “Graph-Theoretic Analyses”).

semantic development by A. M. Saxe et al. (2019) suggests that the learning of shared representation based on statistical similarities reflects the outcome of an optimal learning process.[42] Thus, the same principle — that learning of shared representation between tasks reflects an optimization process in learning statistical regularities over a set of inputs — seems to apply across cognitive domains, from simple sensorimotor tasks to more complex domains such as language. In the General Discussion, we relate these ideas more generally to the study of semantic cognition and category formation.

3.2.2 Multitasking Practice and Representational Separation. Despite constraints on multitasking, a number of studies have suggested that the ability to execute two or more tasks simultaneously can improve with extensive practice (Hazeltine et al., 2002; Liepelt et al., 2011; Ruthruff et al., 2006; Schumacher et al., 2001). While some have suggested that these improvements can result from practice on performing each single task alone (Ruthruff et al., 2006), others have argued that larger improvements can be achieved through multitasking training (Liepelt et al., 2011). Simulation Study 5 is consistent with the latter observation, showing that repeated simultaneous execution of multiple tasks can lead to greater improvements in multitasking performance compared to single task training.

The benefit of dual-task training over single task training has lead some researchers to suggest that dual-task training improves inter-task coordination that can generalize to other dual-task conditions (Bier, de Boysson, & Belleville, 2014; Hirst, Spelke, Reaves, Caharack, & Neisser, 1980; Kramer, Larish, & Strayer, 1995; Liepelt et al., 2011; Strobach, Frensch, & Schubert, 2012). While this may be true, Simulation Study 5 suggests an alternative possibility: that dual-task practice promotes the acquisition of separated, task-dedicated representations in order to minimize processing conflict — a training signal that is generally absent in single task practice.[43] The results

[42] A. M. Saxe et al. (2019) define optimal learning as identifying the smallest norm weights in a linear neural network to solve a given task.

[43] Note that, in our simulations, we observed a small amount of representational separation even with single task training. This is consistent with the observation that single task training alone can improve dual-task performance (Liepelt et al., 2011; Strobach et al., 2012; Ruthruff et al., 2006)

of Simulation 6 further suggest that representational separation between tasks may be sufficient to improve dual-tasking performance, and does not require improvements in inter-task coordination. Critically, representational separation would predict no positive transfer of practice from one dual-task condition to other dual-task conditions because representational separation would only apply to the tasks being practiced. This is consistent with the results of empirical studies that have found little or no such transfer effects (Strobach et al., 2012; Liepelt et al., 2011).

3.2.3 Neural Mechanisms Underlying Multitasking Performance.

While most behavioral studies of dual-task training suggest that performance can improve with sufficient practice, they have not addressed the neural mechanisms that underlie such improvements. Neuroimaging studies have suggested at least three plausible candidate mechanisms: (1) improved efficiency of existing brain regions (*efficiency account*; Dux et al., 2009; Jonides, 2004; Kelly & Garavan, 2005; Poldrack, 2000), (2) a reduced recruitment of brain regions associated with cognitive control with concomitant redistribution of task processes to other areas (*redistribution account*; Chein & Schneider, 2012; Dux et al., 2009; Kelly & Garavan, 2005; Petersen, Van Mier, Fiez, & Raichle, 1998) and (3) the segregation of neural representations between tasks within a task-specific brain region (*divergence account*; Garner & Dux, 2015). The efficiency account suggests that multitasking improvements can be attributed to more efficient processing of individual tasks, e.g. by a strengthening of synapses or formation of new synapses in underlying brain regions responsible for a single task (Münte, Altenmüller, & Jäncke, 2002; Rioult-Pedotti, Friedman, & Donoghue, 2000; Schlaug, 2001). This account is consistent with the proposition that multitasking improvements can be accomplished by reducing temporal overlap between tasks in the presence of processing bottlenecks, e.g. by compiling task processes into smaller chunks (Newell & Rosenbloom, 1981; Rosenbloom, Laird, & Newell, 1993; Taatgen & Anderson, 2002, see Section “A Mechanistic Account of Control-Dependent Versus Automatic Processing Based on Shared Versus Separated Representations” in the General Discussion). The redistribution account is based on the assumption that multitasking limitations arise from the reliance on capacity-limited mechanisms in brain regions associated with cognitive control, such as the prefrontal cortex. A number of fMRI studies have observed that task practice leads to a decreased activity of prefrontal regions in conjunction with increased activity in other brain areas during multitasking (Debaere, Wenderoth, Sunaert, Van Hecke, & Swinnen, 2004; Sakai et al., 1998; Shadmehr & Holcomb, 1997). Thus, the redistribution account postulates that improvements in multitasking through training are accomplished by re-routing of task processes away from regions presumed to implement capacity-limited control mechanisms to task-specific sensory-motor pathways (Dux et al., 2009). Finally, the divergence account suggests that multitasking training leads to a separation of task-representations, thereby reducing interference between them. Garner and Dux (2015) showed that if participants are explicitly trained to multitask, they are able to do so by developing separated task representations. Improvements in multitasking were highest for participants whose task representations were most separated after multitasking training.

The results of Simulation Study 5 are most consistent with the divergence account, suggesting that improvements in multitasking training can be achieved through a separation of task representations. Those simulation results suggest that single task training alone can lead to some representational separation between tasks, but that effect is modest (see Footnote 43). Representational separation is substantially greater if: (1) a network is trained to execute multiple tasks simultaneously; and (2) executing multiple tasks simultaneously leads to response conflict (i.e., the tasks are trained on incongruent as opposed to congruent stimuli). Note that Garner and Dux (2015) found that the relationship between representational separation and multitasking improvement was specific to frontoparietal and subcortical brain regions, suggesting that multitasking limitations can be attributed to shared representation between tasks in those regions. However, other studies have found that the relationship between representational separation and multitasking performance may not be specific to any particular region (Nijboer, Borst, van Rijn, & Taatgen, 2014). The present work suggests that representational separation may be greatest in regions that encode task-relevant associations between stimulus and response dimensions, rather than regions that just exert control over those[44].

Simulation Study 5 also provides a mechanistic basis for the findings offered in support of the redistribution account; viz., that training on multitasking leads to diminished engagement of control-related areas (e.g., Dux et al., 2009). While this is interpreted as evidence that multitasking training reduces reliance on control, it does not say how or why this comes about. Simulation Study 5 provides such an explanation. As illustrated in Fig. 3 in Part I, the minimal basis set representation (with overlapping task processing pathways) requires two control units per task — one to specify the relevant stimulus dimension and one to specify the relevant response dimension — whereas the tensor product representation (with separated task processing pathways) requires only a single control unit per task. Thus, the separation of task representations through multitasking reduces the representational requirements for control. Note that this inverts the traditional interpretation that a lesser engagement of control regions reflects the need to circumvent capacity limitations associated with the control system (Dux et al., 2009). However, according to the framework presented here, lesser engagement of those regions may reflect a reduced requirement for control due to the separation of representations between tasks in other regions. That is, in terms of the analogy of firemen as the control system and fire as processing conflict induced by shared representations, the absence of firemen as a need to avoid them, the absence of firemen may simply be interpreted as the absence of the fire itself.

3.2.4 Learning Efficacy Versus Multitasking Capability. Our analyses provide an indication of why agents may favor shared over separated representations, at least initially during training: shared representations afford faster learning if the tasks involved have similar structure (by way of more frequent weight updating for representations that are shared), or if a new task must be learned that can profitably exploit existing representations (i.e., by sharing with those existing ones). The

[44] See General Discussion for how the costs and benefits of shared versus separated representations may apply to regions relevant for task control.

mathematical analysis of linear models also predicts that the efficacy of learning gained by shared representation is dependent on the task environment: The higher the structural similarity between tasks, the larger the improvements in learning speed due to shared representation. Thus, the tradeoff between learning efficacy and multitasking capability should only be present in environments with shared structure between tasks. The results from Simulation Study 6 indicated that both of these effects also apply to non-linear networks. They also showed that biases toward shared representation only benefit learning if tasks are related to one another. However, regardless of the task environment, biases toward shared representation between tasks yield lower multitasking performance. Altogether, the mathematical and computational results provide both a quantitative and normative foundation for multiple resource theory.

One may argue that the benefits to learning of shared representation observed in Simulation Study 6 are relatively small, and thus insufficient to outweigh the detrimental effects for multitasking performance. However, a recent computational analysis of this tradeoff in deep neural networks revealed even larger effects of shared representation on learning speed in such networks(Ravi, Musslick, Hamin, Willke, & Cohen, 2020). In that study, a multilayered neural network was trained to perform various visual recognition tasks in a virtual environment. The network was provided with two stimulus dimensions: an input providing coordinates that designate the location of the object in a 3D image space, and a 2D image resembling the object. The network was trained to perform four tasks: (1) map 3D coordinates provided as input to a location in the 2D space of an image (coordinates → location)(2) label the object at a specified location in 3D space (coordinates → label); (3) identify the location of the object in the 2D image (image → location); and (4) label the object in the 2D image (image → label). Forcing the network to share representations between tasks with the same stimulus dimension (e.g. Tasks 3 and 4 performed on the image) lead to large benefits in learning speed. However, as predicted by the mathematical and computational studies presented above, this resulted in poor multitasking performance. Furthermore, learning benefits were larger for the two tasks relying on a more complex stimulus dimension (e.g., tasks relying on a visual image versus simple location coordinates). In addition, the benefits of shared representations for learning increased even further when the difficulty of learning both tasks was increased (by adding white noise to the inputs). Together, these observations suggest that more complex task environments impose a higher pressure for neural agents to rely on shared representations at the expense of multitasking capacity. In the next section we discuss the role of these benefits for cognitive flexibility, i.e. rapid transfer to novel tasks.

3.2.5 Cognitive Flexibility, Control, and Multitasking Capability. The term "cognitive flexibility" includes the human ability to rapidly acquire novel tasks (Kriete et al., 2013; Shiffrin & Schneider, 1977; Verguts, 2017).[45] As discussed in Simulation Study 7, a growing literature in machine learning suggests that agents can learn novel tasks more quickly if they leverage existing representations for pre-trained tasks. In that field, "multi-task learning" refers to the ability of an agent to learn multiple different tasks from experience with only a limited subset of those tasks during training (Bengio et al., 2013; Caruana, 1997; Collobert & Weston, 2008; Zamir et al., 2018). The development of a common representation through the acquisition of auxiliary tasks can be understood as an inductive bias (Caruana, 1997) that causes the neural network to prefer some representations over others when learning a novel, but related task. Building on a decision-theoretic framework for neural networks (Haussler, 1992), Baxter (1995) showed that the number of samples required to achieve good generalization performance for a task decreases with the number of related tasks on which a network is trained. In practice, training a network on multiple, related tasks has been shown to significantly improve learning in computer vision (Girshick, 2015; Long & Wang, 2015; Lu, Li, & Mou, 2014), natural language processing (Collobert & Weston, 2008; Duong, Cohn, Bird, & Cook, 2015) as well as speech recognition (Deng,

[45] Cognitive flexibility is also used to refer to other distinct characteristics of human cognitive function, including problem solving, planning and, in the task switching domain, the ability to rapidly *switch* between already acquired tasks (Goschke, 2000; Kiesel et al., 2010; Koch et al., 2018; Musslick, Jang, Shvartsman, Shenhav, & Cohen, 2018). In this section, we focus specifically on the ability to learn novel tasks from scratch.

Hinton, & Kingsbury, 2013). In line with intuitions from multi-task learning, results from Simulation Study 7 indicated that novel tasks were acquired much faster if the network was pre-trained on a set of similar, related tasks. Such positive transfer effects were promoted by leveraging existing representations for tasks that shared the same stimulus dimension.

The benefit of shared representation for transfer suggests a “snowball effect:” Once novel tasks build on existing representations, those representations are honed by more training signals, making them more useful for other tasks (Baxter, 1995) and further enhancing learning benefits as more tasks use those representations. The results we present here suggest that this is also associated with a correspondingly rapid increase in the potential for interference, and thus reliance on control and concomitant constraint on how many tasks can be performed at once. This suggests that the use of shared representations can provide a mechanistic account of the association between cognitive flexibility, control-dependence, and constraints on multitasking. This was evidenced in the results of the empirical study reported above (see Section “Empirical Study: Learning, Shared Representations and Functional Dependence”), in which participants were able to quickly learn a new task (i.e., map color words onto an arbitrary set of response keys) by using an existing set of (orthographic) representations, although this prevented them from being able to multitask this with another task (color naming) due to the sharing of those representations with an interfering task (word reading). These findings are consistent with a bias toward exploiting the advantages of shared representations for cognitive flexibility (i.e., learning and transfer), at the cost of constraints in multitasking capability and the potential demands for control that this may impose. Nevertheless, we did observe impairments in multitasking performance for tasks that were hypothesized to be independent. Such impairments may be attributed to other factors, such as meta-control processes required to determine the best strategy to execute two tasks in parallel (Liepelt et al., 2011; Fischer & Plessow, 2015; Jiang, Saxe, & Kanwisher, 2004), or a default bias toward suppressing other response modalities that is generically useful in single task performance.

3.2.6 Shared Representations, Semantics, and Multitasking. As noted earlier (in Section “Shared Representations and Multiple Resource Theory”), the principles governing the development of shared representations in the service of simple sensorimotor tasks may be closely related to those that govern the development and use of semantic representations. In general, the role played by control in the models presented here is comparable to that of the effects of higher level representations of context in models of language processing, such as the effects of words on letter perception (Plaut & Booth, 2000), or of previous sentential elements and/or discourse representations on interpretation of word meaning (McClelland, St. John, & Taraban, 1989). The present work provides suggests a direct relationship between reliance on such context effects and the nature of the underlying semantic representations that have been learned for different linguistic element (i.e., whether those are shared or distinct, e.g. T. T. Rogers & McClelland, 2004). More specifically, it suggests that the sharing of representations should be associated with corresponding reliance on control (i.e., relevant context representations), and even perhaps effects of constraints on multitasking in semantic inference that parallel those we have reported here for sensorimotor tasks. Support for this conjecture comes from a study of semantic interference in lexical decision tasks, that provides behavioral evidence for an association between the use of shared representation and constraints on multitasking (Chen & Rogers, 2010). In their study, Chen & Rogers deployed a dual-task paradigm to examine whether lexical decisions (e.g. word recognition) rely on semantic processing (or other non-semantic forms of linguistic processing, such as phonological and/or orthographic “lookup”). To investigate this question, participants were asked to perform a lexical decision task in conjunction with a sound judgement task. The lexical decision task required participants to press a key to indicate whether a string of letters was a word or a non-word. The sound judgement task required participants to categorize a sound using a verbal response. In the non-semantic condition, the participants performed the sound judgement task on complex tones to indicate whether the sound was ascending or descending in pitch. In the semantic condition, participants judged whether an animal sound was produced by a bird or not. The authors hypothesized that if the lexical decision relied on semantic representations then participants should perform worse in the visual lexical decision task when simultaneously executed with the semantic sound judgement task compared to the non-semantic sound judgement task[46]. The authors observed that the recognition of word stimuli was significantly impaired when executed in conjunction with the semantic sound judgement task compared to the non-semantic sound judgement task. Interestingly, the semantic sound judgement task exhibited less interference if the orthographic structure of the words provided a sufficiently strong cue for lexicality, suggesting that when access to the (presumably shared) semantic representations was not necessary, multitasking was easier. These results provide convergent support, from the domain of language processing and semantics, that reliance on shared representations for different tasks — even when they involve highly distinct modalities (such as lexical decision and acoustic judgements) — comes at the cost of a limitation in multitasking performance. In the General Discussion we elaborate on the potential of dual-task interference for investigating the nature of semantic representations.

3.2.7 Rationalizing the Trajectory From Controlled to Automatic Processing. While under many circumstances people prefer the speed of learning afforded by shared representations, there are clearly others in which they are willing to devote the time and effort to develop automaticity, In the final section of Part II, we reviewed recent work that directly examined the tradeoff between learning efficacy versus processing efficiency, including a model that provided a formal analysis of this tradeoff. While that model made a number of assumptions, it provides a promising foundation for a formally rigorous, normative theory of how people might chose between learning to perform a task quickly but at the expense of control dependence and seriality versus expending the additional time (and effort) to learn to perform it in a way that affords automaticity and the efficiency of multitasking. One of the important simplifications of that model was that each form of processing was learned

[46] Both sound judgement tasks were matched in terms of overall difficulty.

independently of the other. This was addressed in the study by (Ravi et al., 2020) described above, that explored the tradeoff between learning efficacy and processing efficiency in a deep learning network. That model, which involved the learning of representations at multiple layers of the network, also implemented a Bayes-optimal meta-learning mechanism responsible for deciding on each trial whether to train on single task or multitask performance. In that network, there were no constraints on the extent to which single task training transferred to multitasking or vice versa (the network was free to develop and use whatever representations it liked). The results replicated those of (Sagiv et al., 2018), with the meta-learner preferring single task over multitask training if the seriality penalty was low (and also, as noted above, if the environment was noisy), and conversely if the seriality penalty was high. In the General Discussion, we consider the question of meta-control within the broader context of the Expected Value of Control Theory (Shenhav et al., 2013, 2017), which proposes that the human cognitive system has the capacity to evaluate the portfolio of control-dependent tasks that it can pursue in any given setting, and select ones that it estimates will yield the greatest value, factoring in a cost of control.

All in all, the results presented in this article lay the foundation for a normative and mechanistic account of the trajectory from controlled to automatic processing: In novel and/or noisy environments, shared representations afford the ability to generalize what has been learned in other domains, thus enhancing cognitive flexibility. For example, people can quickly learn how to play a melody on a piano by using their knowledge of how to place fingers at designated locations. This reliance on existing representations comes at the cost of a seriality constraint: they can only be used for one purpose at a time (e.g., placing only one finger on the keyboard at a time). However, with sufficient motivation and time (e.g., the desire to become a concert pianist, and the opportunity to take lessons and practice) it is possible to acquire task-dedicated, separated representations that afford automaticity and the capacity for parallel processing (i.e., simultaneously and independently configuring all of the fingers required to play a given chord).

4 General Discussion

The limited ability to perform multiple control-dependent tasks at the same time is one of the most salient characteristics of human cognition, and is universally considered a defining feature of cognitive control (Posner & Snyder, 1975; Shiffrin & Schneider, 1977). Despite these facts, the *source(s)* of multitasking constraints associated with control have received considerably less attention in research than the observation itself. Here we build on the idea that multitasking limitations arise from shared representations between tasks (Allport et al., 1972; Allport, 1980; Kieras & Meyer, 1997; Kinsbourne & Hicks, 1978; Navon & Gopher, 1979; McCracken & Aldrich, 1984; Meyer & Kieras, 1997b; Walley & Weiden, 1973; Wickens, 1991) and provide a formal framework that permits studying the relationship between learned task representations and the multitasking limitations associated with controlled processing in neural architectures. The framework suggests that

- The multitasking capability of a network architecture decreases drastically with the amount of overlap among task representations (i.e., sharing) – an effect that is nearly invariant to the dimensionality of representations within layers of the network, and exacerbated by the number of layers. Moreover, the particular pattern of overlap among task representations can be used to predict the multitasking profile of the network as a whole. Taken together, these factors provide a quantitative grounding for multiple resource theory.

- The dependence among tasks induced by (1) shared representation, (2) the amount of conflict and (3) the persistence of representations provides a single mechanistic framework within which to account for the conditions under which parallel processing and concurrent multitasking capability is possible (at an extreme), and the rate at which tasks can be switched when serial execution is required. This in turn provides a coherent account for psychological phenomena, such as the PRP effect and performance costs associated with task switching, that have mostly been treated as distinct in the cognitive literature.

- Neural network architectures are subject to a fundamental tension between the sharing of representation that promotes efficacy of learning efficiency and generalization, and the separation of representations that permits parallel execution and interference-free multitasking. When trained on single tasks, neural systems exhibit a bias to learn shared representations in environments where there is shared structure between tasks, which in turn is associated with a seriality constraint on processing and a reliance on control to manage that constraint. Conversely, training explicitly on multitasking, or in environments in which task structure is not shared, networks favor the generation of separated (task-dedicated representations) that permit parallel processing, full concurrent multitasking capability and a minimization of reliance on control for those tasks.

- The foregoing factors provide a mechanistically explicit, and formally rigorous, and potentially normative account of the commonly observed trajectory in skill acquisition from controlled to automatic processing: When acquiring one or more tasks that share structure (with each other or existing ones), the immediate value of exploiting shared representations (faster acquisition) may usually be preferred over the future discounted value of increased multitasking capability and processing efficiency that comes with learning separated, task-dedicated representations, but at the expense of slower acquisition (and greater effort). Thus, on average, novel tasks are learned quickly, but at the expense of a seriality constraint and control-dependence. However, when it is deemed worthwhile through explicit training on multitasking (or possibly passively, with sufficient experience; see Footnote 43), separated representations can be acquired that afford parallel processing and multitasking capability – that is, automaticity.

In the remainder of this section we discuss the implications of these observations and their relationship to fundamental principles in other domains of cognition.

4.1 Relationship to Existing Theories of Dual-Task Limitations

There is a large literature on decrements in human performance associated with the attempt to execute two tasks simultaneously (Kahneman, 1973; Logan & Gordon, 2001; Meyer & Kieras, 1997b; Pashler, 1994), commonly referred to as dual-task interference. Broadly, three classes of theories have been proposed to account for the observed effects, each of which points to a different source of dual-task limitations: (1) structural bottleneck theories that attribute dual-task limitations to a central, structural bottleneck in processing that can process only a single task at a time; (2) capacity sharing theories that posit all tasks rely on a unitary, limited resource, and that parallel execution can occur provided the resource is sufficient, but that competition arises as it is depleted; and as we have discussed, (3) multiple-resource theories that assume dual-task limitations arise only when the two tasks rely on use of a shared *local* resource (i.e., specific to those tasks) for different purposes. The historical progression among these theories, and the empirical evidence that has been offered in support of each, is well reviewed in other work (e.g. Logan & Schulkind, 2000; Meyer & Kieras, 1997b; Pashler, 1994; Wickens, 1991). Here, we focus on the core assumptions of these theories, and compare them with the multitasking framework presented in this article.

4.1.1 Structural Bottleneck Theories. Structural bottleneck theories build on Telford's suggestion (1931) that organisms might be subject to a PRP that prevents the rapid successive execution of two tasks. Telford argued that the PRP is analogous to the refractory period of neurons that prevents the rapid initiation of an action potential immediately after a preceding action potential.[47] To explain the PRP and related findings (e.g. Craik, 1948; Vince, 1948), Welford (1952) postulated a central information processing channel that takes some "organizing time" to initiate a response

[47] The analogy is flawed in the sense that the refractory period of neurons is a recovery phenomenon whereas the PRP is thought to result from an actual bottleneck that precludes the second task from being processed *while* the first is still executing (Meyer & Kieras, 1997b). Moreover, the neuronal refractory period can be overcome by amplifying the input signal to the neuron. In contrast, the dual-task PRP does not seem to become shorter if the intensity of the second stimulus is increased (Pashler, 1994).

to information provided by a stimulus. Critically, Welford suggested that *"no two central organizing times can overlap, so that information from a stimulus arriving while information from a preceding stimulus is being dealt with has to be 'held in store' until the central mechanisms are free"* (Welford, 1952, p. 18).

This single-channel hypothesis assumes that humans can only process one task a time (Welford, 1952, 1967; Davis, 1959). While Welford postulated that the central channel *"deal[s] with the information provided by a stimulus and [. . .] initiate[s] a response to it"* (Welford, 1952, p. 18) it remained unclear whether the bottleneck encompasses stimulus perception and/or motor execution, leading to subsequent debates about the locus of the bottleneck. For instance, Broadbent's (1957) early-selection model of attention assumed that the bottleneck is located in the selection of task-relevant stimulus features. Conversely, Keele (1973) contended that tasks may be processed in parallel from perception up through response selection (see also Logan & Burkell, 1986; Norman & Shallice, 1986; De Jong, 1993), but that there is a bottleneck in response *initiation.* However, perhaps the most prominent, or at least enduring account of the single channel hypothesis localizes the bottleneck to the response selection process (De Jong, 1993; Pashler, 1984, 1994; Welford, 1967), described as a decision mechanism that *"converts the stimulus code to an abstract symbolic code for a physical response based on some set of innate or previously learned stimulus-response associations"* (Meyer & Kieras, 1997b, p. 4). The decision mechanism is assumed to be central in the sense that it is modality-independent; i.e. it handles response-selection for all tasks. Despite growing evidence against a structural processing bottleneck (see Section "Summary, Discussion and Conclusions for Part I"), the presumption of such a bottleneck has had a profound influence on thinking about dual-task interference.

The input, hidden and output layer of the neural network models presented in this article can be regarded as successive stages of processing. However, this model violates a core assumption of structural bottleneck model about the modularity of stages of processing: Once one is completed, factors influencing that stage cannot have any subsequent effects on processing of a task (Sternberg, 1969). In contrast, neural network models allow that processing at one layer can continue to influence processing in layers to which they project, including the output layer responsible for selecting a response (McClelland, 1979). In the models we have considered, this can happen until one of the output units reaches its response threshold. Simulation Study 3 showed that a neural network model with such continuous processing can exhibit effects comparable to the PRP, effects traditionally attributed to a structural bottleneck. Unlike other models that implement versions of the multiple resources theory (De Jong, 1993; Keele, 1973; Meyer & Kieras, 1997b), response initiation in our models *could* occur in parallel. Nevertheless, maximizing reward rate required that the network delayed initiating a response to the second task until cross-talk from the first task had sufficiently decayed. As noted in Section "Summary, Discussion and Conclusions for Part I", such cross-talk can arise from functional or structural dependence between tasks due to shared representations in the hidden layer. Insofar as the hidden layer links stimulus features to responses, much as the response selection does in other models, then the presence of structural and/or functional interference between two tasks can be thought of as imposing a response-selection bottleneck. As stated by Pashler, *"the predictions described [...] do not require strict successiveness and might well be compatible with selective influence on processes that normally operate in cascade (McClelland, 1979)). (Key predictions depend on the idea that once a stage is completed, factors selectively influencing that stage cannot have any later effects; in a cascade model, this would still be the case if a stage reached its asymptotic output level and then maintained that state for some period of time until following stages began to use that output.)"* (Pashler, 1994, p. 238).

Nevertheless, a key distinction between neural network models and traditional structural bottleneck models is that the former do not assume that a central constraint in processing obtains for all tasks (this would be tantamount to assuming that all tasks share representations at the hidden layer) – separated, task-dedicated representations can exist at a given layer that the corresponding tasks to be performed simultaneously with one another and/or others. Furthermore, such models provide a mechanistic and formally rigorous approach to understanding why and when tasks are likely to share representations, exhibiting what amounts to a structural bottleneck and, as posited by bottleneck theories, rely on control for execution.

4.1.2 Unitary Resource Theories. Structural bottleneck theories assert that attention cannot be divided between tasks. Troubled by this assumption, and the observation that under many conditions people *can* multitask, Kahneman (1973) and others (Navon & Gopher, 1979; Navon & Miller, 2002; Tombu & Jolicœur, 2003) proposed that attention constitutes a central resource that can be shared between multiple tasks, but that it has a limited capacity. According to Kahneman's theory, tasks such as naming the color of a Stroop stimulus rely on dedicated structures (e.g. for categorizing a color as green). Activation of a structure is assumed to depend on attention allocated to that structure, as well as the presence of a specified stimulus (e.g. a color patch), similar to a population of neurons coding for a task process. Attention is assumed to be limited and may be allocated in a graded fashion between structures.[48] Furthermore, allocation of attention is subject to voluntary control and the amount of allocated attention depends on the demands of the task(s) being executed. Kahneman assumed that increases in attention are generally insufficient to compensate for increases in task complexity, as well as the demands imposed by executing more than one task at a time. Thus, dual-tasking interference is primarily attributed to attentional demands of competing tasks. Norman and Bobrow (1975) elaborated Kahneman's theory, suggesting that, in addition to attentional limitations, task performance may also be "data-limited" which explains cases in which additional attention cannot improve performance (e.g. if the signal-to-noise ratio of the sensory input is too low). The assumptions that underlie unitary resource make distinctive predictions with respect to dual-task phenomena. The first prediction concerns the voluntary aspect of graded attentional allocation, suggesting that participants can trade off performance on one task against performance on another task when performing the two tasks simultaneously

[48] The limit itself is subject to momentary fluctuations and is assumed to be correlated with physiological indices of arousal, such as pupil dilation (Kahneman, 1973).

(Norman & Bobrow, 1975). Several studies provide support for this claim, showing that participants are able to trade off performance between two tasks[49] (Sperling & Melchner, 1978; Gopher, Brickner, & Navon, 1982). Another prediction concerns the presumption that dual-task interference reflects (global) “capacity interference”, that is, competing demands for a *central* capacity-limited mechanism. The latter suggests that dual-task interference should arise even if two tasks do not share any local resources, e.g. for perception or motor execution. To support this claim, Kahneman (1973) describes the observation that people often stop walking when asked to perform complex mental arithmetic, suggesting that walking and mental arithmetic cannot be performed simultaneously, despite seemingly independent neural circuits. The example, however, neglects the possibility that walking may involve navigational processes, and that those may draw upon representations shared with mental arithmetic (e.g. representations of space and linearity, see Section “A Mechanistic Account of Control-Dependent Versus Automatic Processing Based on Shared Versus Separated Representations” in the General Discussion). A third prediction is that dual-task interference depends *“in part on the load which each of the [tasks] imposes, i.e. on the demands of the competing [tasks] for effort or attention”* (Kahneman, 1973, p. 179). This suggests that increasing the complexity of one task should have detrimental effects on joint performance with another task. However, the prediction does not apply in all cases, as demonstrated by critics of central capacity-limited mechanisms (Wickens, 1991; North, 1977). For instance, increasing the complexity of a digit processing task was found to have no influence on the simultaneous performance of an object tracking task (North, 1977).

In his seminal work, Wickens (1991) outlined four behavioral phenomena that challenge the assumption of a unitary attentional resource. The first concerns a set of studies demonstrating that the difficulty of one task can have little to no effect on the joint performance with another (“difficulty insensitivity”, (Briggs, Peters, & Fisher, 1972; Johnston, Greenberg, Fisher, & Martin, 1970; Kantowitz & Knight, 1974;

[49] Note that smooth performance tradeoffs are also compatible with the assumption of a structural bottleneck (Pashler, 1994).

Kantowitz & Knight Jr, 1976; Wickens & Kessel, 1979). As outlined above, Kahneman's theory predicts that increases in complexity of one task should generally decrease performance of a second, simultaneously executed task, unless performance of the latter is data-limited (lower amounts of attention allocated to the task do not change it's performance; Norman & Bobrow, 1975; Wickens, 1991). Second, Wickens pointed out that a unitary capacity-limited resource cannot explain perfect time sharing, assuming executing multiple tasks requires a higher amount of attention than is available. The third criticism concerns sensitivity of dual-task interference to the compatibility of stimulus-response mappings between concurrent tasks (Greenwald, 1970; Greenwald & Shulman, 1973; Göthe et al., 2016; Oberauer et al., 2016; Halvorson et al., 2013; Hazeltine et al., 2006; Liepelt et al., 2011). This includes instances in which changes in the processing structure of one task (e.g. requiring a verbal instead of a manual response) alter interference with another task, even if the difficulty of the two tasks stays the same (Treisman & Davies, 1973; Wickens, 1991).[50] The fourth criticism regards observations in which the more difficult of two tasks brings about less interference with a third task than the easier one ("uncoupling of difficulty and structure", Wickens, 1991).

The neural network models presented here share at least three assumptions with Kahneman's theory: First, tasks structures (i.e. task representations) require both

[50] It is worth mentioning that Wickens' third criticism, i.e. that dual-task interference can vary as a function of task similarity, is accommodated by auxiliary assumptions of Kahneman's theory. As Kahneman notes, multitasking interference may also arise if tasks *"occupy the same mechanisms of perception or response"* (Kahneman, 1973, p. 196). Thus, Kahneman's theory does not only assume a unitary resource, but also multiple task-specific resources ("structures") that, when demanded by two tasks for different purposes, lead to structural interference. Kahneman further concedes that this assumption can accommodate effects of task similarity on dual-task interference, and thus, *"it is useful to retain the term of structural interference for situations of strong interaction between similar tasks, and to apply the label of capacity interference to situations where difficulty is the main determinant of results"* (Kahneman, 1973, p. 199). From this perspective, Kahneman's theory constitutes a special case of multiple resource theory, with the additional assumption of a unitary resource required by all tasks.

sensory input and control to be sufficiently activated, unless a task process is highly practiced. Second, multitasking interference arises when two tasks make competing use of a shared resource (i.e. set of processing units in the neural network). Third, it is assumed that the cognitive system can allocate cognitive control between tasks in a voluntary and graded fashion, based on the demands of the tasks and the needs of the agent.[51] However, several critical assumptions about the nature and role of cognitive control contrast with those of unitary resource theories. First, in our models, control is not constrained by some upper bound on its allocation, as long as there is sufficient influence of the control system over task representations (e.g. in the form of neural connectivity). This turns is a crucial factor, as it permits three of the four phenomena posed by Wickens (1991), that is, virtually perfect time sharing, insensitivity to task difficulty, as well as the decoupling of difficulty and structure. That said, as discussed in Section "A Mechanistic Account of Control-Dependent Versus Automatic Processing Based on Shared Versus Separated Representations" below, there may be practical constraints on how much control can be allocated, as a function of the current degree of representational sharing in the network; although even this can be mitigated by an investment in the acquisition of separated representations and automaticity, as discussed in Section "Multitasking Practice and Representational Separation" in the Summary and Discussion of Part II. This view leads to a different perspective on the role of control in multitasking interference: Contrary to Kahneman's theory, our work suggests that multitasking interference can arise from allocating too much control to too many tasks at the same time, rather than allocating too little control, since too much control brings about the risk of cross-talk between task processes.

4.1.3 Multiple Resource Theories. Multiple resource theories renounce the concept of a central processing bottleneck or unitary resource. Instead, they contend that a cognitive system is equipped with many independent, specialized resources and that different tasks rely on different such resources in various combinations. According to this class of theories, multitasking limitations are the result of conflicts that arise

[51] We assume that control is allocated such that reward rate is maximized.

when two or more tasks demand use of the same resource for different purposes at the same time. Instances of multiple resource theory vary in their assumptions about whether a resource can ever be shared between two tasks at the same time, and whether two tasks with different resources can interact with one another. Here, we review three types of multiple-resource theories before contrasting them with the present framework.

Early instances of multiple resource theory borrowed from Kahneman's notion of capacity limitation, suggesting that each resource has its own capacity that can be divided among several concurrent tasks (Navon & Gopher, 1979; Wickens, 1991).[52] A cognitive system would then supply resources to meet the demand determined by the desired level of task performance for each task, subject to constraints imposed by external and internal task parameters (e.g. predictability of the stimulus or task practice, respectively). Building on the ideas developed by Kahneman (1974) Norman and Bobrow (1975), Navon and Gopher (1979) and Wickens (1991), proposed a taxonomy of such resources, categorizing them into stages of processing (encoding, processing, responding), sensory modalities (visual, auditory), processing codes (verbal, spatial), as well as response modalities (manual, vocal).[53]

Other instances of multiple resource theory assume that each resource can only be executed by one task at a time (Allport et al., 1972; Byrne & Anderson, 2001; Meyer & Kieras, 1997b; Salvucci & Taatgen, 2008). For instance, using the symbolic architecture "EPIC", Meyer and Kieras (1997b) proposed multiple perceptual and motor processors, as well as a central cognitive processor and working memory. Operations in different

[52] Note that Navon and Gopher (1979) assume that the capacity of each resource is fixed and independent of task load, unlike the unitary resource proposed by Kahneman (1974).

[53] While Wickens (1991) assumed, multiple specialized mechanisms, he acknowledged the possibility of an undifferentiated, central mechanism "which is available to and competed for by all tasks, modalities, codes and stages as required [. . . and] may be assumed to represent that which is conventionally labelled attention, consciousness, the bottleneck, or the [limited capacity central processor] of the structural theories" (Wickens, 1991, p. 25). However, Wicken's also points out that the acknowledgement of a central capacity-limited mechanism does not deflate the value of the multiple resource concept.

perceptual processors can be carried out in parallel, however each can only be used for one purpose at a time, and thus conflict can arise when different tasks rely on simultaneous use of the same processor for different purposes (i.e. for processing incongruent information). In contrast, motor processors can only execute one task process at a time, irrespective of the information being processed. Unlike earlier instances of multiple resource theory, Meyer & Kieras entirely eliminated the assumption of central processing limitations, and allowed that the central cognitive processor could, in principle, execute an unlimited number of operations (called "productions") in parallel; it was constrained only by the potential for conflict among task-related actions, in which case it could strategically suspend execution (strategic response deferment) to avoid conflict among tasks in perceptual and/or motor processors. Byrne and Anderson (2001) proposed a similar model, referred to as "ACT-R/PM", in which they adhered to the assumption of the ACT-R framework (Anderson & Lebiere, 2014) that a central processor can operate only one task process at the time, and show that this model can account for effects concerning the PRP just as well as EPIC. Finally, Salvucci and Taatgen (2008) proposed a theory of threaded cognition which is based on a production rule architecture in which all resources (perceptual, cognitive and motor) were constrained to process only one request at a time. Unlike prior proposals, their threaded cognition model implemented the coordinative function of a general executive without dedicating a specific or central mechanism to it: the scheduling of task processes was distributed among the mechanisms responsible for execution of each task, following simple rules intrinsic to the architecture. For instance, tasks ("threads") were assumed to demand mechanisms in a "greedy" manner, as soon as they were needed, and release resources to other tasks in a "polite" manner, as soon as they were no longer required.

A third stream of multiple resource theories is often referred to as "cross-talk models". Cross-talk models assume that dual-tasking interference may occur even if the tasks involved do not directly compete for the same resource. For instance, Kinsbourne and Hicks (1978) proposed that the brain supplies tasks with limited "cerebral space" akin to the notion of a generalized processing. According to this account, much like the unitary resource theories, high performance on a task requires more cerebral space However, Kinsbourne's version adds that the closer the functional cerebral space for two tasks —measured in terms of the connectivity of associated brain regions— the more likely they are predicted to interfere with one another. Similarly, Navon and Miller (1987) suggest cross-talk between the processing channels of two tasks may lead to "outcome conflict", especially if the information content being processed in one task is incongruent with the information content being processed in another task, as is the case in the extended Stroop task described in this work. While Navon & Miller's proposition posed an interesting challenge for multiple resource theories, namely to account for the phenomenon that dual-task interference is dependent on the information content being processed, it did not come with a formal framework to test these predictions. Townsend and Wenger (2004) provided such a framework, and used it to study cross-talk in holistic cognitive processes, such as Gestalt-like phenomena. Similar to Navon & Miller (1987), they argue that cross-talk between different processing channels can be both facilitatory and detrimental, depending on the information content being processed (see Section "Interference Versus Facilitation" in the General Discussion). The interaction between resources, as well as the sensitivity of dual-task interference to the information content being processed is a distinct prediction of such cross-talk models.[54] However, Townsend and Wenger (2004) remained agnostic to the neural mechanisms underlying such cross-talk.

All three classes of multiple resource theories can account for a broad range of experimental phenomena, including ones that troubled unitary resource-models. Moreover, some of them are expressive enough to account for complex multitasking scenarios outside the lab, such as driving a car while attempting to dial (Brumby, Howes, & Salvucci, 2007; Brumby, Salvucci, & Howes, 2009; Salvucci & Macuga, 2002).

[54] Some instances of multiple resource theory assume that resources can be used by tasks in parallel if the information content being processed is congruent (Meyer & Kieras, 1997b; Byrne & Anderson, 2001; Salvucci & Taatgen, 2008). However, they lack a mechanistic explanation for this policy.

However, multiple-resource theories also face a number of theoretical concerns. First, unlike theories that posit a central processing mechanism, multiple resource theories must explain why multitasking appears to be so commonly limited to a small number of tasks (e.g. in the absence of limitations imposed by motor or sensory processes, why can we only have one stream of thought at the same time?), despite the enormous structural capacity of the human brain. In this light, it is perhaps not surprising that most multiple resource theories concede the existence of a central capacity-limited mechanism on which many, or even most processed rely (Byrne & Anderson, 2001; Navon & Gopher, 1979; Meyer & Kieras, 1997b; Salvucci & Taatgen, 2008; Wickens, 1991). Second, multiple-resource theories rely on auxiliary assumptions about the number and types of task-dedicated resources and are thus, less parsimonious compared to theories that posit a central limitation. Too much freedom in the choice of resources would make a multiple resource theory too flexible, and thus, unfalsifiable (Kinsbourne & Hicks, 1978; Navon & Gopher, 1979; Wickens, 1991). While some resource taxonomies are informed by effects of task-similarity on dual-task interference (e.g., Meyer & Kieras, 1997b; Wickens, 1991), this risks circularity (Treisman & Davies, 1973) that, to be avoided, requires more than behavioral criteria when deciding about the number and types of task-dedicated mechanisms. That is, there is a risk of adding an increasingly large number of auxiliary assumptions about resource sharing as the number of explained behavioral phenomena grows. Finally, multiple resource theory, let alone any account of the PRP, lack mechanistically explicit implementations above and beyond symbolic architectures.

The work presented in this paper addresses the theoretical limitations of multiple resource theory by leveraging the formalisms offered by neural network modeling. First, it provides a more stringent test of the multiple resource theory by evaluating multitasking capability in architectures that, *prima facie*, extensive resources (i.e., numbers of processing units and pathways) are available. Our finding that shared representation drastically limits multitasking capability, even in large networks, formally supports multiple resource theory in such settings. That is, neither the assumption of a central unitary resource, nor a single local resource bottleneck may be necessary to account for the striking limitations of human multitasking behavior. Second, it formalizes the construct of local resources as the source of constraints in multiple resource theory, in terms of the extent to which the representations used by different tasks overlap with one another; i.e., are shared. Furthermore, it directly relates this to statistical similarities between tasks in the environment, as well as the training regime: two tasks are more likely to share a representation if they rely on similar features, and if both tasks were acquired without pressure to perform them simultaneously.

Formulating the extent to which different tasks rely on shared resources also allowed us to extend the analysis from direct, structural interference on which most previous instances of multiple resource theory have focused, to the case of functional interference addressed by cross-talk models: Even if two simultaneously executed tasks don't directly share the same resources, they may still interfere with one another by means of a third task that introduces functional dependence between the two. The phenomenon of functional dependence, as illustrated in the extended Stroop task, results from the role of control in processing: It is assumed that, in order to execute a task, cognitive control needs to be allocated to the representations for that task. Allocating control to two structurally independent tasks (e.g. color naming and word mapping) may implicitly engage a third task (e.g. word reading) that shares a representation with one of the tasks (e.g. word representation shared between word mapping and word reading), and a representation with the other task (e.g. verbal output representation shared between color naming and word reading). We showed that multitasking performance can be reliably predicted from the measurement of such functional dependencies. Moreover, the present theory provides a mechanistic explanation for why dual-task interference depends on the content of information being processed. Interference between two functionally dependent tasks (e.g. color naming and word mapping) is predicted to be higher if the stimulus features relevant to the interfering task (word reading) are associated with a different response than the stimulus features relevant to the task subject to interference (color naming). We found evidence for this interaction in the extended Stroop task where dual-task interference between color-naming and word mapping was modulated by the response-congruency of colors and words. Thus, the neural network models presented in this article combine assumptions of classic, symbolic multiple resource models regarding structural interference with the assumption of functional dependence from cross-talk models.

4.2 A Mechanistic Account of Control-Dependent Versus Automatic Processing Based on Shared Versus Separated Representations

Multitasking limitations are a defining feature of control-dependent processing (Posner & Snyder, 1975; Shiffrin & Schneider, 1977). That is, cognitive control is defined to be associated with capacity limitations, generally assumed to reflect a dependence on serial processing, whereas automatic processes can operate in parallel (Shiffrin & Schneider, 1977). This has been interpreted as evidence that: (1) execution of control-dependent processes requires the engagement of a control mechanism (e.g a particular set of activated nodes in a short-term store (Anderson & Lebiere, 2014; Schneider & Shiffrin, 1977); a particular set of units in a neural network (J. D. Cohen et al., 1990; Botvinick et al., 2001; O'Reilly & Frank, 2006; Verguts, 2017); or a set of attentional weights (Logan & Gordon, 2001); and that (2) that mechanism is limited in how many control-dependent processes it can support at the same time (e.g., a limited number of nodes available in a short-term store, or competition among activated units responsible for control in a neural network). The latter has been commonly been interpreted, in turn, as evidence that the control mechanism itself is capacity-limited, (Anderson & Lebiere, 2014; Posner & Snyder, 1975). This view is in line with the single-channel hypothesis (Welford, 1952, 1967) reviewed above.

4.2.1 Constraints on Concurrent Multitasking. While the neural network models presented here implement the first assumption – that control-dependent processes require support of a mechanism responsible for control – it does not require the second assumption, that that mechanism has an intrinsic capacity constraint. Rather, control mechanisms are assumed to consist of a set of units that can encode the relevant task to be performed (corresponding to a task cue presented to the network), and use this to provide additional activity to units relevant to performing the task itself. While there is no structural constraint on the number of task units that can be activated at the same time, the capacity for control-dependent processing is functionally limited by the cross-talk that can arise when the tasks to be performed share processing units. That is, on this view, the purpose of control is to avoid such cross-talk, by limiting processing to only one of a set of tasks that share representations. From this perspective, constraints on control-dependent processing are a rational response to the presence of shared representations, and do not necessarily imply that such constraints reflect a limitation of the control mechanism itself. At the same time, this does not *preclude the possibility* of such limitations. One such possibility is that, given the prevalence of shared representations within certain domains of processing (e.g., ones that rely on abstract, highly general representations, such as language), and therefore the likelihood that engaging more than one task in such domains will lead to conflict, control mechanisms that serve such domains may have “hard coded” constraints on the number of processes that can be executed at once (e.g., as inhibitory weights among tasks units). From an implementational perspective, this could be viewed as a processing constraint within the control mechanism itself (that is, only one task representation could be active at a time). This might even be expressed in the brain by limitations in connectivity between areas responsible for control

4.2.2 Constraints on Task Switching and Serial Processing. The neural network models we have presented also account for constraints on the ability to execute multiple control-dependent tasks in rapid succession, in terms of the same mechanisms underlying constraints on concurrent multitasking, thus providing a unifying account of these phenomena. The former manifest as a cost in performance when switching from one task to another, as compared to repeating the same task (“switch costs”; Allport et al., 1994; Jersild, 1927; R. D. Rogers & Monsell, 1995). Theorists have proposed a variety of mechanisms to explain such switch costs, including, the demands of retrieving the relevant task goal for a given task cue (Logan & Bundesen, 2003; Logan &

Schneider, 2006), interference from competing stimulus-task associations (Waszak et al., 2004; Wylie & Allport, 2000), inhibition from the persistent activity of a previous task set in working memory (“task-set inertia hypothesis”, Mayr & Keele, 2000; Allport et al., 1994). While these have all largely been treated as distinct from the mechanisms responsible for constraints on concurrent multitasking (for a review, see Koch et al., 2018), the models presented here provide a common mechanism for both phenomena: persisting interference between tasks due to shared representation. This view reflects a combination of the multiple resource theory which assumes that shared representations between tasks pose a limit on the number of tasks that can be executed concurrently (Allport et al., 1972; Navon & Gopher, 1979; Wickens, 1991), on the one hand, and the task-set inertia hypothesis (Allport et al., 1994) which poses that such interference can persist when switching from one task to another. It is important to note, however, that this view is by no means a complete account of task switch costs. A large body of evidence suggests other mechanisms to contribute to performance costs involved in task switching, such as costs associated with active task reconfiguration (Mayr & Kliegl, 2000; Meiran, 1996; R. D. Rogers & Monsell, 1995; Rubinstein et al., 2001).

4.2.3 Cognitive Control and Flexibility of Processing. The present work suggests a functional connection between seriality constraints described above, and another defining feature of cognitive control: The ability to flexibly acquire and implement arbitrary mappings between stimuli and responses. Cognitive control is often defined as the latter (J. D. Cohen, 2017; Goschke, 2000; Verguts, 2017); that is, as a collection of mechanisms that support the learning and execution of novel task rules. Recent modeling work suggests that this flexibility in task acquisition can be achieved through oscillatory dynamics between existing task modules (Verbeke & Verguts, 2019; Verguts, 2017). For instance, Verguts (2017) shows that novel stimulus-response mappings (e.g. for pressing a button according to a word) can be acquired by synchronizing the neural population encoding the relevant stimulus feature (e.g. the word “red”) with the neural population encoding the desired response (e.g. “press right button”). Verguts argues that such flexible bindings would enable a cognitive system to acquire novel tasks in a rapid fashion, such as mapping words to button presses (see the extended Stroop experiment in Part II). Yet, flexible task acquisition can only be achieved in this way if existing neural populations for stimuli and responses are repurposed; that is, if they are shared between different tasks (Badre, Bhandari, Keglovits, & Kikumoto, 2020). The computational and behavioral studies in Part II of this article suggest that such repurposing of task representations across tasks enables a cognitive system to rapidly implement arbitrary task rules (i.e. cognitive flexibility) but that this comes at the cost of the seriality constraints imposed by overlapping task pathways. Thus, two defining features of cognitive control, constraints on multitasking and cognitive flexibility, may be two reflections of the same underlying factor: the use of shared representations for different tasks.

This characterization of control-dependent processing may also help explain why processes underlying language and mathematical reasoning, are subject to such striking limitations in multitasking, and considered prime examples of control-dependent processes. Both language and mathematical reasoning rely on the use of symbolic representations. Such representations, by their very definition, are general purpose; that is, they can be used for a wide — and in the limit, arbitrary and unlimited — number of tasks. By our reasoning, the more tasks that can make use of a symbolic representation, the more it should rely on control to determine how it is used in a given context. From this perspective, the very feature that makes the use of language and mathematical reasoning so powerfully flexible also explains why they are so canonically representative of control-dependent, serial processing.

Conversely, automatic processes are defined to be free of interference in that they can be executed concurrently with other tasks. The work we have presented here suggests that, when a task is deemed to be automatic, it is because it is being executed in a setting in which it is independent of any of the other tasks called upon for execution in that setting; that is, the representations on which it relies are not shared with any of those other tasks. This is consistent with previous arguments that automaticity is best thought of as a relative attribute, that is based on the strength of the processing pathways required to perform the task, relative to that of other that may be competing with it (e.g., word reading vs. color naming in the Stroop task (e.g., J. D. Cohen et al., 1990). Here, in addition to the relative *strength* of processing pathways, we add the degree to which it shares representations with those other tasks (that is, the extent to which the pathways overlap) as a factor that determines the automaticity of a task in a given setting. This suggests that tasks that share representations with many other tasks are less likely to be automatic (i.e., they are *more* likely to rely on control – a factor that, as discussed just above, may help explain the profile of tasks such as language processing and mathematics that are prototypically control-dependent tasks. It also provides an account of the common trajectory in skill acquisition from control-dependent to automatic processing.

It is a longstanding observation that when many tasks are first acquired they appear to rely on control, as evidenced by their susceptibility to multitasking interference, but can become free of interference from other tasks with sufficient practice (Logan, 1978, 1985; Schneider et al., 1987; Schneider & Chein, 2003). Modeling efforts relying on symbolic architectures such as ACT-R suggest that continued practice on a task leads to improved scheduling of task processes through a central executive (Kieras et al., 2000), the compilation of sub processes into smaller chunks (Newell & Rosenbloom, 1981; Rosenbloom et al., 1993; Taatgen & Anderson, 2002), or improved memory retrieval of task-relevant information (Logan & Bundesen, 2003). These models suggest that interference-free task execution is primarily achieved by gradually reducing *temporal* overlap between task processes in a given resource. Simulation studies in Part II suggest another mechanism by which practice on a task may lead to automaticity: by separating representations between interfering tasks. As discussed in Simulation Study 5, evidence consistent with this has been observed in a functional neuroimaging study (Garner & Dux, 2015). Future models of skill acquisition may therefore benefit from combining mechanisms that underlie reductions in temporal overlap, as proposed by production system architectures, as well as mechanisms of reducing overlap in task representations as suggested here. Furthermore, as discussed above, reliance on shared representations may provide a normative account of why tasks are so often acquired in a form that depends on control: exploiting the use of pre-existing representations permits more rapid acquisition (i.e., cognitive flexibility), albeit at the expense of greater dependence on control and serial processing that comes with the use of shared representations.

4.3 The Relationship of Cognitive Control to Working Memory

One approach to explaining the constraints on multitasking of control-dependent processes has been to attribute these to the reliance for control on a central limited-capacity working memory mechanism. While this is not explicitly specified by most central bottleneck theories (Welford, 1967; Pashler, 1994), it would provide a mechanistic account of the capacity constraint associated with control. It assumes that task representations required to exert control over processing (i.e., goals, instructions, and/or other forms of context information needed to specify the task) must be actively maintained, and that doing so relies on a centralized working memory mechanism that subserves the control system, and that is assumed to be subject to a strict capacity constraint (Cowan et al., 2012; Kriete et al., 2013; Luck & Vogel, 1997; G. A. Miller, 1956; Schneider & Detweiler, 1988). Interestingly, however, while all computational accounts of control-dependent processing incorporate some mechanism(s) of working memory, they do not generally assume that is a single, centralized mechanism.

For instance, symbolic processing systems, such as ACT-R (Anderson, Reder, & Lebiere, 1996; Anderson & Lebiere, 2014) and EPIC (Meyer & Kieras, 1997b) define working memory as the set of propositional representations currently active in declarative memory. In those frameworks, while it is assumed that there is a limit on the amount of activity available to representations in declarative memory, it is also assumed that declarative memory itself may be subdivided into domain-specific modules e.g., Anderson et al. (1996). Thus, while there may be limitations in the amount of activity available for WM within each domain, this does not place a constraint on the number of control-dependent processes that can be executed *across* domains, other than the number of domain-specific modules, to the extent that each makes use of WM in a different domain.

Nevertheless, although most computational frameworks do not tie the seriality of control-dependent processing to a *centralized* working memory mechanism, limitations in the capacity of working memory are relevant to control in at least two important ways. One is simply a re-expression of the point that has been the focus of this article: To the extent that working memory refers to the set of representations activated within a given resource, and this is limited – whether by assumption as in production system models, or due to interference as in neural network models (see Footnote 1) – then any processes that require different representations to be activated within that resource must be executed serially and therefore rely control. The role of representation sharing for limitations in working memory capacity is illustrated in a recent neural network model by Bouchacourt and Buschman (2019). The model consists of two layers: a sensory network that is composed of independent sub-networks, each dedicated to represent a visual object in a different location in space; and a separate network that is randomly and reciprocally connected to the sensory network. Representations for stimuli in the sensory network lead to corresponding activations in the random network that feed back to the same representations in the sensory network. This reciprocal connectivity ensures that representations for stimuli are maintained, despite removal of external input (the stimulus) to the sensory network. The random connections ensure that the network is flexible enough to represent arbitrary sets of stimuli. However, as a consequence, stimuli from different sensory sub-networks can share representations in the random network. The authors demonstrate that such representation sharing can lead to interference between items, limiting the number of objects that the whole network can maintain.

As we have emphasized throughout this article, in cases such as those described above, the seriality constraint should really be thought of as a reflection of the *purpose* of control, rather than a limitation intrinsic to the mechanism(s) responsible for its execution. However, there are cases in which seriality might be construed as reflecting a constraint that is in fact intrinsic to the mechanism(s) responsible for control. This is when information required to control two or more tasks (e.g., task goals, instructions, and/or other context information necessary to specify the tasks) must itself be represented within the same resource. In this case, the activation of such information is presumably subject to the same working memory limitations that constrain any other resource, as illustrated in the network model of Bouchacourt and Buschman (2019). A similar situation can apply to the models presented in this work. For example, in the models presented in Parts I and II, units in the task layer (e.g., representations designating the dimension of the stimulus to which to respond in the Stroop paradigm) may compete with one another (as a result of feedforward cross-inhibition and/or recurrent inhibitory weights within the task layer) , thus constraining only one to be active. While there is no reason *a priori* that this must be so, the system may learn over experience that for tasks that share representations it is best not to perform both at once, and thus develop inhibitory connections among the relevant task representations. This is particularly likely for tasks that rely on representations that are shared among many processes (as are those for colors and orthography in the Stroop task; we consider this issue more generally with respect to the binding problem in perception, in the section titled "The Binding Problem, Attention, and Shared Representations" below). In such cases, the sets of such mutually exclusive task representations can be thought of a resource (see Footnote 1), used to execute control, and the limitation on the number of representations that can be active within that resource could be described as a WM limitation and considered to be a constraint that is intrinsic to the mechanism responsible for control itself. However, note that this constraint reflects an adaptation that arose from the sharing of representations among the tasks over which those representations preside. Thus, while in a structural sense the constraint is intrinsic to the mechanism responsible for control, it can still be traced *functionally* to the sharing of representations among the tasks over which it presides. Furthermore, because the resource shared by one set of task representations need not be the same as the ones shared by others (e.g., along lines similar to domain-specific modules in production system architectures), it seems reasonable to consider such constraints as also falling within the explanatory purview of the multiple resources theory, rather than as reflecting a single, centralized, capacity-constrained bottleneck in processing.

4.4 The Binding Problem, Attention, and Shared Representations

The relationship of between cognitive control and shared representations may also help provide a mechanistic account of the long recognized relationship between visual attention and the feature binding problem in perception (Treisman, 1996, 1999). The feature binding problem concerns the identification of stimulus features with the objects to which they belong. This arises when multiples objects (e.g., a blue square and a yellow circle) are present in the stimulus at the same time: How are the features (i.e., the colors and shapes of each) represented in such a way that each is assigned to the correct object (e.g., without misperceiving a blue circle and yellow square)?

One solution to the binding problem is the conjunctive coding of features; that is, representing each object as an explicit conjunction of its features. While this solution prevents any misattributions, it requires an encoding of all possible combinations of features to accommodate the range of possible objects, which grows combinatorially with the number of dimensions and features along them. In the limit, this leads to the classic "grandmother" cell problem: The requirement for a unit dedicated to every possible object (e.g., a different unit for a blue square, yellow circle, as well as blue circle, yellow square, orange triangle, etc. Barlow, 1972; Riesenhuber & Poggio, 1999). It is questionable whether a system, even as large as the human brain, can accommodate the number of combinations needed. However, to the extent that conjunctive encoding is used, it should be possible to recognize multiple objects in parallel, without crosstalk among their features. This is sometimes the case (Cave & Wolfe, 1990; McLeod, Driver, & Crisp, 1988). However, often it is *not* the case, as has been well established in a landmark series of studies showing that, under many conditions, object recognition seems to require serial search (Shiffrin & Schneider, 1977; Treisman & Gelade, 1980; Woodman & Luck, 2003) and, furthermore, can be subject to the kinds of misattributions indicative of the binding problem. This has been interpreted as evidence that the visual system can also use a different solution to the binding problem, as proposed by Feature Integration Theory (FIT, Treisman & Gelade, 1980). FIT suggests that object recognition can rely on representations of features that are general to all objects – sometimes referred to as "compositional coding," to reflect their using in various combinations to represent different objects – by attending to only one object at a time. This ensures that only the features within the limited focus of attention are integrated into that object, avoiding misattributions of the features belonging to other objects. However, this comes at the expense of serial search over objects to identify each.

This summary above should make clear the parallels between the binding problem in object recognition and the "multitasking problem" in task performance (for a similar argument, see Logan & Gordon, 2001). Conjunctive feature coding for individual objects corresponds directly to what we have referred to as separated (tensor product) representations dedicated to particular tasks, with its attendant representational demands but efficiency of parallel processing; while "compositional coding" corresponds to the use of shared, general purpose (minimal basis set) representations that are both more flexible and representationally more efficient, but come at the cost of a requirement for serial processing to avoid feature misattribution or interference. In object recognition, serial processing (e.g., visual search) is assumed to rely on the allocation of attention, that in turn is assumed to rely mechanisms of control (Shiffrin & Schneider, 1977; Treisman & Gelade, 1980). Similarly, as discussed extensively in this article, the serial execution of multiple tasks is assumed to rely on the execution of control. It seems reasonable to suggest, therefore, that the value of shared representations and their relationship to control-dependent processing reflect general principles of processing in neural network architectures, that apply equally across domains, including perception and action. That is, the problem of simultaneously detecting multiple objects can be thought of as comparable to, and governed by the same principles as, the problem of executing multiple tasks at the same time (Logan & Gordon, 2001). However, there have been differences of emphasis and interpretation across these domains.

In the context of object perception, the role of attention has been proposed to be the integration of features with the representations of objects (Treisman & Gelade, 1980). Something similar might be said of the role of control in task performance, in mapping features onto desired responses. However, where attention has been implicated in object recognition, seriality of processing has been interpreted as reflecting constraints associated with control mechanisms responsible for directing attention (e.g., to guide visual search), resting on the traditional assumption that control mechanisms are associated within an intrinsic processing capacity limits (Shiffrin & Schneider, 1977). The work we present here provides an alternative view: Seriality of processing is a solution to the binding problem imposed by attention, rather than a reflection of its limitations, just as control is a solution to the problem of interference in task performance. In both cases, the solution is required to avert problems that arise from the use of shared representations.

This perspective may also help explain why and when compositional representations may be favored over conjunctive ones, despite binding problem and the requirements it carries for serial processing and attention: By relying on shared representations, compositional encoding facilitates learning and transfer. In the case of object detection, conjunctive codes of features (e.g. modular encoding of colors and locations) support spatial invariance (e.g. the ability to detect the color of an object irrespective of its location), and are commonly observed across stages in the visual system (Desimone, 1991; Tanaka, 1996; Rolls & Tovee, 1995). Neurally inspired mechanisms for spatial invariance have also enabled recent advances of artificial object recognition (LeCun et al., 1989; LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). That is, object recognition may face the same tradeoff between learning efficiency promoted by the sharing of compositional representations of features across multiple objects, versus more efficient, simultaneous detection of multiple objects afforded by conjunctive representations. This may also provide insight into how object representations develop in the brain. For example, learning about a new object (i.e., involving a new combination of features) may exploit compositional rather than conjunctive representations, committing dedicated representations to individual objects only after considerable experience, or when parallel recognition of multiple objects is important.

4.5 Interference Versus Facilitation

In this article we have focused primarily on the deleterious effects of shared representations with respect to processing efficiency; that is, the potential for interference. This assumes that when two or more tasks make use of shared representations, the specific representations they require differ, and thus interfere with one another (e.g., an incongruent Stroop stimulus). However, it is also possible that different tasks may require the same representation (e.g., a congruent stimulus), in which case shared representations should produce facilitation that would *improve* rather than degrade performance. Here, we first motivate this focus, but then consider conditions under which facilitation arising from shared representations may be a relevant factor.

Our focus on interference was guided by the observation that, in general, the conditions that favor facilitation due to shared representations are far less likely than those favoring interference, under the assumption that, in general, the features along different dimensions of a stimulus are statistically independent of one another. For example, consider a Stroop stimulus in which the two relevant dimensions (colors and words) may each take on one of three features (e.g., red, green or blue). Assuming uniform, independent sampling along each dimension, stimuli are twice as likely to be incongruent as congruent (2/3 vs. 1/3). This asymmetry grows exponentially as both the number of dimensions and features within each dimension grows. Thus, it seems reasonable to assume that, in realistically rich environments, the likelihood of congruence among tasks that share representations is low. Furthermore, it has often been observed that facilitation effects due to congruence are substantially smaller in magnitude than those of interference (D. S. Lindsay & Jacoby, 1994; Macleod, 1998). Although the reasons for this are beyond the scope of this article (for potential accounts, see J. D. Cohen et al., 1990; Herd, Banich, & O'Reilly, 2006; Logan, 1980), this too suggests that it is reasonable to consider the potential costs of interference due to shared representations as outweighing, on average, the potential for facilitation.

Nevertheless, there are some conditions under which shared representations can lead to facilitation that are relevant not only to single task, but also multitasking performance. For example, Townsend and Nozawa (1995); Townsend and Wenger (2004) have shown that, under certain conditions, a task process can execute faster if it is performed in conjunction with other task processes compared to when it is performed alone, and referred to this as "super capacity". Formally, a parallel processing system is assumed to reach super-capacity if the probability $P_{AB}(T_A \leq t \text{ AND } T_B \leq t)$ of reaching a response for two processes T_A and T_B before time point t exceeds the probability $\min[P_A(T_A \leq t), P_B(T_B \leq t)]$ of responding to the slower of the two processes before time point t.[55]. The work presented in this article suggest that this can arise from shared of representations in the same way that stimulus congruence can produce facilitation in single task performance. In the latter, the features of the stimulus relevant to the task to be performed and another one are both associated with the same representation within the set that is shared, so that any partial activation provided by the irrelevant task reinforces the representation needed to perform the relevant task (e.g., J. D. Cohen et al., 1990). In the context of dual task performance, such facilitation will produce performance that is better than when each task is performed in isolation of the other; that is, when *no* information is available along the other dimension (e.g., naming the color of patch or the letters XXX).

Our graph-theoretic analysis of shared representations, and in particular the construct of functional dependence, may also have relevance to associative processes and their relationship to measurements of creativity. The latter have been operationalized in the form of the Remote Association Test (RAT, Mednick, 1962), in which participants are presented with three cue words (e.g. "home", "sea", "bed") and

[55] This condition represents a violation of an inequality formulated by Colonius and Vorberg (1994). The violation of this inequality is sufficient but not necessary for super-capacity.

are asked to identify a solution word that relates to all of the three cue words (e.g. “sick”). Performance on this task has been interpreted in terms of a semantic graph, in which nodes represent individual words and the edges between nodes represent the semantic association between them (Kajić et al., 2017; Schatz et al., 2018). The ability to retrieve the solution word is assumed to depend on how effectively activity spreads from nodes representing the cue words to the node representing the solution word and, in particular, to ones that are not directly connected. This might be viewed as a form of associative facilitation that arises from chains of shared representations. If so, the graph theoretic methods we described for evaluating functional dependence may provide a formal approach to quantifying such effects in neural networks. In such networks, concepts are generally represented as distributed patterns of activity rather than discretely as individual nodes. However, the methods we described for constructing a bipartite graph from a neural network (see Section “Graph-Theoretic Analyses”) could, in principle, be used to construct a semantic graph from semantic neural networks (e.g., Hinton et al., 1986; Kajić et al., 2017; Schatz et al., 2018; T. T. Rogers & McClelland, 2004); and, from that, to construct an interference graph that could be used to determine functional dependence – that is, the prevalence of indirect sharing that could be used for inference. That, in turn, could used to predict scores in the RAT, providing a bridge from detailed, process models of semantic cognition to measures of associative abilities and creativity.

4.6 Learning, Memory and Semantic Cognition

The computational tradeoff between shared and separated representations is closely related to another, well characterized computational dilemma in neural architectures: the tension between the ability to rapidly acquire new information without interfering with or over-writing existing knowledge (McCloskey & Cohen, 1989). This problem, known as “catastrophic interference” can be avoided by biasing a neural network towards non-overlapping (sparse) representations (French, 1999). Biases towards interference-free learning through the use of sparse representation, however, forgo inference and transfer to novel tasks that is achieved through interactions between learned representations. This dilemma motivated the Complementary Learning Systems (CLS) hypothesis, according to which two separate learning systems interact in the human brain, one that relies on shared representations to support inference (semantic memory, subserved by neocortical structures), and another that uses separated representations to support independent encoding and retrieval of information (episodic memory, subserved by medial temporal structures, possibly among others)(McClelland et al., 1995). This suggests that the limitations associated with interference-free processing in the domain of cognitive control may reflect the same underlying dilemma posed by the problem of catastrophic interference in the domain of learning and memory. How these solutions relate to each other is a potentially important future direction of research. For example, how might separated representations that can be rapidly formed in the medial temporal cortex structures (i.e., episodic memory) interact with both shared (minimal basis set) and separated (tensor product) representations that can be formed in neocortex (i.e. semantic memory). Better understanding how such interactions might help explain the remarkable flexibility characteristics of human behavior. Interesting, this is a direction that has begun to attract the attention of work in machine learning (Graves, Wayne, & Danihelka, 2014; Lake, 2019; Ritter et al., 2018; Webb et al., 2020).

A related issue concerning representational learning is the transfer (inductive generalization) of concepts in semantic cognition (e.g. reasoning from multiple instances of birds that all birds lay eggs; Abel et al., 2015; Jackson, Rogers, & Ralph, 2019; Ralph, Jefferies, Patterson, & Rogers, 2017). Here, we have argued that shared representation across tasks facilitates inference and transfer in control-dependent processing. Similarly, in semantic cognition shared representation across stimulus modalities and contexts can achieve transfer of concepts (Jackson et al., 2019; T. T. Rogers & McClelland, 2004; Rumelhart et al., 1993). In their recent work, Jackson et al. (2019) showed that the latter is facilitated in networks that allow information from different modalities to converge in the same “hub” for shared representation. Interestingly, the acquisition of semantic concepts follows a developmental trajectory which, at the level of representation learning, resembles the trajectory from controlled to automatic processing described above. That is, children were observed to learn broad semantic distinctions (e.g. between living and non-living things) earlier than more fine grained distinctions (e.g. between a sheep and a goat; Mandler, Bauer, & McDonough, 1991; Pauen, 2002). Neural network models, similar in architecture to the one described here (Rumelhart et al., 1993), suggest that this behavioral trajectory underlies a transition from representation sharing across categories to the separation of category-dedicated representations (T. T. Rogers & McClelland, 2004), a transition that reflects the progressive extraction of common statistical structure across objects (A. M. Saxe et al., 2019).

If the principles and methods – discussed in this article with respect to sensorimotor tasks – also apply to semantic inference, then it should be possible to use dual-task interference as a novel, and potentially sensitive probe of semantic representations (as discussed in Section “Shared Representations, Semantics, and Multitasking”, see Chen & Rogers, 2010). For example, one might ask whether size judgements of different semantic categories (such as animals and furniture) rely on a shared, canonical representation for “size”? This could be addressed using a semantic version of the extended Stroop task described in Section “Empirical Study: Learning, Shared Representations and Functional Dependence”, in which participants are presented with a picture of an animal (e.g. a hamster) and a word overlaying the picture designating a furniture (e.g. “CHAIR”). Participants could then be asked to name the color of the animal in the picture while indicating the size of the furniture with a button press. If animals and furniture share the same representation for size judgements, then one may expect dual-task interference in conditions where the presented animal and furniture don’t match in size. Such a dual-task experiment may permit inferences about the amount of representation sharing between different semantic categories (e.g. animals and furniture) with respect a particular feature dimension (e.g. size), and thus, may provide a novel avenue for the study of semantic cognition.

4.7 Bounded Rationality, Normative Models of Control Allocation and the Cost of Control

Bounded rationality refers to the proposition that aspects of human cognition and behavior, which appear irrational when viewed through the lens of simple formal analysis (Kahneman & Tversky, 1972; Tversky & Kahneman, 1974), may in fact reflect rational adaptations to constraints under which the system operates, and be found to be optimal or near optimal when these are taken into consideration (Gershman, Horvitz, & Tenenbaum, 2015; Gigerenzer, 2008; Griffiths, Lieder, & Goodman, 2015; Griffiths & Tenenbaum, 2006; Lewis, Howes, & Singh, 2014; Simon, 1957; Todd & Gigerenzer, 2012).[56] Here, we consider how the work presented in this article situates our understanding of cognitive control within this framework.

Recently, there has been renewed effort to frame cognitive control as an optimization problem, inspired by early work on control theory in engineering (Wiener, 2019), its application to psychology (Atkinson & Shiffrin, 1968; G. A. Miller, Galanter, & Pribram, 1960), as well as work in computer science on bounded optimality (Russell & Subramanian, 1994). The latter proposes that an agent maximizes reward per unit time given the limitations of its computational architecture (Russell & Subramanian, 1994). With respect to cognitive control, the primary limitation has been assumed to be constraints on its allocation. Kurzban et al. (2013) proposed that these constraints impose an opportunity cost on the allocation of control, that may help explain subjective phenomena with which it is associated, such as mental effort and fatigue: These may reflect internal signals that signify the cost of allocating control to one process in terms of the opportunities that are forgone for doing so to others (see also (Agrawal, Mattar, Cohen, & Daw, 2020; Shenhav et al., 2017) for formal treatments of

[56] This general idea has been expressed using other terms, such as “satisficing,” “resource rationality,” and “bounded optimality.” While these terms reflect some differences in approach and/or emphasis, those differences are beyond the scope of the present article. Here, we focus on the fundamental idea they have in common: that a consideration of the constraints under which the system operates can lead to a deeper understanding of the determinants of its function.

this idea). This, in turn, has led to the development of theories that formulate the allocation of control allocation in terms of a cost-benefit analysis, that selects among candidate tasks the one(s) that promise the greatest returns, by weighing the expected value of investing in each against the costs of doing so (i.e., forestalling or foregoing others). This idea has been expressed in general form as the Expected Value of Control theory (EVC, Shenhav et al., 2013), and formalized in a number of settings, including the selection between cognitive heuristics (Lieder & Griffiths, 2015), model-based planning (Kool et al., 2017), and the learning of the value of control (Musslick, Shenhav, Botvinick, & Cohen, 2015).

This EVC Theory, and related approaches provide a rational account of control allocation under the assumption that capacity is bounded; that is, the allocation of control carries opportunity costs. However, it does not provide an account of the bound *itself*; that is, *why* is the allocation of control is limited? Here, we provide an answer to that question, that suggests a more nuanced formulation of the problem faced by the control system, and its relationship to mechanisms of learning. Constraints on the allocation of control, and attendant opportunity costs, arise from a rational adaptation to another form of cost: the risk of interference associated with shared representations. We have argued that this, in turn, reflects another form of adaptation, favoring the efficacy of learning over the efficiency of processing. This account not only provides a mechanistic understanding of the conditions under which control is required (when the tasks under consideration share representations) and a normative account of its engagement (to optimize performance by minimizing the risk of conflict), but also ties this to a normative account of why such conditions may arise (a bias toward the efficacy of learning over the efficiency of processing). From this perspective, capacity constraints associated with control-dependent processing are a bound rationally by control, necessitated by the use of shared representations in the service of more effective learning and generalization. To impose a rational bound on control, the brain may rely on meta-control mechanisms for estimating its constraints on multitasking capability, the study of which remains an important objective for future research. In addition, the brain may balance learning efficacy against processing efficiency, possibly through meta-optimization. We reviewed possible mechanisms for the latter in Section "Summary and Discussion of Part II". These suggest that it can be optimal, under finite time horizons, for neural agents to harvest immediate rewards from tasks that are learned quickly, at the cost of having to execute them in serial (Sagiv et al., 2018; Ravi et al., 2020).

While the work presented in this article provides a rational basis for the opportunity costs associated with the allocation of control that arises from a constraint in the *number* of tasks to which control can be safely allocated, there also appear to be costs associated with the *intensity* of control allocated to a task. This is evidenced by the observation that people can exhibit aversion to the allocation of control even to a single task (Kool, McGuire, Rosen, & Botvinick, 2010; Westbrook & Braver, 2015). This is puzzling from a normative perspective: Why would a system refrain from allocating maximal control to a task to which it is already committed, assuming that performance scales with the intensity of control allocated? One answer to this question that has been proposed is that this reflects another tradeoff faced by control mechanisms, referred to as the stability-flexibility dilemma, that has been been formalized in terms of the dynamics of processing in neural networks (Durstewitz & Seamans, 2008; Musslick et al., 2017; Ueltzhöffer, Armbruster-Genç, & Fiebach, 2015): Increasing the activity of the representation(s) responsible for control of a task may improve performance of that task and make it more robust to interference. However, this will also induce greater persistence of activity of those representation(s), and the ones in the pathways responsible for task execution. As discussed in Section "Performance Costs Associated with Task Switching" in the Summary, Discussion and Conclusions for Part I, this can incur greater switch costs, that will impair performance in settings requiring the flexibility to rapidly switch between tasks. Note, however, that as shown in Simulation Study 3, such costs scale with the extent to which representations are shared among the tasks involved – that is, the extent to which they are control-dependent.

Musslick, Bizyaeva, Agaron, Naomi, and Cohen (2019); Musslick et al. (2018)

have illustrated these effects, and their ability to reproduce effects observed in human performance, using a model that implemented control representations as attractors in the recurrent layer of a neural network. Furthermore, they showed that constraining the activity of control representations was optimal (i.e., yielded higher overall rates of reward) in environments with a higher demand for switches between tasks. These observations suggest that constraints on the intensity of control can be a rational response in environments that require flexibility, and that this may be signalled by the costs associated with intensity of control allocation. More generally, the framework presented in this article provides a unified understanding of the costs associated with control – both in the number of tasks to which it is allocated and the intensity allocated to each – showing how these relate to (and scale with) the use of shared representations, and reflect the value placed on flexibility by the cognitive system in the ability to acquire new tasks and switch between them.

4.8 Machine Learning and Artificial Intelligence

As noted at several points in this article, the observation that shared representations promote more rapid learning and generalization has become an important foundation of machine learning methods that make use of neural network architectures. Such methods are largely concerned with building artificial agents that can generalize what they learn from observed (training) data to unseen (test) data. One challenge to doing so has been characterized as the bias-variance tradeoff, which is closely related to the tradeoff between shared and separated representations. The bias-variance tradeoff refers to the problem that can arise from overfitting, in which generalization and transfer performance are impaired if a learner has more parameters than data points, as is often the case for neural network architectures. If a network is equipped with too many parameters (i.e., processing units and/or connections), it can accommodate the variance in the data by simply memorizing all of the data points with a dedicated parameter for each, without encoding any relationships that might exist among them. This corresponds to the formation of separated, tensor product representations as described in this article. This can occur even when the space of parameters is smaller than the number of data points, if the learner has trouble segregating meaningful structure from noise in the data. Absent any biases, if neural networks are given too many parameters, they are known to overfit; that is, to tune their parameters to fit all of the variance in data, including any due to noise. This can also happen if they are trained for too many epochs on the same data set. To prevent this, machine learning researchers introduce biases, constraining the space of parameters using various regularization techniques, which reduce the degree to which the network adjusts its parameters to variance in the data. While these biases can reduce the learner's susceptibility to variance caused by noise by the data, and help it discover lower-dimensional structure more quickly (for example, shared, or minimal basis set representations) it can also cause it to miss discovering meaningful higher-dimensional structure. This is known as the bias-variance tradeoff.

The bias-variance tradeoff can help provide insight into the factors that influence the development of shared representations. For example, training on multiple tasks ("multi-task training", such as we used in Simulation Study 7) can be interpreted as an inductive bias that reduces noise which might otherwise obscure the shared structure across tasks if they are learned in isolation of one another, by allowing this to be averaged over over training, and thereby learn a shared representation that corresponds to the average (Caruana, 1997; Ruder, 2017). This use of low-dimensional representations can be formalized as a bias of the learner's hypothesis space (Baxter, 1995), that is, the set of all hypotheses a learner may use to acquire new tasks. In the analyses in Simulation 7, we formalized the hypothesis space in terms of the number of distinct task representations encoded in the hidden layer of a network, and showed that a small number of shared representations (i.e. the minimal basis set representation) facilitated transfer to novel tasks. However, we also showed that biases toward representation sharing introduce a systematic error when multiple tasks are executed concurrently. Separated representations, on the other hand, increase the representational complexity of the network, thereby making the network more susceptible to noise in the training data. Yet, separated representations avoid systematic errors (i.e. cross-talk) when the network is tasked to execute multiple tasks concurrently.

More generally, understanding how the bias-variance tradeoff relates to the use of shared versus separated representations, and the impact this has on human cognitive function, may facilitate productive cross-influences of research on humans and machines. Understanding factors that influence the bias-variance tradeoff in machine learning (e.g., initialization, subtle forms of regularization) may help guide the formation of hypotheses about whether and how such factors are exploited in the brain and impact cognitive function, including its development. For example, in machine learning it has been recognized that initializing the weights of a network with small random values produces a bias toward the development of shared representations – one that seems neurobiologically plausible and a factor that we exploited in our simulations (e.g. Simulation Studies 6 and 7). These can be thought of starting with a single (albeit uninformative) representation that is segregated into a greater number only under the pressure of the evidence (i.e., learning); that is, it favors the use of the fewer representations shared over more inputs unless "forced" to do otherwise. Conversely, an understanding how the brain manages the bias-variance tradeoff may provide insights into the uniquely adaptive character of human cognition function, that may prove useful in the design of more powerful artificial agents. For example, efforts to understand how the decision is made when to favor the use of shared representations and rely on control-dependent processing for the acquisition of a new concept or skill, versus the investment in automatization to improve the efficiency of performance through the development of separated, task dedicated representations (e.g., Sagiv et al., 2018; Ravi et al., 2020) may inform the effort to design artificial systems that are capable of more sophisticated forms of adaptation, that are both more robust to but can also function more efficiently in a broader range of environments. For example, wouldn't it be nice if a computer had the ability to use flexible, general purpose (e.g., "interpreted") methods to recognize and interact with novel devices, but also develop more efficient, device-dedicated routines ("drivers") for devices with which it continued to interact regularly, and be able to recognize when it was worth it to do so, and to do so on its own?

4.9 Shared Communication Channels

Multitasking capability, as we have considered it here, bears a close relationship to issues that arise in the design of electronic communication systems, that seek to optimize the efficiency of transmission through distributed, parallel communications while avoiding the risks of cross talk introduced by shared communication channels (Alon, Moitra, & Sudakov, 2012; Birk, Linial, & Meshulam, 1993; Chlamtac & Kutten, 1985). Communication channels require balancing channel capacity, i.e. the number of messages that can be simultaneously transmitted between source stations (senders) and destinations stations (receivers), and structural efficiency, i.e. sharing connections between senders and receivers. Shared communication channels are deployed when it is too expensive, or otherwise prohibitive, to build point-to-point communication channels between senders and receivers; as it is the case for the standard computer bus, cellular systems or local area networks (Birk et al., 1993). Thus, analogous to the way in which neural architectures exploit shared representation for the purpose of learning efficiency, shared communication channels rely on shared connectivity in the service of structural efficiency.

Analyses of communication systems may be useful for analyzing and understanding multitasking capability in neural networks, and vice versa. For example, one implementation of shared communication channels is the shared directional multichannel (SDM), which obeys the following protocol: (1) a message transmitted to a sender is broadcast to all receivers connected to the sender, and (2) a message is considered correctly retrieved by a receiver if no other messages are transmitted to the receiver (Birk, 1987). The SDM can be viewed a special case of the bipartite task graph introduced in Part 1 of this article, in which a sender corresponds to a stimulus dimension (input node), a receiver corresponds to a response dimension (output node) and the transmission of a message from a sender to an receiver corresponds to the execution of a task (directed edge). However, unlike in the SDM, stimulus dimensions do not automatically broadcast information to all response dimensions connected to them. Rather, we assume that executing a task requires cognitive control to engage (activate) the stimulus and response dimensions relevant to that task. Thus, the SDM corresponds to the special case of a multitasking agent whose control policy is to engage all stimulus dimensions and all response dimensions simultaneously. Not surprisingly, the capacity of an SDM can be studied by formulating it as a bipartite graph, and it is determined by the largest subset of edges in the graph for in which none of the edges share a node (i.e. no structural dependence), and for which there exists no other edge in the entire graph that connects an input node of an edge in the subset to an output node of a different edge in the subset (i.e. no functional dependence). As for the network architectures considered in our formal analysis, the capacity of an SDM corresponds to the maximum independent set of its dependency graph (Birk et al., 1993). Thus, the tools developed for the study of multitasking capability in neural architectures may inform the design of communication channels, when balancing structural efficiency against channel capacity. Conversely, this work suggests that the analytic tools developed for the study of communication channels continue to provide a promising avenue for the study of human multitasking capacity (for pioneering applications thereof, see Craik, 1948; Welford, 1967; Townsend et al., 1983). For instance, theoretic analyses of parallel processing capability in complex, multi-layered communication channels may be useful for characterizing the multitasking capability of deep (i.e. multi-layered) neural networks.

4.10 Limitations and Future Directions

While we hope that the work presented in this article advances the effort to lend formal rigor and quantitative precision to multiple resource theory, it relied on a number of simplifying assumptions. First, for the graph-theoretic analyses we assumed that representational sharing is a binary factor: either tasks share or don't share representations. In reality, of course, degree of sharing is likely to be a graded factor. We were not able to do address this in the graph theoretic analyses we reported, as it requires the analysis of weighted graphs which is considerably more complex (Alon et al., 2018). Accordingly, the graph-theoretic analyses we presented converted graded degrees of representational overlap in source networks into a discrete graph structure. However, such overlap is an important factor in determining the multitasking capability of a neural system. For example, the simulations in Part I showed that multitasking interference degrades in a graded fashion with the amount of representational overlap between tasks. They also showed that multitasking performance is dependent on other factors, such as the amount of conflict induced by shared representation or persistence of neural activity, both of which are graded effects that scale with the extent of sharing. Therefore, by treating sharing as all or nothing, these effects are not captured by the current form of graph-theoretic analysis methods. For all of these reasons, the further development of those methods to incorporate weighted graphs, that can express graded effects of degree of overlap and temporal dynamics of neural activity, is an important direction for future research.

A simplification in most of the theoretical work presented here, as well as the empirical study, is its focus on tasks that involve simple direct mappings between inputs and outputs. In more realistic scenarios, tasks (such as driving a car) may involve multiple internal (re-)mappings and/or temporally extended sequences of actions. Such tasks are well accommodated by symbol-oriented cognitive architectures that decompose tasks into subtasks, or “chunks” (Meyer & Kieras, 1997a; Salvucci & Taatgen, 2008). Neural network architectures can also accommodate such tasks as a sequence of computations that is carried out over multiple layers, as is the case in recurrent or deep neural networks. This would allow a task to be implemented through multiple paths through the network. However, at the same time the likelihood of interference between pathways implementing different tasks increases with the number of intermediate layers (i.e., opportunities for intersection), as illustrated in Section “Analysis of Multitasking Capability” of this article and in related work (Alon et al., 2017). Further work in this direction, that more fully extends the framework presented here to multi-layer networks may help advance our understanding of multitasking limitations in more complex tasks, beyond the simple stimulus-response mappings of the sort on which we focused here.

The models presented here are, to our knowledge, the first to use a neural network architecture to provide an integrated account of phenomena associated with control-dependent processing and multitasking, that includes the Stroop, PRP, and task switching paradigms, as well as behavioral measures of parallel vs. serial processing channels. However, these represent only a small subset of the wide array of relevant empirical findings that remain to be addressed. We hope that the present work offers insights and approaches that, together with other developments in computational and cognitive neuroscience and machine learning (e.g. Badre et al., 2020; Flesch, Balaguer, Dekker, Nili, & Summerfield, 2018; Graves et al., 2014; A. M. Saxe et al., 2019; A. Saxe, Nelli, & Summerfield, 2020; Townsend & Wenger, 2004), can contribute to the construction of unified models of cognition using neural network architectures, that can approach the scope of those that have been developed using symbol-processing frameworks such as ACT-R and SOAR.

We also hope that the work presented here motivates new, theoretically-guided empirical studies of the neural mechanisms underlying multitasking and skill acquisition. For example, one prediction that derives from this work is that improvements in multitasking should be accompanied by a separation of representations that are responsible for cross-task interference. It should be possible to use the graph theoretic methods presented in Section "Analysis of Multitasking Capability" to analyze brain imaging data in participants trained to multitask color naming and word mapping in the extended Stroop task described in Section "Empirical Study: Learning, Shared Representations and Functional Dependence", to evaluate dependence among the tasks, and how this evolves with multitasking training. The models presented in Simulation Studies 4-5 predict that, initially, representations (measured as patterns of neural activity) should be shared between word mapping and word reading, and that this be associated with functional interference (by way of the effects of word reading on color naming) and poor multitasking performance. However, with training, it is predicted that word mapping representations should diverge from those observed during reading, and that this should be correlated with improvements in multitasking performance. Moreover, improvements in multitasking performance should be influenced by the extent to which training involved incongruent versus congruent trials, with prevalence of the former leading both to greater improvements in performance and separation of word mapping representations. If successful, real-time imaging methods using closed-loop feedback (in which online decoding of neural activity can be used to adapt the training regime) could be used, as they have in other domains (e.g. Iordan, Ritvo, Norman, Turk-Browne, & Cohen, 2020; Stoeckel et al., 2014), to more directly determine the causality of changes in neural representations and performance, as well as feedback-guided training methods that may help augment the acquisition of multitasking capabilities.

4.11 Conclusion

In this work, we presented a formal framework for understanding the constraints associated with control-dependent processing in neural architectures, that suggests these reflect a rational response to the bounds on processing imposed by the use of shared representations, rather than a bound that is intrinsic to the mechanisms responsible for executing control. Analyses carried out within this framework indicate that neural learning systems, whether natural or artificial, are subject to a tension between the use of shared representations that exploit similarity structure between tasks in the service of more effective learning and generalization, but are constrained to serial execution to avoid cross-task interference, versus the use of separated, task-dedicated representations that support concurrent parallelism of execution and thereby efficient processing, but take longer to learn. This computational tradeoff between shared and separated representations can help explain a number of fundamental principles of cognitive function and associated phenomena, and also has applications in machine learning research. Here, we focused on the implications of this tradeoff for control-dependent processing, and argued that limitations thereof reflect a rational choice of the control system to avoid cross-talk between overlapping task pathways. This work helps explain the commonly-observed trajectory from controlled to automatic processing as a rational optimization of the tradeoff between shared and separated representation: a bias toward shared representations affords the flexibility of being able to more rapidly acquire new tasks associated with control-dependent processing, whereas the capacity for automatization through the development of separated representations can be used to configure processing to be more efficient and robust to interference for tasks that require this. This provides a formally rigorous framework for furthering our understanding of how and why people choose to rely on control-dependent processing versus investing in automatization, that may also inform the design of more intelligent artificial agents, that are capable of more sophisticated forms of adaptation and can function over a wider range of task and environments.

Acknowledgements

This work is supported by a grant from the Templeton Foundation. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation. We thank Kayhan Özcimder, Biswadip Dey, Giovanni Petri, Nesreen Ahmed and Theodore L. Willke for their help in carrying out the graph-theoretic analyses presented in this work. Furthermore, we thank Anne Mennen for her support with behavioral data collection. Finally, we thank Gregory Henselman-Petrusek, Giovanni Petri, Tyler Giallanza and Michael Lesnick for helpful feedback and discussions.

References

Abel, T. J., Rhone, A. E., Nourski, K. V., Kawasaki, H., Oya, H., Griffiths, T. D., … Tranel, D. (2015). Direct physiologic evidence of a heteromodal convergence region for proper naming in human left anterior temporal lobe. *Journal of Neuroscience*, *35*(4), 1513–1520.

Agrawal, M., Mattar, M. G., Cohen, J. D., & Daw, N. D. (2020). The temporal dynamics of opportunity costs: A normative account of cognitive fatigue and boredom. *bioRxiv*.

Albers, A. M., Kok, P., Toni, I., Dijkerman, H. C., & De Lange, F. P. (2013). Shared representations for working memory and mental imagery in early visual cortex. *Current Biology*, *23*(15), 1427–1431.

Allport, A. (1980). Attention and performance. *Cognitive psychology: New directions*, *1*, 12–153.

Allport, A., Antonis, B., & Reynolds, P. (1972). On the division of attention: A disproof of the single channel hypothesis. *Quarterly journal of experimental psychology*, *24*(2), 225–235.

Allport, A., Styles, E. A., & Hsieh, S. (1994). Shifting intentional set: Exploring the dynamic control of tasks. In C. Umilta & M. Moscovitch (Eds.), *Conscious and nonconscious information processing: Attention and performance xv* (p. 421-452). Cambridge: MIT Press.

Alon, N., Cohen, J. D., Griffiths, T. L., Manurangsi, P., Reichman, D., Shinkar, I., … Yu, A. (2018). Multitasking capacity: Hardness results and improved constructions. *arXiv preprint arXiv:1809.02835*.

Alon, N., Moitra, A., & Sudakov, B. (2012). Nearly complete graphs decomposable into large induced matchings and their applications. In *Proceedings of the forty-fourth annual acm symposium on theory of computing* (pp. 1079–1090).

Alon, N., Reichman, D., Shinkar, I., Wagner, T., Musslick, S., Cohen, J. D., … Özcimder, K. (2017). A graph-theoretic approach to multitasking. advances in neural information processing systems. In *Advances in Neural Information*

Processing Systems (pp. 2097—2106.). Long Beach, CA.

Altmann, E. M. (2007). Cue-independent task-specific representations in task switching: Evidence from backward inhibition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *33*(5), 892.

Altmann, E. M., & Gray, W. D. (2008). An integrated model of cognitive control in task switching. *Psychological review*, *115*(3), 602.

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological review*, *89*(4), 369.

Anderson, J. R. (1984). Cognitive psychology. *Artificial Intelligence*, *23*(1), 1–11.

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

Anderson, J. R., & Lebiere, C. J. (2014). *The atomic components of thought.* Psychology Press.

Anderson, J. R., Reder, L. M., & Lebiere, C. (1996). Working memory: Activation limitations on retrieval. *Cognitive psychology*, *30*(3), 221–256.

Atkinson, R. C., & Shiffrin, R. M. (1968). Chapter: Human memory: A proposed system and its control processes. *The psychology of learning and motivation*, *2*, 89–195.

Badre, D., Bhandari, A., Keglovits, H., & Kikumoto, A. (2020). The dimensionality of neural representations for control.

Badre, D., & Wagner, A. D. (2007). Left ventrolateral prefrontal cortex and the cognitive control of memory. *Neuropsychologia*, *45*(13), 2883–2901.

Barlow, H. B. (1972). Single units and sensation: a neuron doctrine for perceptual psychology? *Perception*, *1*(4), 371–394.

Baxter, J. (1995). Learning internal representations. In *Proceedings of the eighth annual conference on computational learning theory* (pp. 311–320).

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In

Proceedings of the 26th annual international conference on machine learning (pp. 41–48).

Berman, P., & Fürer, M. (1994). Approximating maximum independent set in bounded degree graphs. In *Soda* (Vol. 94, pp. 365–371).

Bernardi, S., Benna, M. K., Rigotti, M., Munuera, J., Fusi, S., & Salzman, D. (2018). The geometry of abstraction in hippocampus and prefrontal cortex. *bioRxiv*, 408633.

Bier, B., de Boysson, C., & Belleville, S. (2014). Identifying training modalities to improve multitasking in older adults. *Age*, *36*(4), 9688.

Birk, Y. (1987). *Concurrent communication among multi-transceiver stations over shared media* (Tech. Rep.). STANFORD UNIV CA COMPUTER SYSTEMS LAB.

Birk, Y., Linial, N., & Meshulam, R. (1993). On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Transactions on Information Theory*, *39*(1), 186–191.

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological review*, *113*(4), 700.

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological review*, *108*(3), 624.

Bouchacourt, F., & Buschman, T. J. (2019). A flexible model of working memory. *Neuron*, *103*(1), 147–160.

Brass, M., Ruge, H., Meiran, N., Rubin, O., Koch, I., Zysset, S., … von Cramon, D. Y. (2003). When the same response has different meanings:: recoding the response meaning in the lateral prefrontal cortex. *Neuroimage*, *20*(2), 1026–1031.

Braver, T. S., Barch, D. M., & Cohen, J. D. (1999). Cognition and control in schizophrenia: a computational model of dopamine and prefrontal function. *Biological psychiatry*, *46*(3), 312–328.

Briggs, G. E., Peters, G. L., & Fisher, R. P. (1972). On the locus of the divided-attention effects. *Perception & Psychophysics*, *11*(4), 315–320.

Broadbent, D. E. (1957). A mechanical model for human attention and immediate memory. *Psychological review*, *64*(3), 205.

Broadbent, D. E. (1958). *Perception and communication. elmsford, ny, us.* Pergamon Press. http://dx. doi. org/10.1037/10037-000.

Brown, J. W., Reynolds, J. R., & Braver, T. S. (2007). A computational model of fractionated conflict-control mechanisms in task-switching. *Cognitive psychology*, *55*(1), 37–85.

Brumby, D. P., Howes, A., & Salvucci, D. D. (2007). A cognitive constraint model of dual-task trade-offs in a highly dynamic driving task. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 233–242).

Brumby, D. P., Salvucci, D. D., & Howes, A. (2009). Focus on driving: How cognitive constraints shape the adaptation of strategy when dialing while driving. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1629–1638).

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review*, *108*(4), 847.

Callaway, F., Lieder, F., Das, P., Gul, S., Krueger, P. M., & Griffiths, T. (2018). A resource-rational analysis of human planning. In *Cogsci.*

Cameron, K. (1989). Induced matchings. *Discrete Applied Mathematics*, *24*(1-3), 97–102.

Carlson, T. A., Simmons, R. A., Kriegeskorte, N., & Slevc, L. R. (2014). The emergence of semantic meaning in the ventral temporal pathway. *Journal of cognitive neuroscience*, *26*(1), 120–131.

Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41–75.

Cave, K. R., & Wolfe, J. M. (1990). Modeling the role of parallel processing in visual search. *Cognitive psychology*, *22*(2), 225–271.

Chang, M. B., Gupta, A., Levine, S., & Griffiths, T. L. (2018). Automatically composing representation transformations as a means for generalization. *arXiv* *preprint arXiv:1807.04640*.

Chein, J. M., & Schneider, W. (2012). The brain's learning and control architecture. *Current Directions in Psychological Science*, *21*(2), 78–84.

Chen, L., & Rogers, T. T. (2010). Nonverbal semantic processing disrupts visual word recognition in healthy adults. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 32).

Chlamtac, I., & Kutten, S. (1985). On broadcasting in radio networks-problem analysis and protocol design. *IEEE Transactions on Communications*, *33*(12), 1240–1246.

Chung, S., Lee, D. D., & Sompolinsky, H. (2018, jul). Classification and Geometry of General Perceptual Manifolds. *Physical Review X*, *8*(3), 031003. doi: 10.1103/PhysRevX.8.031003

Cohen, J. D. (2017). Cognitive control: core constructs and current considerations. *The Wiley handbook of cognitive control*, 1–28.

Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychological review*, *97*(3), 332.

Cohen, U., Chung, S., Lee, D. D., & Sompolinsky, H. (2019). Separability and geometry of object manifolds in deep neural networks. *bioRxiv*, 644658.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).

Colonius, H., & Vorberg, D. (1994). Distribution inequalities for parallel models with unlimited capacity. *Journal of Mathematical Psychology*, *38*(1), 35–58.

Connolly, A. C., Gobbini, M. I., & Haxby, J. V. (2012). Three virtues of similaritybased multivariate pattern analysis: An example from the human object vision pathway. *Understanding visual population codes: Toward a common multivariate framework for cell recording and functional imaging*, 335–55.

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.

Cowan, N. (2010). The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science*, *19*(1), 51–57.

Cowan, N., Rouder, J. N., Blume, C. L., & Saults, J. S. (2012). Models of verbal working memory capacity: What does it take to make them work? *Psychological review*, *119*(3), 480.

Craik, K. J. (1948). Theory of the human operator in control systems. ii. man as an element in a control system. *British journal of psychology*, *38*(3), 142.

Curtis, C. E., & Lee, D. (2010). Beyond working memory: the role of persistent activity in decision making. *Trends in cognitive sciences*, *14*(5), 216–222.

Davis, R. (1959). The role of "attention" in the psychological refractory period. *Quarterly Journal of Experimental Psychology*, *11*(4), 211–220.

Debaere, F., Wenderoth, N., Sunaert, S., Van Hecke, P., & Swinnen, S. (2004). Changes in brain activation during the acquisition of a new bimanual coordination task. *Neuropsychologia*, *42*(7), 855–867.

Decety, J., & Sommerville, J. A. (2003). Shared representations between self and other: a social cognitive neuroscience view. *Trends in cognitive sciences*, *7*(12), 527–533.

De Jong, R. (1993). Multiple bottlenecks in overlapping task performance. *Journal of Experimental Psychology: Human Perception and Performance*, *19*(5), 965.

Deng, L., Hinton, G., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 ieee international conference on acoustics, speech and signal processing* (pp. 8599–8603).

Desimone, R. (1991). Face-selective cells in the temporal cortex of monkeys. *Journal of cognitive neuroscience*, *3*(1), 1–8.

Diamond, A. (2013). Executive functions. *Annual review of psychology*, *64*, 135–168.

Diestel, R. (2005). *Graph theory (graduate texts in mathematics)*. Springer. Hardcover. Retrieved from `http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20{\&}path=ASIN/3540261826`

Duong, L., Cohn, T., Bird, S., & Cook, P. (2015). Low resource dependency parsing:

Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)* (pp. 845–850).

Durstewitz, D., & Seamans, J. K. (2008). The dual-state theory of prefrontal cortex dopamine function with relevance to catechol-o-methyltransferase genotypes and schizophrenia. *Biological psychiatry*, *64*(9), 739–749.

Dux, P. E., Tombu, M. N., Harrison, S., Rogers, B. P., Tong, F., & Marois, R. (2009). Training improves multitasking performance by increasing the speed of information processing in human prefrontal cortex. *Neuron*, *63*(1), 127–138.

Eidels, A., Townsend, J. T., & Algom, D. (2010). Comparing perception of stroop stimuli in focused versus divided attention paradigms: Evidence for dramatic processing differences. *Cognition*, *114*(2), 129–150.

Ellenbogen, R., & Meiran, N. (2008). Working memory involvement in dual-task performance: Evidence from the backward compatibility effect. *Memory & Cognition*, *36*(5), 968–978.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*(1), 71–99.

Engle, W., Kane, J., & Tuholski, S. (1999). *Models of working memory (eds. myake, a. & shah, p.) 102–134.* Cambridge University Press, Cambridge.

Fagot, C. A. (1995). *Chronometric investigations of task switching.* (Unpublished doctoral dissertation). ProQuest Information & Learning.

Feng, S. F., Schwemmer, M., Gershman, S. J., & Cohen, J. D. (2014). Multitasking versus multiplexing: Toward a normative account of limitations in the simultaneous execution of control-demanding behaviors. *Cognitive, Affective, & Behavioral Neuroscience*, *14*(1), 129–146.

Fifić, M., Townsend, J. T., & Eidels, A. (2008). Studying visual search using systems factorial methodology with target—distractor similarity as the factor. *Perception*

& Psychophysics, *70*(4), 583–603.

Fischer, R., Gottschalk, C., & Dreisbach, G. (2014). Context-sensitive adjustment of cognitive control in dual-task performance. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *40*(2), 399.

Fischer, R., & Plessow, F. (2015). Efficient multitasking: parallel versus serial processing of multiple tasks. *Frontiers in psychology*, *6*, 1366.

Flesch, T., Balaguer, J., Dekker, R., Nili, H., & Summerfield, C. (2018). Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences*, *115*(44), E10313–E10322.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, *3*(4), 128–135.

Garner, K., & Dux, P. E. (2015). Training conquers multitasking costs by dividing task representations in the frontoparietal-subcortical system. *Proceedings of the National Academy of Sciences*, *112*(46), 14372–14377.

Gershman, S. J., Horvitz, E. J., & Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, *349*(6245), 273–278.

Gigerenzer, G. (2008). Why heuristics work. *Perspectives on psychological science*, *3*(1), 20–29.

Gilbert, S. J., & Shallice, T. (2002). Task switching: A pdp model. *Cognitive psychology*, *44*(3), 297–337.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the ieee international conference on computer vision* (pp. 1440–1448).

Glaser, M. O., & Glaser, W. R. (1982). Time course analysis of the stroop phenomenon. *Journal of Experimental Psychology: Human Perception and Performance*, *8*(6), 875.

Glucksberg, S. (1963). Rotary pursuit tracking with divided attention to cutaneous, visual and auditory signals. *Journal of engineering psychology*.

Godsil, C., & Royle, G. (2001). Graduate texts in mathematics. *Algebraic graph theory*, *207*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT press.

Gopher, D., Brickner, M., & Navon, D. (1982). Different difficulty manipulations interact differently with task emphasis: Evidence for multiple resources. *Journal of Experimental Psychology: Human Perception and Performance*, *8*(1), 146.

Goschke, T. (2000). Intentional reconfiguration and j-ti involuntary persistence in task set switching. *Control of cognitive processes: Attention and performance XVIII*, *18*, 331.

Göthe, K., Oberauer, K., & Kliegl, R. (2016). Eliminating dual-task costs by minimizing crosstalk between tasks: The role of modality and feature pairings. *Cognition*, *150*, 92–108.

Grange, J. A., & Houghton, G. (2014). Models of cognitive control in task switching. *Task switching and cognitive control*, 160–199.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Greenwald, A. G. (1970). Sensory feedback mechanisms in performance control: With special reference to the ideo-motor mechanism. *Psychological review*, *77*(2), 73.

Greenwald, A. G., & Shulman, H. G. (1973). On doing two things at once: Ii. elimination of the psychological refractory period effect. *Journal of experimental psychology*, *101*(1), 70.

Grice, G. R., Canham, L., & Boroughs, J. M. (1984). Combination rule for redundant information in reaction time tasks with divided attention. *Perception & Psychophysics*, *35*(5), 451–463.

Grice, G. R., Canham, L., & Gwynne, J. W. (1984). Absence of a redundant-signals effect in a reaction time task with divided attention. *Perception & Psychophysics*, *36*(6), 565–570.

Griffiths, T. L., Lieder, F., & Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in cognitive science*, *7*(2), 217–229.

Griffiths, T. L., & Tenenbaum, J. B. (2006). Optimal predictions in everyday cognition. *Psychological science*, *17*(9), 767–773.

Halvorson, K. M., Ebner, H., & Hazeltine, E. (2013). Investigating perfect timesharing: The relationship between im-compatible tasks and dual-task performance. *Journal of Experimental Psychology: Human Perception and Performance*, *39*(2), 413.

Hasson, U., Chen, J., & Honey, C. J. (2015). Hierarchical process memory: memory as an integral component of information processing. *Trends in cognitive sciences*, *19*(6), 304–313.

Haussler, D. (1992). Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, *100*(1), 78–150.

Hazeltine, E., Ruthruff, E., & Remington, R. W. (2006). The role of input and output modality pairings in dual-task performance: Evidence for content-dependent central interference. *Cognitive Psychology*, *52*(4), 291–345.

Hazeltine, E., Teague, D., & Ivry, R. B. (2002). Simultaneous dual-task performance reveals parallel response selection after practice. *Journal of Experimental Psychology: Human Perception and Performance*, *28*(3), 527.

Henselman-Petrusek, G., Segert, S., Keller, B., Tepper, M., & Cohen, J. D. (2019). Geometry of shared representations. In *Conference on Cognitive Computational Neuroscience.*

Herculano-Houzel, S. (2009). The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience*, *3*, 31.

Herd, S. A., Banich, M. T., & O'Reilly, R. C. (2006). Neural mechanisms of cognitive control: An integrative model of stroop task performance and fmri data. *Journal of cognitive neuroscience*, *18*(1), 22–32.

Herd, S. A., Hazy, T. E., Chatham, C. H., Brant, A. M., Friedman, N. P., et al. (2014). A neural network model of individual differences in task switching abilities. *Neuropsychologia*, *62*, 375–389.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., & Lerchner, A. (2018). Towards a definition of disentangled representations. *arXiv preprint* *arXiv:1812.02230*.

Hinton, G. E., et al. (1986). Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society* (Vol. 1, p. 12).

Hirst, W., & Kalmar, D. (1987). Characterizing attentional resources. *Journal of Experimental Psychology: General*, *116*(1), 68.

Hirst, W., Spelke, E. S., Reaves, C. C., Caharack, G., & Neisser, U. (1980). Dividing attention without alternation or automaticity. *Journal of Experimental Psychology: General*, *109*(1), 98.

Hommel, B. (1998). Automatic stimulus–response translation in dual-task performance. *Journal of Experimental Psychology: Human Perception and Performance*, *24*(5), 1368.

Hopcroft, J. E., & Karp, R. M. (1973). An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, *2*(4), 225–231.

Iordan, M. C., Ritvo, V. J., Norman, K. A., Turk-Browne, N. B., & Cohen, J. D. (2020). Sculpting new visual concepts into the human brain. *bioRxiv*.

Jackson, R. L., Rogers, T. T., & Ralph, M. A. L. (2019). Reverse-engineering the cortical architecture for controlled semantic cognition. *bioRxiv*, 860528.

Jensen, A. R. (1988). Speed of information processing and population differences. *Human abilities in cultural context*, 105–145.

Jersild, A. T. (1927). Mental set and shift. *Archives of psychology*.

Jiang, Y., Saxe, R., & Kanwisher, N. (2004). Functional magnetic resonance imaging provides new constraints on theories of the psychological refractory period. *Psychological Science*, *15*(6), 390–396.

Johnston, W. A., Greenberg, S. N., Fisher, R. P., & Martin, D. W. (1970). Divided attention: A vehicle for monitoring memory processes. *Journal of Experimental Psychology*, *83*(1p1), 164.

Jonides, J. (2004). How does practice makes perfect? *Nature neuroscience*, *7*(1), 10–11.

Kahneman, D. (1973). *Attention and effort* (Vol. 1063). Citeseer.

Kahneman, D., & Tversky, A. (1972). Subjective probability: A judgment of representativeness. *Cognitive psychology*, *3*(3), 430–454.

Kajić, I., Gosmann, J., Stewart, T. C., Wennekers, T., & Eliasmith, C. (2017). A spiking neuron model of word associations for the remote associates test. *Frontiers in psychology*, *8*, 99.

Kalanthroff, E., Davelaar, E. J., Henik, A., Goldfarb, L., & Usher, M. (2018). Task conflict and proactive control: A computational theory of the stroop task. *Psychological review*, *125*(1), 59.

Kantowitz, B. H., & Knight, J. L. (1974). Testing tapping time-sharing. *Journal of Experimental Psychology*, *103*(2), 331.

Kantowitz, B. H., & Knight Jr, J. L. (1976). Testing tapping timesharing, ii: Auditory secondary task. *Acta Psychologica*, *40*(5), 343–362.

Karlin, L., & Kestenbaum, R. (1968). Effects of number of alternatives on the psychological refractory period. *Quarterly Journal of Experimental Psychology*, *20*(2), 167–178.

Keele, S. W. (1973). *Attention and human performance*. Goodyear Publishing Company.

Kelly, A. C., & Garavan, H. (2005). Human functional neuroimaging of brain changes associated with practice. *Cerebral cortex*, *15*(8), 1089–1102.

Kerr, B. (1973). Processing demands during mental operations. *Memory & Cognition*, *1*(4), 401–412.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Human–Computer Interaction*, *12*(4), 391–438.

Kieras, D. E., Meyer, D. E., Ballas, J. A., & Lauber, E. J. (2000). Modern computational perspectives on executive mental processes and cognitive control: Where to from here. *Control of cognitive processes: Attention and performance XVIII*, 681–712.

Kiesel, A., Steinhauser, M., Wendt, M., Falkenstein, M., Jost, K., Philipp, A. M., &

Koch, I. (2010). Control and interference in task switching—a review. *Psychological bulletin*, *136*(5), 849.

Kinsbourne, M., & Hicks, R. E. (1978). Functional cerebral space: A model for overflow, transfer and interference effects in human performance. *Attention and performance VII*, 345–362.

Koch, I., Poljac, E., Müller, H., & Kiesel, A. (2018). Cognitive structure, flexibility, and plasticity in human multitasking—an integrative review of dual-task and task-switching research. *Psychological bulletin*, *144*(6), 557.

Kool, W., & Botvinick, M. (2018). Mental labour. *Nature human behaviour*, *2*(12), 899–908.

Kool, W., Gershman, S. J., & Cushman, F. A. (2017). Cost-benefit arbitration between multiple reinforcement-learning systems. *Psychological science*, *28*(9), 1321–1333.

Kool, W., McGuire, J. T., Rosen, Z. B., & Botvinick, M. M. (2010). Decision making and the avoidance of cognitive demand. *Journal of Experimental Psychology: General*, *139*(4), 665.

Kosslyn, S. M., Pascual-Leone, A., Felician, O., Camposano, S., Keenan, J. P., Ganis, G., . . . others (1999). The role of area 17 in visual imagery: convergent evidence from pet and rtms. *Science*, *284*(5411), 167–170.

Kramer, A. F., Larish, J. F., & Strayer, D. L. (1995). Training for attentional control in dual task settings: a comparison of young and old adults. *Journal of experimental psychology: Applied*, *1*(1), 50.

Kriegeskorte, N., & Kievit, R. A. (2013). Representational geometry: integrating cognition, computation, and the brain. *Trends in cognitive sciences*, *17*(8), 401–412.

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, *2*, 4.

Kriete, T., Noelle, D. C., Cohen, J. D., & O'Reilly, R. C. (2013). Indirection and symbol-like processing in the prefrontal cortex and basal ganglia. *Proceedings of* *the National Academy of Sciences*, *110*(41), 16390–16395.

Krueger, K. A., & Dayan, P. (2009). Flexible shaping: How learning in small steps helps. *Cognition*, *110*(3), 380–394.

Kurzban, R., Duckworth, A., Kable, J. W., & Myers, J. (2013). An opportunity cost model of subjective effort and task performance. *Behavioral and brain sciences*, *36*(6), 661–679.

Laird, J. E. (2012). *The soar cognitive architecture.* MIT press.

Lake, B. M. (2019). Compositional generalization through meta sequence-to-sequence learning. In *Advances in neural information processing systems* (pp. 9791–9801).

Lavie, N., Hirst, A., De Fockert, J. W., & Viding, E. (2004). Load theory of selective attention and cognitive control. *Journal of Experimental Psychology: General*, *133*(3), 339.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

LeCun, Y., et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, *19*, 143–155.

Lerner, Y., Honey, C. J., Silbert, L. J., & Hasson, U. (2011). Topographic mapping of a hierarchy of temporal receptive windows using a narrated story. *Journal of Neuroscience*, *31*(8), 2906–2915.

Lesnick, M., Musslick, S., Dey, B., & Cohen, J. D. (2020). A formal framework for cognitive models of multitasking.
doi: https://doi.org/10.31234/osf.io/7yzdn

Lewis, R. L., Howes, A., & Singh, S. (2014). Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in cognitive science*, *6*(2), 279–311.

Lieder, F., & Griffiths, T. L. (2015). When to use which heuristic: A rational solution to the strategy selection problem. In *Cogsci.*

Lieder, F., Shenhav, A., Musslick, S., & Griffiths, T. L. (2018). Rational metareasoning and the plasticity of cognitive control. *PLoS computational biology*, *14*(4), e1006043.

Lien, M.-C., & Proctor, R. W. (2002). Stimulus-response compatibility and psychological refractory period effects: Implications for response selection. *Psychonomic bulletin & review*, *9*(2), 212–238.

Liepelt, R., Fischer, R., Frensch, P. A., & Schubert, T. (2011). Practice-related reduction of dual-task costs under conditions of a manual-pedal response combination. *Journal of Cognitive Psychology*, *23*(1), 29–44.

Lindsay, D. S., & Jacoby, L. L. (1994). Stroop process dissociations: The relationship between facilitation and interference. *Journal of Experimental Psychology: Human Perception and Performance*, *20*(2), 219.

Lindsay, P., Taylor, M., & Forbes, S. (1968). Attention and multidimensional discrimination1. *Perception & Psychophysics*, *4*(2), 113–117.

Linnainmaa, S. (1970). The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. *Master's Thesis (in Finnish), Univ. Helsinki*, 6–7.

Logan, G. D. (1978). Attention in character-classification tasks: Evidence for the automaticity of component stages. *Journal of Experimental Psychology: General*, *107*(1), 32.

Logan, G. D. (1980). Attention and automaticity in stroop and priming tasks: Theory and data. *Cognitive psychology*, *12*(4), 523–553.

Logan, G. D. (1985). Executive control of thought and action. *Acta Psychologica*, *60*(2-3), 193–210.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological review*, *95*(4), 492.

Logan, G. D., & Bundesen, C. (2003). Clever homunculus: Is there an endogenous act of control in the explicit task-cuing procedure? *Journal of Experimental Psychology: Human Perception and Performance*, *29*(3), 575.

Logan, G. D., & Burkell, J. (1986). Dependence and independence in responding to double stimulation: A comparison of stop, change, and dual-task paradigms.

Journal of Experimental Psychology: Human Perception and Performance, *12*(4), 549.

Logan, G. D., & Gordon, R. D. (2001). Executive control of visual attention in dual-task situations. *Psychological review*, *108*(2), 393.

Logan, G. D., & Schneider, D. W. (2006). Priming or executive control? associative priming of cue encoding increases “switch costs” in the explicit task-cuing procedure. *Memory & Cognition*, *34*(6), 1250–1259.

Logan, G. D., & Schulkind, M. D. (2000). Parallel memory retrieval in dual-task situations: I. semantic memory. *Journal of Experimental Psychology: Human Perception and Performance*, *26*(3), 1072.

Long, M., & Wang, J. (2015). Learning multiple tasks with deep relationship networks. *arXiv preprint arXiv:1506.02117*, *2*, 1.

Lu, X., Li, X., & Mou, L. (2014). Semi-supervised multitask learning for scene recognition. *IEEE transactions on cybernetics*, *45*(9), 1967–1976.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, *390*(6657), 279–281.

Ma, W. J., & Huang, W. (2009). No capacity limit in attentional tracking: Evidence for probabilistic inference under a resource constraint. *Journal of Vision*, *9*(11), 3–3.

Ma, W. J., Husain, M., & Bays, P. M. (2014). Changing concepts of working memory. *Nature neuroscience*, *17*(3), 347.

Macleod, C. M. (1998). Training on integrated versus separated stroop tasks: The progression of interference and facilitation. *Memory & Cognition*, *26*(2), 201–211.

MacLeod, C. M., & Dunbar, K. (1988). Training and stroop-like interference: Evidence for a continuum of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*(1), 126.

Major, G., & Tank, D. (2004). Persistent neural activity: prevalence and mechanisms. *Current opinion in neurobiology*, *14*(6), 675–684.

Mandler, J. M., Bauer, P. J., & McDonough, L. (1991). Separating the sheep from the goats: Differentiating global categories. *Cognitive Psychology*, *23*(2), 263–298.

Marill, T. (1957). Psychological refractory phase. *British Journal of Psychology*, *48*(2), 93–97.

Mayr, U., & Keele, S. W. (2000). Changing internal constraints on action: The role of backward inhibition. *Journal of Experimental Psychology: General*, *129*(1), 4.

Mayr, U., & Kliegl, R. (2000). Task-set switching and long-term memory retrieval.

Mazurek, M. E., Roitman, J. D., Ditterich, J., & Shadlen, M. N. (2003). A role for neural integrators in perceptual decision making. *Cerebral cortex*, *13*(11), 1257–1269.

McClelland, J. L. (1979). On the time relations of mental processes: an examination of systems of processes in cascade. *Psychological review*, *86*(4), 287.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, *102*(3), 419.

McClelland, J. L., & Rogers, T. T. (2003). The parallel distributed processing approach to semantic cognition. *Nature reviews neuroscience*, *4*(4), 310–322.

McClelland, J. L., Rumelhart, D. E., Group, P. R., et al. (1986). Parallel distributed processing. *Explorations in the Microstructure of Cognition*, *2*, 216–271.

McClelland, J. L., St. John, M., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. *Language and cognitive processes*, *4*(3-4), SI287–SI335.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation* (Vol. 24, pp. 109–165). Elsevier.

McCracken, J., & Aldrich, T. (1984). *Analyses of selected lhx mission functions: Implications for operator workload and system automation goals* (Tech. Rep.). ANACAPA SCIENCES INC FORT RUCKER AL.

McLeod, P. (1977). Parallel processing and the psychological refractory period. *Acta Psychologica*, *41*(5), 381–396.

McLeod, P., Driver, J., & Crisp, J. (1988). Visual search for a conjunction of movement and form is parallel. *Nature*, *332*(6160), 154–155.

Mednick, S. (1962). The associative basis of the creative process. *Psychological review*, *69*(3), 220.

Meiran, N. (1996). Reconfiguration of processing mode prior to task performance. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(6), 1423.

Meiran, N., Chorev, Z., & Sapir, A. (2000). Component processes in task switching. *Cognitive psychology*, *41*(3), 211–253.

Meyer, D. E., & Kieras, D. E. (1997a). A computational theory of executive cognitive processes and multiple-task performance: Part 2. accounts of psychological refractory-period phenomena. *Psychological review*, *104*(4), 749.

Meyer, D. E., & Kieras, D. E. (1997b). A computational theory of executive cognitive processes and multiple-task performance: Part i. basic mechanisms. *Psychological review*, *104*(1), 3.

Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, *24*(1), 167–202.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, *63*(2), 81.

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). Plans and the structure of behavior.

Miller, J. (1982). Divided attention: Evidence for coactivation with redundant signals. *Cognitive psychology*, *14*(2), 247–279.

Miyake, A., & Friedman, N. P. (1998). Individual differences in second language proficiency: Working memory as language aptitude. *Foreign language learning: Psycholinguistic studies on training and retention*, 339–364.

Morton, J., & Chambers, S. M. (1973). Selective attention to words and colours. *The Quarterly Journal of Experimental Psychology*, *25*(3), 387–397.

Münte, T. F., Altenmüller, E., & Jäncke, L. (2002). The musician's brain as a model of neuroplasticity. *Nature Reviews Neuroscience*, *3*(6), 473–478.

Musslick, S., Bizyaeva, A., Agaron, S., Naomi, E. L., & Cohen, J. D. (2019). Stability-flexibility dilemma in cognitive control: A dynamical system perspective. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society* (pp. 2420—2426). Montreal, CA.

Musslick, S., Dey, B., Özcimder, K., Patwary, M., Willke, T. L., & Cohen, J. D. (2016). Controlled vs. automatic processing: A graph-theoretic approach to the analysis of serial vs. parallel processing in neural network architectures. In *Proceedings of the 38th Annual Meeting of the Cognitive Science Society* (pp. 1547—1552). Philadelphia, PA.

Musslick, S., Jang, J. S., Shvartsman, M., Shenhav, A., & Cohen, J. D. (2018). Constraints associated with cognitive control and the stability-flexibility dilemma. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society* (pp. 806—811). Madison, WI.

Musslick, S., Saxe, A., Özcimder, K., Dey, B., Henselman, G., & Cohen, J. D. (2017). Multitasking capability versus learning efficiency in neural network architectures. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society* (pp. 829—834). London, UK.

Musslick, S., Shenhav, A., Botvinick, M. M., & Cohen, J. D. (2015). A computational model of control allocation based on the expected value of control. In *Reinforcement Learning and Decision Making Conference 2015.*

Nairne, J. S. (1990). A feature model of immediate memory. *Memory & Cognition*, *18*(3), 251–269.

Navon, D., & Gopher, D. (1979). On the economy of the human-processing system. *Psychological review*, *86*(3), 214.

Navon, D., & Miller, J. (1987). Role of outcome conflict in dual-task interference. *Journal of Experimental Psychology: Human Perception and Performance*, *13*(3), 435.

Navon, D., & Miller, J. (2002). Queuing or sharing? a critical evaluation of the single-bottleneck notion. *Cognitive psychology*, *44*(3), 193–251.

Newell, A., & Rosenbloom, S. (1981). and the law of practice. *Cognitive skills and their acquisition*.

Nijboer, M., Borst, J., van Rijn, H., & Taatgen, N. (2014). Single-task fmri overlap predicts concurrent multitasking interference. *NeuroImage*, *100*, 60–74.

Norman, D. A., & Bobrow, D. G. (1975). On data-limited and resource-limited processes. *Cognitive psychology*, *7*(1), 44–64.

Norman, D. A., & Shallice, T. (1986). Attention to action. In *Consciousness and self-regulation* (pp. 1–18). Springer.

North, R. A. (1977). *Task components and demands as factors in dual-task performance.* (Tech. Rep.). ILLINOIS UNIV AT URBANA-CHAMPAIGN SAVOY AVIATION RESEARCH LAB.

Notebaert, W., Gevers, W., Verguts, T., & Fias, W. (2006). Shared spatial representations for numbers and space: the reversal of the snarc and the simon effects. *Journal of Experimental Psychology: Human Perception and Performance*, *32*(5), 1197.

Oberauer, K., Farrell, S., Jarrold, C., & Lewandowsky, S. (2016). What limits working memory capacity? *Psychological Bulletin*, *142*(7), 758.

Oberauer, K., & Kliegl, R. (2006). A formal model of capacity limits in working memory. *Journal of memory and language*, *55*(4), 601–626.

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.

Otto, A. R., Skatova, A., Madlon-Kay, S., & Daw, N. D. (2014). Cognitive control predicts use of model-based reinforcement learning. *Journal of cognitive neuroscience*, *27*(2), 319–333.

Page, M., & Norris, D. (1998). The primacy model: a new model of immediate serial recall. *Psychological review*, *105*(4), 761.

Palmer, J. (1990). Attentional limits on the perception and memory of visual information. *Journal of Experimental Psychology: Human Perception and Performance*, *16*(2), 332.

Pashler, H. (1984). Processing stages in overlapping tasks: evidence for a central bottleneck. *Journal of Experimental Psychology: Human Perception and Performance*, *10*(3), 358.

Pashler, H. (1994). Dual-task interference in simple tasks: data and theory. *Psychological bulletin*, *116*(2), 220.

Pashler, H., & Sutherland, S. (1998). *The psychology of attention (vol. 15).* Cambridge, MA: MIT press.

Pauen, S. (2002). Evidence for knowledge–based category discrimination in infancy. *Child Development*, *73*(4), 1016–1033.

Pelvig, D. P., Pakkenberg, H., Stark, A. K., & Pakkenberg, B. (2008). Neocortical glial cell numbers in human brains. *Neurobiology of aging*, *29*(11), 1754–1762.

Petersen, S. E., Van Mier, H., Fiez, J. A., & Raichle, M. E. (1998). The effects of practice on the functional anatomy of task performance. *Proceedings of the National Academy of Sciences*, *95*(3), 853–860.

Petri, G., Musslick, S., Öczimder, K., Dey, B., Ahmed, N., Willke, T., & Cohen, J. D. (2020). Universal limits to parallel processing capability of network architectures. *arXiv*, 1708.03263.

Plaut, D. C., & Booth, J. R. (2000). Individual and developmental differences in semantic priming: empirical and computational support for a single-mechanism account of lexical processing. *Psychological review*, *107*(4), 786.

Poldrack, R. A. (2000). Imaging brain plasticity: conceptual and methodological issues—a theoretical review. *Neuroimage*, *12*(1), 1–13.

Posner, M. I., & Snyder, C. (1975). *Attention and cognitive control. information processing and cognition: The loyola symposium.* Hillsdale NJ: Erlbaum.

Quinn, P. C., & Johnson, M. H. (1997). The emergence of perceptual category representations in young infants: A connectionist analysis. *Journal of experimental child psychology*, *66*(2), 236–263.

Ralph, M. A. L., Jefferies, E., Patterson, K., & Rogers, T. T. (2017). The neural and computational bases of semantic cognition. *Nature Reviews Neuroscience*, *18*(1), 42.

Ratcliff, R., & Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological science*, *9*(5), 347–356.

Ravi, S., Musslick, S., Hamin, M., Willke, T., & Cohen, J. D. (2020). Navigating the tradeoff between multi-task learning and learning to multitask in deep neural networks. *arXiv*, 2007.10527.

Ridderinkhof, K. R., Van Den Wildenberg, W. P., Segalowitz, S. J., & Carter, C. S. (2004). Neurocognitive mechanisms of cognitive control: the role of prefrontal cortex in action selection, response inhibition, performance monitoring, and reward-based learning. *Brain and cognition*, *56*(2), 129–140.

Riesenhuber, M., & Poggio, T. (1999). Are cortical models really bound by the "binding problem"? *Neuron*, *24*(1), 87–93.

Rioult-Pedotti, M.-S., Friedman, D., & Donoghue, J. P. (2000). Learning-induced ltp in neocortex. *Science*, *290*(5491), 533–536.

Ritter, S., Wang, J. X., Kurth-Nelson, Z., Jayakumar, S. M., Blundell, C., Pascanu, R., & Botvinick, M. (2018). Been there, done that: Meta-learning with episodic recall. *arXiv preprint arXiv:1805.09692*.

Roelofs, A. (2003). Goal-referenced selection of verbal action: modeling attentional control in the stroop task. *Psychological review*, *110*(1), 88.

Rogers, R. D., & Monsell, S. (1995). Costs of a predictible switch between simple cognitive tasks. *Journal of experimental psychology: General*, *124*(2), 207.

Rogers, T. T., & McClelland, J. L. (2004). *Semantic cognition: A parallel distributed processing approach*. MIT press.

Rohde, D. L., & Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, *72*(1), 67–109.

Rolls, E. T., & Tovee, M. J. (1995). The responses of single neurons in the temporal visual cortical areas of the macaque when more than one stimulus is present in the receptive field. *Experimental Brain Research*, *103*(3), 409–420.

Rosenbloom, P. S., Laird, J., & Newell, A. (1993). The soar papers: Research on integrated intelligence.

Rubinov, M., & Sporns, O. (2010). Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, *52*(3), 1059–1069.

Rubinstein, J. S., Meyer, D. E., & Evans, J. E. (2001). Executive control of cognitive processes in task switching. *Journal of experimental psychology: human perception and performance*, *27*(4), 763.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Rumelhart, D. E., Todd, P. M., et al. (1993). Learning and connectionist representations. *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience*, *2*, 3–30.

Russell, S. J., & Subramanian, D. (1994). Provably bounded-optimal agents. *Journal of Artificial Intelligence Research*, *2*, 575–609.

Ruthruff, E., Johnston, J. C., Van Selst, M., Whitsell, S., & Remington, R. (2003). Vanishing dual-task interference after practice: Has the bottleneck been eliminated or is it merely latent? *Journal of Experimental Psychology: Human Perception and Performance*, *29*(2), 280.

Ruthruff, E., Van Selst, M., Johnston, J. C., & Remington, R. (2006). How does practice reduce dual-task interference: Integration, automatization, or just stage-shortening? *Psychological research*, *70*(2), 125–142.

Sagiv, Y., Musslick, S., Niv, Y., & Cohen, J. D. (2018). Efficiency of learning vs. processing: Towards a normative theory of multitasking. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society* (pp. 1004—1009). Madison, WI.

Sakai, K., Hikosaka, O., Miyauchi, S., Takino, R., Sasaki, Y., & Pütz, B. (1998). Transition of brain activation from frontal to parietal areas in visuomotor sequence learning. *Journal of Neuroscience*, *18*(5), 1827–1840.

Salamoura, A., & Williams, J. N. (2007). Processing verb argument structure across languages: Evidence for shared representations in the bilingual lexicon. *Applied Psycholinguistics*, *28*(4), 627–660.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human factors*, *48*(2), 362–380.

Salvucci, D. D., & Macuga, K. L. (2002). Predicting the effects of cellular-phone dialing on driver performance. *Cognitive Systems Research*, *3*(1), 95–102.

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

Salvucci, D. D., Taatgen, N. A., & Borst, J. P. (2009). Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1819–1828).

Saxe, A., Nelli, S., & Summerfield, C. (2020). If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*. doi: 10.1038/s41583-020-00395-8

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2019). A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, *116*(23), 11537–11546.

Schatz, J., Jones, S. J., & Laird, J. E. (2018). An architecture approach to modeling the remote associates test. In *Proceedings of the 16th international conference on cognitive modelling (iccm).*

Schlaug, G. (2001). The brain of musicians: a model for functional and structural adaptation. *Annals of the New York Academy of Sciences*, *930*(1), 281–299.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Schneider, W., & Chein, J. M. (2003). Controlled & automatic processing: behavior, theory, and biological mechanisms. *Cognitive science*, *27*(3), 525–559.

Schneider, W., & Detweiler, M. (1988). The role of practice in dual-task performance: toward workload modeling a connectionist/control architecture. *Human factors*, *30*(5), 539–566.

Schneider, W., Detweiler, M., et al. (1987). A connectionist/control architecture for working memory. *The psychology of learning and motivation*, *21*, 53–119.

Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. detection, search, and attention. *Psychological review*, *84*(1), 1.

Schubert, T., Fischer, R., & Stelzel, C. (2008). Response activation in overlapping tasks and the response-selection bottleneck. *Journal of Experimental Psychology: Human Perception and Performance*, *34*(2), 376.

Schumacher, E. H., Seymour, T. L., Glass, J. M., Fencsik, D. E., Lauber, E. J., Kieras, D. E., & Meyer, D. E. (2001). Virtually perfect time sharing in dual-task performance: Uncorking the central cognitive bottleneck. *Psychological science*, *12*(2), 101–108.

Seidenberg, M. S., Tanenhaus, M. K., Leiman, J. M., & Bienkowski, M. (1982). Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. *Center for the Study of Reading Technical Report; no. 240*.

Servan-Schreiber, E., & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*(4), 592.

Shadlen, M. N., & Newsome, W. T. (2001). Neural basis of a perceptual decision in the parietal cortex (area lip) of the rhesus monkey. *Journal of neurophysiology*, *86*(4), 1916–1936.

Shadmehr, R., & Holcomb, H. H. (1997). Neural correlates of motor memory consolidation. *Science*, *277*(5327), 821–825.

Shaffer, L. (1975). Multiple attention in continuous verbal tasks. *Attention and* *performance V*, 157–167.

Shah, P., & Miyake, A. (1996). The separability of working memory resources for spatial thinking and language processing: An individual differences approach. *Journal of experimental psychology: General*, *125*(1), 4.

Shenhav, A., Botvinick, M. M., & Cohen, J. D. (2013). The expected value of control: an integrative theory of anterior cingulate cortex function. *Neuron*, *79*(2), 217–240.

Shenhav, A., Musslick, S., Lieder, F., Kool, W., Griffiths, T. L., Cohen, J. D., & Botvinick, M. M. (2017). Toward a rational and mechanistic account of mental effort. *Annual review of neuroscience*, *40*, 99–124.

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: Ii. perceptual learning, automatic attending and a general theory. *Psychological review*, *84*(2), 127.

Simen, P., Contreras, D., Buck, C., Hu, P., Holmes, P., & Cohen, J. D. (2009). Reward rate optimization in two-alternative decision making: empirical tests of theoretical predictions. *Journal of Experimental Psychology: Human Perception and Performance*, *35*(6), 1865.

Simon, H. (1957). Models of man; social and rational.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sohn, M.-H., & Anderson, J. R. (2001). Task preparation and task repetition: Two-component model of task switching. *Journal of Experimental Psychology: General*, *130*(4), 764.

Sperling, G., & Melchner, M. J. (1978). The attention operating characteristic: Examples from visual search. *Science*, *202*(4365), 315–318.

Sporns, O., Honey, C. J., & Kötter, R. (2007). Identification and classification of hubs in brain networks. *PloS one*, *2*(10).

Stephan, D. N., & Koch, I. (2010). Central cross-talk in task switching: Evidence from manipulating input–output modality compatibility. *Journal of Experimental*

Psychology: Learning, Memory, and Cognition, *36*(4), 1075.

Sternberg, S. (1969). The discovery of processing stages: Extensions of donders' method. *Acta psychologica*, *30*(0), 276–315.

Stoeckel, L. E., Garrison, K. A., Ghosh, S. S., Wighton, P., Hanlon, C. A., Gilman, J. M., ... others (2014). Optimizing real time fmri neurofeedback for therapeutic discovery and development. *NeuroImage: Clinical*, *5*, 245–255.

Strobach, T., Frensch, P. A., & Schubert, T. (2012). Video game practice optimizes executive control skills in dual-task and task switching situations. *Acta psychologica*, *140*(1), 13–24.

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of experimental psychology*, *18*(6), 643.

Sudevan, P., & Taylor, D. A. (1987). The cuing and priming of cognitive operations. *Journal of Experimental Psychology: Human perception and performance*, *13*(1), 89.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science*, *12*(2), 257–285.

Taatgen, N. A., & Anderson, J. R. (2002). Why do children learn to say "broke"? a model of learning the past tense without feedback. *Cognition*, *86*(2), 123–155.

Tanaka, K. (1996). Inferotemporal cortex and object vision. *Annual review of neuroscience*, *19*(1), 109–139.

Tarjan, R. E., & Trojanowski, A. E. (1977). Finding a maximum independent set. *SIAM Journal on Computing*, *6*(3), 537–546.

Telford, C. W. (1931). The refractory phase of voluntary and associative responses. *Journal of Experimental Psychology*, *14*(1), 1.

Telgarsky, M. (2016). Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*.

Todd, P. M., & Gigerenzer, G. E. (2012). *Ecological rationality: Intelligence in the world.* Oxford University Press.

Tombu, M., & Jolicœur, P. (2003). A central capacity sharing model of dual-task performance. *Journal of Experimental Psychology: Human Perception and Performance*, *29*(1), 3.

Townsend, J. T. (1972). Some results concerning the identifiability of parallel and serial processes. *British Journal of Mathematical and Statistical Psychology*, *25*(2), 168–199.

Townsend, J. T. (1990). Serial vs. parallel processing: Sometimes they look like tweedledum and tweedledee but they can (and should) be distinguished. *Psychological Science*, *1*(1), 46–54.

Townsend, J. T., & Altieri, N. (2012). An accuracy–response time capacity assessment function that measures performance against standard parallel predictions. *Psychological review*, *119*(3), 500.

Townsend, J. T., Ashby, F., Castellan, N., & Restle, F. (1978). Cognitive theory.

Townsend, J. T., Ashby, F. G., et al. (1983). *Stochastic modeling of elementary psychological processes.* CUP Archive.

Townsend, J. T., & Fifić, M. (2004). Parallel versus serial processing and individual differences in high-speed search in human memory. *Perception & Psychophysics*, *66*(6), 953–962.

Townsend, J. T., & Nozawa, G. (1995). Spatio-temporal properties of elementary perception: An investigation of parallel, serial, and coactive theories. *Journal of Mathematical Psychology*, *39*(4), 321–359.

Townsend, J. T., & Wenger, M. J. (2004). A theory of interactive parallel processing: new capacity measures and predictions for a response time inequality series. *Psychological review*, *111*(4), 1003.

Treisman, A. M. (1996). The binding problem. *Current opinion in neurobiology*, *6*(2), 171–178.

Treisman, A. M. (1999). Solutions to the binding problem: progress through controversy and convergence. *Neuron*, *24*(1), 105–125.

Treisman, A. M., & Davies, A. (1973). *Divided attention to ear and eye, in attention and performance.* S. Kornblum, Editor.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, *12*(1), 97–136.

Turner, M. L., & Engle, R. W. (1986). Working memory capacity. In *Proceedings of the human factors society annual meeting* (Vol. 30, pp. 1273–1277).

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *science*, *185*(4157), 1124–1131.

Ueltzhöffer, K., Armbruster-Genç, D. J., & Fiebach, C. J. (2015). Stochastic dynamics underlying cognitive stability and flexibility. *PLoS computational biology*, *11*(6).

Usher, M., & Cohen, J. D. (1999). Short term memory and selection processes in a frontal-lobe model. In *Connectionist models in cognitive neuroscience* (pp. 78–91). Springer.

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: the leaky, competing accumulator model. *Psychological review*, *108*(3), 550.

Verbeke, P., & Verguts, T. (2019). Learning to synchronize: How biological agents can couple neural task modules for dealing with the stability-plasticity dilemma. *PLoS computational biology*, *15*(8), e1006604.

Verguts, T. (2017). Binding by random bursts: A computational model of cognitive control. *Journal of Cognitive Neuroscience*, *29*(6), 1103–1118.

Vince, M. A. (1948). Corrective movements in a pursuit task. *Quarterly Journal of Experimental Psychology*, *1*(2), 85–103.

von Neumann, J. (1958). *The computer and the brain.* USA: Yale University Press.

Walley, R. E., & Weiden, T. D. (1973). Lateral inhibition and cognitive masking: a neuropsychological theory of attention. *Psychological review*, *80*(4), 284.

Warren, R. E. (1972). Stimulus encoding and memory. *Journal of Experimental Psychology*, *94*(1), 90.

Waszak, F., Hommel, B., & Allport, A. (2004). Semantic generalization of stimulus-task bindings. *Psychonomic Bulletin & Review*, *11*(6), 1027–1033.

Webb, T. W., Dulberg, Z., Frankland, S. M., Petrov, A. A., O'Reilly, R. C., & Cohen, J. D. (2020). Learning representations that support extrapolation. *arXiv preprint* *arXiv:2007.05059*.

Welford, A. T. (1952). The psychological refractory period and the timing of high-speed performance-a review and a theory. *British Journal of Psychology*, *43*(1), 2.

Welford, A. T. (1967). Single-channel operation in the brain. *Acta psychologica*, *27*, 5–22.

Wendt, M., & Kiesel, A. (2008). The impact of stimulus-specific practice and task instructions on response congruency effects between tasks. *Psychological Research*, *72*(4), 425–432.

Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In *System modeling and optimization* (pp. 762–770). Springer.

West, D. B., et al. (2001). *Introduction to graph theory* (Vol. 2). Prentice hall Upper Saddle River.

West, R. F., & Stanovich, K. E. (1978). Automatic contextual facilitation in readers of three ages. *Child Development*, 717–727.

Westbrook, A., & Braver, T. S. (2015). Cognitive effort: A neuroeconomic approach. *Cognitive, Affective, & Behavioral Neuroscience*, *15*(2), 395–415.

Westbrook, A., van den Bosch, R., Määttä, J., Hofmans, L., Papadopetraki, D., Cools, R., & Frank, M. J. (2020). Dopamine promotes cognitive effort by biasing the benefits versus costs of cognitive work. *Science*, *367*(6484), 1362–1366.

Wickens, C. D. (1976). The effects of divided attention on information processing in manual tracking. *Journal of Experimental Psychology: Human Perception and Performance*, *2*(1), 1.

Wickens, C. D. (1991). Processing resources and attention. *Multiple-task performance*, *1991*, 3–34.

Wickens, C. D., & Kessel, C. (1979). The effects of participatory mode and task workload on the detection of dynamic system failures. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 24–34.

Wiener, N. (2019). *Cybernetics or control and communication in the animal and the machine*. MIT press.

Wilken, P., & Ma, W. J. (2004). A detection theory account of change detection. *Journal of vision*, *4*(12), 11–11.

Woodman, G. F., & Luck, S. J. (2003). Serial deployment of attention during visual search. *Journal of Experimental Psychology: Human Perception and Performance*, *29*(1), 121.

Wylie, G., & Allport, A. (2000). Task switching and the measurement of "switch costs". *Psychological research*, *63*(3-4), 212–233.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, *111*(23), 8619–8624.

Young, M. P. (1993). The organization of neural systems in the primate cerebral cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *252*(1333), 13–18.

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., & Savarese, S. (2018). Taskonomy: Disentangling task transfer learning. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3712–3722).

Appendix A: Graph Theory Preliminaries

Throughout the main text and the appendix, we make extensive use of some basic definitions and notation from graph theory. In this section, we review these. Additional background and information concerning graph theory can be found in Diestel (2005) and D. B. West et al. (2001).

An directed graph G is composed of a finite set of vertices, V and a set of edges, E which is a subset of the family of all 2-tuples of V. Namely each edge is an *ordered pair* (u, v) where both $u, v \in V$. We write $G = (V, E)$ to signify a graph G that consists of a vertex set V and edge set E. We say that a vertex y is an *neighbor* of x if $(x, y) \in E$. Alternatively, we say that y is adjacent to x.

The *degree* of x is defined as the number of neighbors of x. Given a list of vertices, $v_1, v_2, ..., v_r$ the degree sequence of these vertices is simply the list of the degrees of $v_1, v_2, ..., v_r$. The average degree of a graph is simply the sum of the degrees normalized by the number of vertices.

A *simple path* is a set of *distinct* vertices $v_1, v_2, ..., v_k$ such that for every $1 \leq i < k$, v_i is a neighbor to v_{i+1}. The *length* of the path is the number of vertices in the path minus one (that is, $k - 1$).

An *independent set* is a subset I of vertices that contains no edges. We refer to an independent set of maximal cardinality as an MIS (standing for maximal independent set). $G = (V, E)$ is *bipartite* if the vertex set of V is the union of two disjoint independent sets.

A *matching* is a set of edges M that are pairwise disjoint. Namely, no two edges in M share a vertex as an endpoint. A matching M' is *induced* if no two edges in M' are connected by a third edge.

The *line graph* $L(G)$ of a graph $G = (V, E)$ is a graph whose vertex set is the edges of G and two vertices in $L(G)$ are connected by an edge in $L(G)$ if the edges corresponding to them in G share a vertex (observe that the line graph may have parallel edges, namely if $v(e)$ and $v(f)$ are two edges corresponding to the edges e and f in G then the line graph may contain both the $(v(e), v(f))$ edge as well as the

$(v(f), v(e))$ edge. The *square* of a graph $G = (V, E)$ denote by G^2 has the same vertex set V as G. Two vertices in G^2 are connected if and only if there is a path of length at most 2 connecting them in G. It can be verified a set of vertices in the square of the line graph $L(G)$ is an independent set if and only if the edges in G, that correspond to these vertices in $L(G)$, form an induced matching.

Appendix B: Multitasking Capability in Deep Networks

Here, we derive an upper bound for the multitasking capability in deep networks. Recall from the section on "Analysis of Multitasking Capability" in the main text that we assume that we are given a network G that has $r \geq 2$ layers $L_1, ..., L_r$ where each layer is of size n. Every layer is an independent set and for every $i < r$, every vertex in L_i is connected to every vertex in L_{i+1} independently with probability p . In other words, for every $i < r$, the graph connecting L_i and L_{i+1} is a random bipartite graph where every $u \in L_i$ is connected to L_{i+1} with probability p independently of all other edges. Observe that we assume there are no "skip connections": there are no edges connecting L_i and L_j if $|i - j| > 1$.

Recall that a family of induced paths of size k is a set of k paths from L_1 to L_r that are vertex disjoint and furthermore, for any two vertices u, v belonging to two different paths, there is no edge in G connecting u to v. We use the first moment method commonplace in random graph theory to upper bound the likely size of k. We first upper bound the *expected* number of families of k induced paths going from the first layer to the rth layer. The expected number of such paths is

$$\binom{n}{k}^r p^{k(r-1)}((1-p)^{2(r-1)})^{k(k-1)/2} \tag{10}$$

.

Indeed, there are $\binom{n}{k}^r$ ways to choose the vertices in the k induced paths (observe that the k induced paths intersect L_i at exactly k vertices for every $1 \leq i \leq r$), the probability all these paths appear is $p^{k(r-1)}$ and the probability no two paths are connected by an edge is $((1-p)^{2(r-1)})^{k(k-1)/2}$. Here, we use the assumption that there are no “skip connections”: Every layer i has connection only to the $i+1$ or $i-1$ layers. Using the inequalities $\binom{n}{k} \le \left(\frac{en}{k}\right)^k$ and $1-p \le e^{-p}$, we get that the expected number of families containing k induced paths is at most

$$\left(\frac{enp^{\frac{r-1}{r}}}{k}\right)^{rk} e^{(-p(r-1)k+p(r-1))k} = \left(\left(\frac{enp^{\frac{r-1}{r}}}{k}\right)^{r} e^{-p(r-1)k+p(r-1)}\right)^{k}. \tag{11}$$

.

To prove the expectation is negligible (tending to zero with n), it suffices to find k such that the term inside the bracket is at most $\frac{1}{e}$. Taking logarithms (all logarithms are to the base of e) we get,

$$k = \left(1+\frac{1}{r-1}\right)\left(\frac{\log en - \log k}{p}\right) - \frac{\log(1/p)}{p} + \frac{1}{(r-1)p} + 1. \tag{12}$$

By Markov’s inequality, we get that with high probability (probability tending to 1 as n tends to infinity) the a family of t induced paths in G satisfies

$$t \le f(r,p,n) = \left(1+\frac{1}{r-1}\right)\left(\frac{\log en}{p}\right) - \frac{\log(1/p)}{p} \tag{13}$$

plus some low-order terms (e.g., terms whose asymptotic growth is much lower than $\frac{\log en}{p}$). Looking at this calculation, we see that for $p \ge \frac{w(n)\lg n}{n}$ where $\lim_{n\to\infty} w(n) = 0$, the largest number of tasks that can be multitasked is *sublinear* in n confirming our simulations and predictions in the main text (for $r = 2$). Assuming that $p \ge 1/n$, we can also see that $f(r,p,n)$ decays in r, and rate of the decay is lower bounded (when compared to the $r = 2$ case) by $1/2(r-1)$. Namely, we have that

$$\frac{f(r,p,n)}{f(2,p,n)} \ge \left(\left(1+\frac{1}{r-1}\right)\left(\frac{\log en}{p}\right) - \frac{\log(1/p)}{p}\right)/(2\log(en)/p) \ge \frac{1}{2(r-1)}. \tag{14}$$

Our bound on the expectation implies that with high probability there is no family of induced paths in G containing significantly more than $f(r,p,n)$ paths. One may ask whether our result is tight: is it true that there exist a family of induced paths with $(1-\delta)f(r,p,n)$ paths (where δ is an arbitrary positive constant smaller than 1) with high probability. While we believe that this is indeed the case, a formal proof or disproof is left for future work.

Appendix C: Tradeoff Between Learning Efficiency and Multitasking Capability in Gated Deep Linear Networks

Consider the setting with M stimulus dimensions $x_i \in R^N, i = 1, \cdots, M$ and M response dimensions $y_i \in R^N, i = 1, \cdots, M$ where each dimension consists of N neurons (processing units in a neural network). There are M^2 single tasks to perform, corresponding to all combinations of linking a stimulus dimension to a response dimension. Given a stimulus dimension m and response dimension n, the task to be performed is a function f linking only the specified stimulus dimension to the specified response dimension, $y_n = f(x_m)$, and all other response dimensions should be zero, $y_k = 0, k \neq n$. That is, the transformation applied from stimulus dimension to response dimension is identical for different tasks, which differ only in which dimensions are relevant. The transformation is learned based on a dataset of P inputs $X \in R^{N\times P}$ and associated desired outputs $Y \in R^{N\times P}$ where examples are placed in columns. Learning speed will depend on the second order statistics $\Sigma^{yx} = YX^T$ and $\Sigma^{xx} = XX^T$, and for simplicity, we assume that the inputs are whitened, $\Sigma^{xx} = I$.

To implement the mapping from input to output, we use a gated deep linear network containing a single hidden layer of neurons (Fig. 26). In this network, signal propagation is linear, except that individual neurons in the hidden and output layers are gated on or off on each example. The gating scheme is hand-specified, and different gating schemes will cause different learning dynamics and multitasking behavior. To describe the gating schemes we consider, it is useful to subdivide the hidden layer of neurons as follows. We divide the hidden layer into Q groups of neurons that will project to different response dimensions, described below; and each group is further subdivided into M sets of N neurons, one for each of the M stimulus dimensions. The overall hidden layer is thus of size QMN, and to foreshadow, the number of groups Q

will interpolate between the minimal basis set representation ($Q = 1$) and the tensor product representation ($Q = M$). We denote the hidden units devoted to stimulus dimension i, group j as the vector $h^{j,i} \in R^N$. We denote the weights from stimulus dimension i to its bank of hidden units in group j as $W_{hs}^{j,i}, i = 1, \cdots, M, j = 1, \cdots, M$. Similarly, we denote the output weights from the i^{th} stimulus dimension's set of hidden units in group j to the k^{th} response dimension as $W_{oh}^{k,j,i}$.

With these definitions, we now describe how the output of the gated deep linear network is computed for a given input. The network's hidden activity in response to an input is given by

$$h^{j,i} = g_h(i, j, c) W_{hs}^{j,i} x_i, \quad i = 1, \cdots, M, j = 1, \cdots, Q \tag{15}$$

where the scalar hidden gating function $g_h(i, j, c)$ is either one or zero (turning on or off this bank of hidden units) and is allowed to depend on the current task c, i.e., the relevant stimulus dimension and response dimensions. This gating function will be hand-chosen as described subsequently. The network's output is then

$$y_k = \sum_{j=1}^{Q} \sum_{i=1}^{M} g_o(k, c) W_{oh}^{k,j,i} h^{j,i}, \quad k = 1, \cdots, M \tag{16}$$

where similarly the output gating function $g_o(k, c)$ is either one or zero (turning on or off this bank of output units) and may depend on the task c. In this network, the impact of nonlinearity is to gate on or off certain sets of hidden and output neurons, depending on task context, via the gating functions g_h and g_o.

To train the network, all weight parameters are adjusted using gradient descent to minimize a loss function, which we choose to be the sum of squared error. The error for a task c is

$$SSE(c) \quad = \quad \frac{1}{2} \sum_{\mu=1}^{P} \sum_{k=1}^{M} \| \bar{y}_k(\mu, c) - y_k(\mu, c) \|_2^2 \tag{17}$$

$$\tag{18}$$

where $\bar{y}_k(\mu, c) \in R^N$ is the correct output for example μ on task c, and we have made the dependence of the network's output on μ and c explicit.

When the network is trained on the set S of all M^2 single-tasking tasks, we have the total loss

$$\mathcal{L} \;=\; \sum_{c\in S} SSE(c). \tag{19}$$

Every weight parameter w in the network is updated via continuous time gradient descent,

$$\tau \frac{d}{dt} w = -\frac{\partial \mathcal{L}}{\partial w}. \tag{20}$$

Taking the derivative for a single task c with respect to the hidden-to-output weights, we have

$$\begin{aligned}
\frac{\partial SSE(c)}{\partial W_{oh}^{q,r,s}} &= \frac{\partial}{\partial W_{oh}^{q,r,s}} \frac{1}{2} \sum_{\mu=1}^{P} \sum_{k=1}^{M} \| \bar{y}_k(\mu, c) - y_k(\mu, c) \|_2^2 && (21)\\
&= \frac{\partial}{\partial W_{oh}^{q,r,s}} \frac{1}{2} \sum_{\mu=1}^{P} \sum_{k=1}^{M} \left\| \bar{y}_k(\mu, c) - \sum_{j=1}^{Q} \sum_{i=1}^{M} g_o(k, c) W_{oh}^{k,j,i} g_h(i, j, c) W_{hs}^{j,i} x_i(\mu) \right\|_2^2 && (22)\\
&= \frac{1}{2} \sum_{\mu=1}^{P} \frac{\partial}{\partial W_{oh}^{q,r,s}} \left\| \bar{y}_q(\mu, c) - \sum_{j=1}^{Q} \sum_{i=1}^{M} g_o(q, c) W_{oh}^{q,j,i} g_h(i, j, c) W_{hs}^{j,i} x_i(\mu) \right\|_2^2 && (23)\\
&= \sum_{\mu=1}^{P} e_q(\mu, c) g_o(q, c) g_h(s, r, c) \left[W_{hs}^{r,s} x_s(\mu) \right]^T && (24)\\
&= \sum_{\mu=1}^{P} \left[\bar{y}_q(\mu, c) - \sum_{j=1}^{Q} \sum_{i=1}^{M} g_o(q, c) W_{oh}^{q,j,i} g_h(i, j, c) W_{hs}^{j,i} x_i(\mu) \right] \times && (25)\\
&\quad g_o(q, c) g_h(s, r, c) \left[W_{hs}^{r,s} x_s(\mu) \right]^T && (26)\\
& && (27)
\end{aligned}$$

Hence the derivative will be zero if the response dimension to which these weights project is gated off ($g_o(q, c) = 0$), or if the hidden group for this output and stimulus dimension is gated off ($g_h(s, r, c) = 0$). When the task c is a single-tasking scenario in which stimulus dimension γ and response dimension ν are relevant,

Now we use the fact that $\bar{y}_q(\mu, c)$ and $g_o(q, c)$ are both zero unless response dimension q is on in task c. Let ν be the response dimension for task c. Then we have

$$\frac{\partial SSE(c)}{\partial W_{oh}^{q,r,s}} = 0 \quad \text{if } q \neq \nu \tag{28}$$

and if $q = \nu$,

$$\frac{\partial SSE(c)}{\partial W_{oh}^{q,r,s}} = \sum_{\mu=1}^{P}\left[\bar{y}_\nu(\mu) - \sum_{j=1}^{Q}\sum_{i=1}^{M} W_{oh}^{\nu,j,i} g_h(i,j,c) W_{hs}^{j,i} x_i(\mu)\right] g_h(s,r,c)\left[W_{hs}^{r,s} x_s(\mu)\right]^T. \quad (29)$$

In single task training, the hidden gating function $g_h(i,j,c)$ is zero unless i corresponds to the desired stimulus dimension γ. Hence $\frac{\partial SSE(c)}{\partial W_{oh}^{q,r,s}} = 0$ if $s \neq \gamma$, and otherwise,

$$\frac{\partial SSE(c)}{\partial W_{oh}^{q,r,s}} = \sum_{\mu=1}^{P}\left[\bar{y}_\nu(\mu) - \sum_{j=1}^{Q} W_{oh}^{\nu,j,\gamma} g_h(\gamma,j,c) W_{hs}^{j,\gamma} x_\gamma(\mu)\right] g_h(\gamma,r,c)\left[W_{hs}^{r,\gamma} x_\gamma(\mu)\right]^T. \quad (30)$$

Finally, $g_h(\gamma,j,c)$ is zero unless group j projects to response dimension ν. Let ξ be the group index for response dimension q. Then we have

$$\frac{\partial SSE(c)}{\partial W_{oh}^{q,r,s}} = \begin{cases} \sum_{\mu=1}^{P}\left[\bar{y}_\nu(\mu) - W_{oh}^{\nu,\xi,\gamma} W_{hs}^{\xi,\gamma} x_\gamma(\mu)\right]\left[W_{hs}^{\xi,\gamma} x_\gamma(\mu)\right]^T & \text{if } q = \nu, r = \xi, s = \gamma \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

Using the fact that all tasks require the same input-output mapping, this can be rearranged to

$$\frac{\partial SSE(c)}{\partial W_{oh}^{q,r,s}} = \begin{cases} \left(\Sigma^{yx} - W_{oh}^{\nu,\xi,\gamma} W_{hs}^{\xi,\gamma}\Sigma^{xx}\right)\left(W_{hs}^{\xi,\gamma}\right)^T & \text{if } q = \nu, r = \xi, s = \gamma \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

Hence, when training in single-tasking context, only the hidden-to-output weights which project from the relevant hidden input group to the relevant response dimension, and are part of a group which are active for this response dimension will change. The form of this change is exactly the same as in a deep linear network, a fact that we will exploit below.

Summing the contributions from all single tasks yields the learning dynamics for the overall loss $\mathcal{L}$ for single task training,

$$\frac{\partial \mathcal{L}}{\partial W_{oh}^{q,r,s}} = \begin{cases} \left(\Sigma^{yx} - W_{oh}^{q,r,s} W_{hs}^{r,s}\Sigma^{xx}\right)\left(W_{hs}^{r,s}\right)^T & \text{if } r = v(q) \\ 0 & \text{otherwise} \end{cases} \quad (33)$$

where $v(q)$ is a function mapping an response dimension to its associated hidden unit group. Hence, under single task training, the hidden-to-output weights between a hidden unit group and its associated output dimension change according to standard dynamics in a deep linear network, and connections from other groups to the relevant output remain unchanged.

We now calculate the derivative for a single task c with respect to the input weights,

$$\begin{align}
\frac{\partial SSE(c)}{\partial W_{hs}^{r,s}} &= \frac{\partial}{\partial W_{hs}^{r,s}} \frac{1}{2} \sum_{\mu=1}^{P} \sum_{k=1}^{M} \|\bar{y}_k(\mu,c) - y_k(\mu,c)\|_2^2 \tag{34}\\
&= \frac{\partial}{\partial W_{hs}^{r,s}} \frac{1}{2} \sum_{\mu=1}^{P} \sum_{k=1}^{M} \left\| \bar{y}_k(\mu,c) - \sum_{j=1}^{Q} \sum_{i=1}^{M} g_o(k,c) W_{oh}^{k,j,i} g_h(i,j,c) W_{hs}^{j,i} x_i(\mu) \right\|_2^2 \tag{35}\\
&= \sum_{\mu=1}^{P} \sum_{k=1}^{M} g_o(k,c) \left(W_{oh}^{k,r,s}\right)^T [\bar{y}_k(\mu,c) \tag{36}\\
&\quad - \sum_{j=1}^{Q} \sum_{i=1}^{M} g_o(k,c) W_{oh}^{k,j,i} g_h(i,j,c) W_{hs}^{j,i} x_i(\mu) \Big] g_h(s,r,c) x_s(\mu)^T \tag{37}\\
& \tag{38}
\end{align}$$

Under the single tasking gating scheme where task c links input dimension m to output dimension n, this simplifies to

$$\begin{align}
\frac{\partial SSE(c)}{\partial W_{hs}^{r,s}} &= \sum_{\mu=1}^{P} (W_{oh}^{n,r,s})^T \left[\bar{y}_n(\mu,c) - W_{oh}^{n,v(n),m} W_{hs}^{v(n),m} x_m(\mu)\right] g_h(s,r,c) x_s(\mu)^T \tag{39}\\
&= \tag{40}
\end{align}$$

where in the first step we have used the fact that $g_o(k,c)$ is zero unless $k = n$ and $g_h(i,j,c)$ is zero unless $i = m, j = v(n)$ ($v(n)$ is the hidden group associated with output group n). Hence the update will be zero, unless $s = m$ and $r = v(n)$. Notably, this means the update can be nonzero for tasks with different output dimensions n.

Summing over all single tasks, we have the update

$$\begin{align}
\frac{\partial \mathcal{L}}{\partial W_{hs}^{r,s}} &= \sum_{\mu=1}^{P} (W_{oh}^{n,r,s})^T \left[\bar{y}_n(\mu,c) - W_{oh}^{n,v(n),m} W_{hs}^{v(n),m} x_m(\mu)\right] g_h(s,r,c) x_s(\mu)^T \tag{41}\\
& \tag{42}
\end{align}$$

$$\frac{\partial \mathcal{L}}{\partial W_{hs}^{r,s}} = \begin{cases} \left(\Sigma^{yx} - W_{oh}^{q,r,s} W_{hs}^{r,s} \Sigma^{xx}\right) \left(W_{hs}^{r,s}\right)^T & \text{if } r = v(q) \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

We thus have the SSE

$$\begin{aligned} SSE &= \frac{1}{2}\sum_{\mu=1}^{M}\sum_{\nu=1}^{M}\left\|Y^{\mu,\nu} - \hat{Y}^{\mu,\nu}\right\|_F^2 && (44)\\ &= \frac{1}{2}\sum_{\mu=1}^{M}\sum_{\nu=1}^{M}\|Y^{\mu,\nu} - W_2^{\mu}W_1^{\nu}X^{\mu,\nu}\|_F^2 && (45)\end{aligned}$$

The gradient is thus

$$\begin{aligned} \frac{\partial SSE}{\partial W_2^{\mu}} &= \frac{1}{2}\sum_{\nu=1}^{M}\frac{\partial}{\partial W_2^{\mu}}\|Y^{\mu,\nu} - W_2^{\mu}W_1^{\nu}X^{\mu,\nu}\|_F^2 && (46)\\ &= \sum_{\nu=1}^{M}\left(Y^{\mu,\nu}(X^{\mu,\nu})^T - W_2^{\mu}W_1^{\nu}X^{\mu,\nu}(X^{\mu,\nu})^T\right)W_1^{\nu^T} && (47)\\ \frac{\partial SSE}{\partial W_1^{\nu}} &= \frac{1}{2}\sum_{\mu=1}^{M}\frac{\partial}{\partial W_1^{\nu}}\|Y^{\mu,\nu} - W_2^{\mu}W_1^{\nu}X^{\mu,\nu}\|_F^2 && (48)\\ &= \sum_{\mu=1}^{M}W_2^{\mu^T}\left(Y^{\mu,\nu}(X^{\mu,\nu})^T - W_2^{\mu}W_1^{\nu}X^{\mu,\nu}(X^{\mu,\nu})^T\right) && (49)\end{aligned}$$

Finally, assuming identical tasks and similar initializations $W_1 = W_1^{\nu},\ W_2 = W_2^{\mu}$ for all μ, ν, we have

$$\begin{aligned} \frac{\partial SSE}{\partial W_2} &= M\left(\Sigma^{yx} - W_2W_1\Sigma^{xx}\right)W_1^T && (50)\\ \frac{\partial SSE}{\partial W_1} &= MW_2^T\left(\Sigma^{yx} - W_2W_1\Sigma^{xx}\right) && (51)\end{aligned}$$

Hence the impact of multitasking is simply to pick up a factor of M in the learning rate, relative to learning each task independently. Using the usual SVD results for linear networks, this means that each mode of the SVD will be learned in time

$$t = \frac{\tau}{Ms}\ln(s/\epsilon) \tag{52}$$

where s is the singular value of the input-output mode, τ is the inverse learning rate, and ϵ is a small cutoff (assuming whitened inputs; this can be relaxed).

Hence this input-output gating scheme learns in time roughly $O(1/M)$, and sits as a midpoint along a continuum: if we knew that all tasks were identical and parameter updates could be fully shared, we could learn the task in time $O(1/M^2)$. If we used a tensor product representation, we would learn each task as though it were completely independent, yielding an $O(1)$ learning time.

Letting $N = M^2$ be the total number of tasks, we can rewrite this as an $O(1/\sqrt{N})$ advantage in learning speed over the tensor product representation.

There is also an advantage in terms of representational resources required. The gating strategy requires $O(MP)$ neurons in its hidden layers to implement the transformation where P is the number of input/output units per dimension. In contrast the tensor product strategy requires $O(M^2P)$; or rephrased in terms of the total number of tasks, $O(P\sqrt{N})$ and $O(PN)$ respectively. This can yield substantial savings.

Performing Multiple Tasks Simultaneously

Can multiple tasks be performed at the same time? One might hope that simply setting the gating variables to allow two tasks to pass through would enable good performance. However this idea fails completely because each task will linearly interfere with the other in the minimal basis set representation. In particular, if tasks (μ_1, ν_1) and (μ_2, ν_2) are attempted simultaneously, the output will be $\hat{y} = W_2W_1(x^{\mu_1,\nu_1} + x^{\mu_2,\nu_2})$ at both output locations.

In the tensor product representation, however, two tasks can errorlessly be performed at the same time simply by activating the appropriate elements in the tensor product. In fact, M tasks can be performed simultaneously (the maximum number which can be accommodated given the M response dimensions).

Are there intermediate options between the $O(1/M)$ learning but $O(1)$ multitasking of the input-output gating scheme and the $O(1)$ learning but $O(M)$ multitasking of the tensor product? Suppose we wish to be able to perform just Q tasks simultaneously. We may divide the M output task dimensions into Q groups, and apply the input gating scheme to each group independently. Each group has M/Q response dimensions which constitute it, and hence is learned in time $O(Q/M)$. We thus have the following tradeoff:

$$t = \frac{\tau Q}{Ms} \ln(s/\epsilon) \tag{53}$$

or $t \propto Q/M$. In words, this is learning speed = # of input/response dimensions divided by # of concurrently executable tasks.