

# A Mechanistic Account of Constraints on Control-Dependent Processing: Shared Representation, Conflict and Persistence

Sebastian Musslick<sup>1,\*</sup>, and Jonathan D. Cohen<sup>1</sup>

<sup>1</sup>Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544, USA.

\*Corresponding Author: [musslick@princeton.edu](mailto:musslick@princeton.edu)

## Abstract

One of the most fundamental and striking limitations of human cognitive function is the constraint on the number of control-dependent processes that can be executed simultaneously. However, the sources of this capacity constraint remain largely unexplored. Previous work has attributed the constraints on control-dependent processing to the sharing of representations between tasks in neural systems. Here, we examine how shared representations interact with two other factors in producing constraints on control-dependent processing. We first demonstrate that the detrimental effects of shared representations on multitasking performance are contingent on the amount of conflict that is induced by the tasks that share representations. We then examine how the persistence of shared representations between tasks affects processing interference during serial task execution. Finally, we discuss how this set of mechanisms can account for various phenomena in neural architectures, including the psychological refractory period, task switch costs, as well as constraints on cognitive control.

**Keywords:** cognitive control; capacity constraint; dual-tasking; psychological refractory period; neural networks

## Introduction

Despite the powerful abilities that cognitive control affords, and its ubiquitous engagement in daily life (e.g., mentally planning a grocery list, or navigating a new route to work), the capacity for controlled processing appears to be strikingly limited (e.g., the inability to plan and navigate at once). This limitation has been literally paradigmatic in defining cognitive control: it has been 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 (Meyer & Kieras, 1997a; Welford, 1952).

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 (Posner & Snyder, 1975; Shiffrin & Schneider, 1977). However, alternative (“multiple-resource”) accounts (Allport, 1980; Meyer & Kieras, 1997a; Navon & Gopher, 1979; Salvucci & Taatgen, 2008) have suggested that the capacity constraints reflect properties of the processes that are being controlled. This proposes that control-demanding tasks, like any others, rely on a constellation of “local” resources; that is, task-specific representa-

tions, and that the inability to perform more than one task at a time may reflect the conflict that arises when the tasks involved demand that the same set of representations be used for different purposes, rather than reliance on a single centralized control mechanism. From this perspective, the very purpose of cognitive control is to prevent interference by limiting the number of task processes that make use of shared representations (Cohen, Dunbar, & McClelland, 1990; Botvinick, Braver, Barch, Carter, & Cohen, 2001).

One may argue that the constraints that shared representations between tasks impose on multitasking are negligibly small in a processing system as large as the human brain. However, simulation studies (Feng, Schwemmer, Gershman, & Cohen, 2014), followed by analytic work (Musslick et al., 2016) have studied the multitasking capability of two-layer neural networks as a function of the sharing of representations among tasks and found that the multitasking capability of a network drops precipitously with an increase in shared representations, and is virtually invariant to network size. Moreover, neural architectures appear subject to a tradeoff between learning efficiency and generalization that is promoted through the use of shared task representations, on the one hand, and processing efficiency and multitasking capability that is achieved through the separation of task representations, on the other hand (Musslick et al., 2017). This suggests that limitations in multitasking may reflect a preference of the neural system to learn tasks more quickly (Musslick et al., 2017; Sagiv, Musslick, Niv, & Cohen, 2018).

The studies above were based on the assumption that shared representations between tasks always cause interference. However, the amount of processing interference received by a single task has been shown to depend on the processing strength (automaticity) of the interfering task (Cohen et al., 1990; MacLeod & Dunbar, 1988). Another assumption made by these neural network studies is that multitasking can only be achieved by processing tasks concurrently. However, this assumption does not capture processing interference observed in the sequential execution of multiple tasks (Pashler, 1984; Welford, 1952), task switching effects (Allport, Styles, & Hsieh, 1994), nor multitasking behavior along a continuum from pure parallelism, through rapid task switching, to pure sequential processing (Salvucci, Taatgen, & Borst, 2009).

In this work, we examine the interactive effect of (a) shared representations between tasks, (b) the conflict induced by

shared representations and (c) the persistence of representations on the constraints on control-dependent processing in two-layered, feed-forward, non-linear networks. Our findings suggest that the detrimental effect of shared representations on multitasking interference is only present if the tasks that share representations induce a sufficient amount of conflict between each other, and that persistence of those representations can lead to delays in the serial execution of two tasks. Finally, we discuss how this set of mechanisms may provide a unifying account of various cognitive phenomena in neural architectures, including the psychological refractory period, task switch costs, as well as constraints on cognitive control.

### Neural Network Model

For the simulations described in the paper we focus on a network architecture that has been used to simulate a wide array of empirical findings concerning human performance (e.g. Cohen et al., 1990; Gilbert & Shallice, 2002), including limitations in multitasking (Musslick et al., 2016). In this section we lay out the architecture of this network, its processing, as well as the task environments used to train it.

### Network Architecture and Processing

The network consisted of the following layers (Figure 1): an input layer with two partitions, one of which represented the current stimulus (nine units) and projected to an associative layer, and another that encoded the current task (five units) and projected to both the associative and output layers; an associative layer (100 units) that projected to the output layer; and an output layer (nine units) that represented the network’s response. Input units were grouped by the stimulus dimensions relevant to performing each task (three units per dimension), and used a one-hot encoding (i.e., a single unit in a stimulus dimension was used to represent the current stimulus feature; the current stimulus feature was clamped to 1 and all others were clamped to 0). 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. Each response dimension of the output layer projected to a leaky competitive accumulator (LCA, Usher & McClelland, 2001) layer (described below), which determined the response for that dimension.

The network was instructed to perform a given task by specifying the current stimulus and task to be performed in 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 associative layer, and then passed through a logistic function to determine the pattern of activity over the units in the associative layer. This pattern was then used (together with the set of direct projections from the task layer) to determine the pattern of activity over the output layer.

The final response within a given response dimension of the network was determined by an LCA (Usher & McClelland, 2001) layer, implementing the assumption that the net-

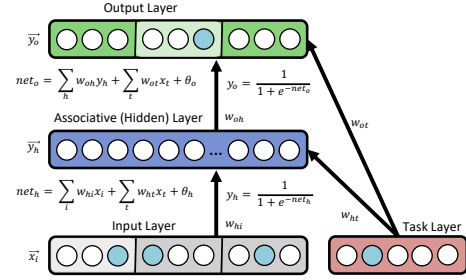


Figure 1: Feedforward neural network used in simulations. The input layer is composed of stimulus vector  $\vec{x}_i$  and task vector  $\vec{x}_t$ . The activity of each element in the associative layer  $y_h \in \vec{y}_h$  is determined by all elements  $x_i$  and  $x_t$ , and their respective weights  $w_{hi}$  and  $w_{ht}$  to  $y_h$ . Similarly, the activity of each output unit  $y_o \in \vec{y}_o$  is determined by all elements  $y_h$  and  $x_t$ , and their respective weights  $w_{oh}$  and  $w_{ot}$  to  $y_o$ . A bias of  $\theta = -2$  is added to the net input of all units  $y_h$  and  $y_o$ . Blue shades in the input and output units (circles) correspond to unit values of 1 and illustrate an example input pattern with its respective output pattern: The second task requires the network to map the vector of values in the first three feature units to one out of three output units (white shade).

work could only provide one response per dimension (e.g. the network cannot say RED and GREEN at the same time). 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 was determined by the accumulation of activity by each LCA unit, and the competition among them, the dynamics of which were given by

$$dr_i = [y_o - \lambda r_i + \alpha f(r_i) - \beta \sum_{j \neq i} f(r_j)] \frac{dt}{\tau} + \xi_i \sqrt{\frac{dt}{\tau}} \quad (1)$$

where  $y_o$  is the activity of the corresponding response unit in response dimension  $k$ ,  $\lambda$  is the decay rate of  $r_i$ ,  $\alpha$  is the recurrent excitation weight of  $r_i$ ,  $\beta$  is the inhibition weight between LCA units,  $\tau$  is the rate constant, and  $\xi$  is noise sampled from a Gaussian distribution with zero mean and standard deviation  $\sigma$ . The activity of each LCA response unit was lower bounded by zero via 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 response 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) for that dimension was the average number of time steps required for the response to reach threshold, scaled by a factor of 0.1. The following parameter values were used for all reported simulations:  $\lambda = 0.4$ ,  $\alpha = 0.2$ ,  $\beta = 0.2$ ,  $\sigma = 0.1$ , and  $z$  for each LCA layer was chosen as the threshold that maximizes reward rate ( $P(\text{correct})_k / (ITI + RT_k)$ ) for that dimension, where ITI corresponds to an inter-trial interval of 1s.

## Task Environment

Stimulus input units are structured according to stimulus dimensions (subvectors of the stimulus pattern), each of which was comprised of three feature units with only one feature unit activated per dimension. A task was defined as a mapping from the three stimulus features of a task-relevant stimulus dimension to three output units of a task-specific response dimension, so that only one of the three relevant output units was permitted to be active (see Fig. 1). For each simulation we considered the tasks A-E shown in Figure 2. Tasks A, B and C each map a different stimulus dimension to a different response dimension. Task D shares a stimulus dimension with Task A and shares a response dimension with Task B. Conversely, Task E shares a stimulus dimension with Task B and shares a response dimension with Task A.

Networks were initialized with a set of small random weights and then trained using the backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986) to produce the task-specified response for each stimulus in each task, while suppressing all other responses (both within the task-relevant output dimension, and all task-irrelevant output dimensions). The network was trained in epochs, with each epoch containing 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 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.

## Shared Representation and Conflict

Multitasking limitations have been attributed to shared representations between tasks as they engender interference. However, the amount of interference introduced by shared representation is known to depend on how much conflict they transmit (Cohen et al., 1990; MacLeod & Dunbar, 1988). To illustrate this, consider the simultaneous execution of Tasks A and B depicted in Figure 2. The network can execute a task by limiting processing to the representations involved for that task. For instance, the network can execute Task A by allocating control to the representation that encodes the task-relevant stimulus features for Task A in the associative layer and to the task-relevant response units for Task A in the output layer. Executing Tasks A & B simultaneously would require allocating control to the representations for both tasks in both layers. However, allocating control to Task A would engage Task D if the two tasks share a representation at the associative layer. Once Task D is engaged, it interferes with Task B at the output layer. Similarly, allocating control to a shared associative representation between Tasks B and E would introduce interference with Task A. Shared representations between Tasks A and D, as well as between Tasks B and E therefore introduce a functional dependence between Tasks A and B (Figure 2; Musslick et al., 2016). In contrast, no such interference is expected when the network performs

Tasks A & C at the same time.

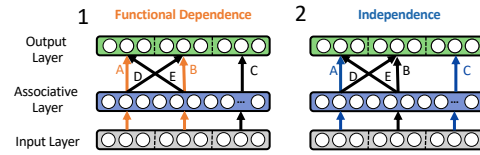


Figure 2: Illustration of dependencies between tasks. (1) Tasks A & B are considered functionally dependent due to shared representations with Tasks D and E, whereas (2) Tasks A & C are considered independent (see text).

In the example above, the amount of conflict introduced by Tasks D and E should decrease if the processing strength of both tasks is weak compared to the processing strength of Tasks A and B. Previous studies have demonstrated that extensive training on a task increases its processing strength which can induce greater conflict with other tasks (Cohen et al., 1990; MacLeod & Dunbar, 1988). This suggests a dilemma: While training on Tasks D and E should improve performance for each individual task, it should also lead to greater interference when dual-tasking seemingly unrelated Tasks A & B. However, dual-tasking performance for the two independent Tasks A & C should be unaffected. Here, we investigated the tradeoff between improvements in single task performance for Tasks D and E, on the one hand, and impairments in dual-task performance for Tasks A & B, as well as Tasks A & C, on the other hand, by varying the amount of training that a network receives for Tasks D and E. We were particularly interested in the amount of training that is required to cause impairments in dual-tasking performance.

We started by initializing 20 networks per training condition. In each condition, we sampled 100 patterns for each of the three Tasks A, B and C per training epoch. However, we varied the number of sampled training patterns for Tasks D and E from 0 (0% task strength) to 150 (150% task strength) across conditions. We then trained every network until it reached performance criterion for Tasks A, B and C. After training, we evaluated whether the network learned shared representations between Tasks A and D, and Tasks B and E in the associative layer of the network. In order to assess the similarity of learned task representations we focus our analysis on the weights from the task units to the associative layer, insofar as these reflect the computations carried out by the network required to perform each task. For a given pair of tasks we compute the learned representational similarity between them as the Pearson correlation of their weight vectors to the associative layer. Finally, we assessed the multitasking accuracy for performing Tasks A & B and the multitasking accuracy for performing Tasks A & C, as well as the single task accuracies of Task D and Task E.

Figure 3.1 shows the correlation between learned task representations in the associative layer of the network, averaged across all networks. As expected, Task A developed a shared representation with Task D in the associative layer since both tasks rely on the same set of stimulus features, as is the case

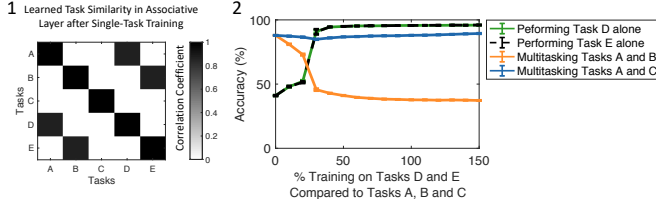


Figure 3: Effects of shared representation and conflict. (1) Average correlations between learned task representations in the associative layer. (2) Multitasking performance for Tasks A & B and Tasks A & C as a function of training on Tasks D and E. Error bars show the standard error of the mean across 20 simulated networks.

for Tasks B and E. Critically, dual-tasking performance for Tasks A & B decreased with the amount of training on Task D and Task E while dual-tasking performance for Tasks A & C was virtually unaffected by the training condition. Even small amounts of training on Tasks D & E (30%) improve performance on these tasks at the expense of impaired multitasking performance of Tasks A & B. Altogether, these results suggest that shared representations alone do not impose constraints on control-dependent processing, but they do so in combination with conflict.

### Shared Representation and Persistence

In the network model described above, limitations in multitasking can be circumvented by executing the individual tasks in series. However, a large body of evidence suggests that humans are subject to dual-task interference, even if they execute two tasks one after another (Welford, 1952). To illustrate this, consider the serial execution of two tasks in the psychological response period (PRP) paradigm (Figure 4). A trial in this paradigm begins with the presentation of a stimulus relevant to the first task, followed by a stimulus for the second task. The time between the onset of the first and second stimuli is referred to as stimulus onset asynchrony (SOA) and serves as an independent variable. Participants tend to respond slower to the second stimulus when the SOA is reduced (Welford, 1952). 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 and serves as the dependent variable.

Symbolic architectures explain the PRP effect in terms of processing bottlenecks that delay execution of the second task while the first task is still being executed (Meyer & Kieras, 1997a; Navon & Gopher, 1979; Salvucci & Taatgen, 2008; Pashler, 1994). While some accounts, such as the EPIC model (Meyer & Kieras, 1997a, 1997b) or the ACT-R/PM model (Byrne & Anderson, 2001) attribute the PRP partly to structural limitations in perceptual processing or motor execution, other accounts claim that the bottleneck is located at a “central” processing stage for response selection (Pashler, 1994) that is preceded by sensory processes and followed by processes for motor execution. However, to date, there is no account of this effect in neural network architectures. For instance, in the feed-forward model considered above, tasks can either be executed concurrently, with the risk of multitasking

interference, or in serial, without any risk of interference.

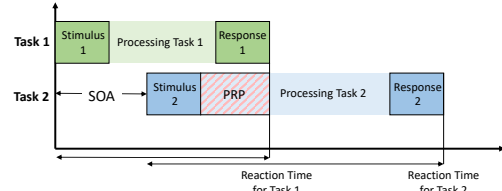


Figure 4: Psychological refractory period paradigm.

A crucial computational feature of neural systems is the integration of information over time, through persisting patterns of activity. Persistence characteristics can account for sequential processing of stimuli (Elman, 1990), working memory (Miyake & Shah, 1999), reconfiguration costs associated with switching tasks (Gilbert & Shallice, 2002) and many other cognitive phenomena. Persistence may also provide a mechanism for how the detrimental effects of shared representation on dual-task interference extend to the sequential execution of two tasks: the more a shared representation of a previously executed task persists in time, the more it may interfere with a subsequent task.

Here, we examine how shared representations interact with the persistence of activity in producing the PRP effect. To examine the PRP effect as a function of both, we first trained 10 networks on Tasks A-E until each network reached the performance criterion across all tasks. After training, we introduced persistence<sup>1</sup> in the computation of the net input of a unit  $i$  in the associative and output layers,

$$\overline{net}_i^T = (1 - p) \cdot net_i^T + p \cdot \overline{net}_i^{T-1}, \quad (2)$$

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. Thus, the higher  $p$ , the longer activity persists over time. For each network, we considered different values for  $p \in \{0, 0.5, 0.8, 0.9\}$ .

We then simulated the PRP paradigm for two pairs of tasks, A & B, as well as A & C. As demonstrated in the previous section, Tasks A & B are functionally dependent and interfere with each other when executed simultaneously whereas Tasks A & C are independent and interfere less. In both cases, the network was instructed to perform Task A second. Thus, we first presented the network with a feature from the stimulus dimension relevant to Task B or Task C, by activating the corresponding unit in the input layer and by keeping all other 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 the second task (Task A), by activating a unit in the input dimension relevant to that task while the stimulus feature for the first task

<sup>1</sup>Note that persistence in neural networks is typically implemented in the form of recurrent connections between the processing units. Here, we chose, for simplicity, to implement persistence by explicitly integrating processed information over time.



(Task B or Task C) was still present. PRP studies commonly instruct participants to give priority to the first task (Koch, Poljac, Müller, & Kiesel, 2018). We therefore activated the task layer unit for the first task at the beginning of each trial<sup>2</sup> and then determined the optimal onset of the task layer unit for the second task such that the joint reward rate for both tasks is maximized,

$$\text{Reward Rate} = \frac{P(\text{correct})_{\text{first task}} P(\text{correct})_{\text{second task}}}{(\text{ITI} + \text{RT}_{\text{total}})} \quad (3)$$

where  $P(\text{correct})_{\text{first task}}$ ,  $P(\text{correct})_{\text{second task}}$  correspond to the accuracies of the first and the second task, respectively, ITI corresponds to an inter-trial interval of 1 s, and  $\text{RT}_{\text{total}}$  is the reaction time of the last executed task, measured from the onset of the trial. We then assessed RTs for the first (Task B or Task C) and the second task (Task A) as a function of SOA, by varying the SOA from 1s to 8s in steps of 1s. Finally, we repeated the same analysis for 10 networks that were trained, within each epoch, on 100 patterns of dual-tasking Tasks A & B, as well as 100 patterns for dual-tasking Tasks A & C, in addition to being trained to perform all single tasks as described above. As in the previous section, we also assessed learned representational similarity between tasks as the Pearson correlation of their weight vectors to the associative layer.

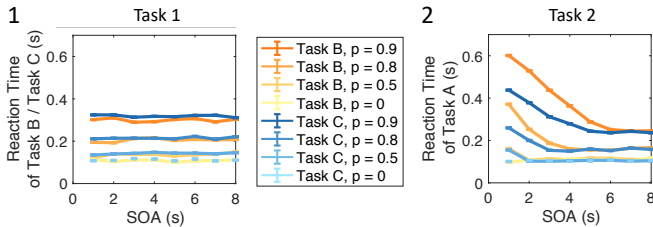


Figure 5: RTs of (1) the first and (2) the second task in the PRP paradigm as a function of persistence  $p$  and task. Error bars show the standard error of the mean across 10 simulated networks trained only on single tasks.

Simulation results indicate that higher persistence prolongs the reaction time for both the first and the second task (Figure 5). Moreover, the model replicates the PRP effect, showing a delay of the second task as a function of SOA (Figure 5.2). The delay in RT is overall higher after executing Task B compared to Task C, indicating that Task B interferes more with the subsequently executed Task A. This observation matches simulation results from the previous section, indicating that shared representations between Tasks A & D, as well as Tasks B & E lead to processing interference. However, shorter SOAs still affected RTs for Task A after executing Task C, indicating that there is processing interference between Tasks A & C that is not captured by shared representations in the associative layer alone. Interestingly, higher persistence amplifies the RT difference between Task A followed by a functionally dependent task and Task A followed by an independent task. In line with prior observations (Marill, 1957; Pashler, 1994),

<sup>2</sup>We assumed that the task layer unit for the first task becomes deactivated as soon as the model responded to the first stimulus.

the RT of the first task remained unaffected by the SOA, irrespective of whether the first task was functionally dependent or independent of the second task. This observation reflects the embedded strategy of the model to first execute the task associated with the first stimulus. Finally, we observed that dual-task training reduces the amount of shared representation between tasks that rely on a common stimulus dimension (Tasks A and D, as well as Tasks B and E; see Figure 6.1), compared to training the network on single tasks only (cf. Figure 3.1). In addition, training on both dual-task conditions yielded significant reductions in the PRP effect despite high levels of persistence (Figure 6.2). For intermediate levels of persistence ( $p \leq 0.5$ ), dual-task training eliminated the PRP entirely. Such “virtually perfect time sharing” has been observed by Schumacher et al. (2001) after training participants extensively on dual-tasking.

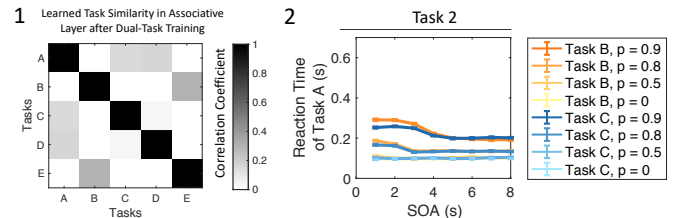


Figure 6: Effects of dual-task training. (1) Average correlations between learned task representations in the associative layer. (2) RT of the second task in the PRP paradigm as a function of persistence  $p$  and task. Error bars show the standard error of the mean across 10 simulated networks.

## General Discussion and Conclusion

One of the most fundamental limitations of human cognitive behavior is the constraint on the number of control-dependent processes that can be executed simultaneously (Posner & Snyder, 1975; Shiffrin & Schneider, 1977). The multiple-resource hypothesis explains such limitations in terms of shared representations that prevent the interference-free execution of multiple control-demanding tasks (Allport, 1980; Navon & Gopher, 1979). While recent neural network studies provided computational and analytic arguments for the detrimental effects of shared representations on the capacity for control-dependent processing, they were either based on the assumption that shared representations always induce conflict or that tasks can only be executed concurrently (Alon et al., 2017; Feng et al., 2014; Musslick et al., 2016).

In this work, we examined the interactive effect of shared representations and two other factors on limitations associated with control-dependent processing. We first demonstrated that the detrimental effect of shared representations on multitasking interference is present only if the tasks that share representations induce a sufficient amount of conflict. This observation extends previous work, showing that performance of single tasks decreases with the amount of conflict induced by a competing task (Cohen et al., 1990; MacLeod & Dunbar, 1988). In both cases, the conflict induced by the competing task scales with the amount of training on that

task. This suggests that training on a task can improve its performance but may come at the cost of inducing interference with another task that shares a representation.

We also demonstrated that the limitations induced by shared representations can extend to situations in which tasks are executed sequentially. The detrimental effect of shared representation scales with the amount of persistence in the network: the more the representation of a task persists in time, the longer it interferes with other tasks. These observations provide a mechanistic interpretation of the psychological refractory period in neural systems. Symbolic architectures explain this effect in terms of a shared resource that can only be accessed by one task at a time (Anderson, 2013; Navon & Gopher, 1979; Meyer & Kieras, 1997a; Salvucci & Taatgen, 2008). In contrast, the neural network model suggests that tasks may always be processed in parallel but that the outcome of a task process may be strategically delayed to prevent interference from persisting representations of previously executed tasks, yielding a PRP.

The neural network model may also have virtue in explaining findings that central processing bottleneck models struggled to explain. For instance, the second task in the PRP paradigm can be prolonged (relative to single task execution) even if the stimulus for the second task was presented after the participant already responded to the first task (Welford, 1952; Marill, 1957). A central processing bottleneck alone cannot account for a delayed execution of the second task in this situation because a bottleneck should no longer be occupied after executing the first task (Pashler, 1994). The neural network model, however, shows that processing interference induced by shared representation with a previously executed task can persist, irrespective of whether a response for that task has already been generated. Furthermore, modality-specific PRP effects have challenged the notion of a domain-general (amodal) central processing bottleneck: Pairs of tasks with compatible stimulus-response mappings (e.g. a visual-manual task paired with an auditory-vocal task) show greater dual-task interference than two tasks with incompatible stimulus-response mappings (a visual-vocal task paired with an auditory-manual task), lending support to cross-talk models that explain dual-task interference in terms of representational overlap between tasks (Liepelt, Fischer, Frensch, & Schubert, 2011; Hazeltine, Ruthruff, & Remington, 2006). Similarly, our simulation results suggest that functional dependence between tasks induced by representational overlap can lead to higher dual-task interference. Finally, empirical work demonstrated that the PRP can be eliminated with dual-tasking practice, suggesting absence of a central processing bottleneck. (Schumacher et al., 2001). The simulation results presented here suggest that dual-task training may promote the learning of separated, task-dedicated representations that promote interference-free processing.

One of the most robust findings in the cognitive literature is the performance cost associated with the sequential execution of different tasks (Alport et al., 1994). One prominent

account of such switch costs is task-set inertia, according to which the task-set of the previously executed task carries over to the next (Alport et al., 1994). Similarly, the findings described here suggest that persistence of task representations lead to a carry over of task interference. The successful sequential execution of two dependent tasks would then afford a temporal switch cost in order to minimize interference-based costs in dual-tasking accuracy. From this perspective, the dependence between tasks induced by shared representation, the amount of conflict, as well as persistence of task representations may all contribute to the performance costs associated with task switches. This suggests that the PRP effect and the costs associated with task switching may originate from the same set of mechanisms in neural systems.

While shared representations may account for limitations in the *number* of control-demanding tasks that can be executed at a time, they do not directly explain limitations in the *amount* of control that can be allocated to a single task (Shenhav et al., 2017). That is, once a commitment has been made to perform a given task (i.e., allocate cognitive control to it), and that precludes the performance of others, then the opportunity cost has already been paid, so why not allocate control maximally to the selected task? Musslick, Jang Jun, Shvartsman, Shenhav, and Cohen (2018) explored the hypothesis that constraints on control intensity (i.e., encoded as cost) reflect, at least in part, an optimal solution to the stability-flexibility dilemma: Allocating more control to a task results in greater activation of its neural representation but also in greater persistence of this activity upon switching to a new task, yielding switch costs. By considering the problem in terms of the parameterization of a nonlinear dynamical system, in which control signals are represented as attractors, Musslick et al. (2018) showed that constraints on the amount of cognitive control allocated to a task can promote cognitive flexibility at the expense of cognitive stability. While this dilemma provides a rationale for why humans should limit the amount of control allocated to a single task it is based on the implicit assumptions that tasks cannot be executed in parallel due to constraints in multitasking capacity and that task representations persist in time. This suggests that both the number of control-demanding tasks that can be executed simultaneously, and the amount of control that can be allocated to a single task may be subject to constraints that arise from (a) the shared use of representation between tasks, (b) the conflict induced by shared representations and (c) persistence of task representations in time.

## References

- Allport, D. A. (1980). Attention and performance. *Cognitive psychology: New directions*, 1, 12–153.
- Alon, N., Reichman, D., Shinkar, I., Wagner, T., Musslick, S., Cohen, J. D., . . . others (2017). A graph-theoretic approach to multitasking. In *NIPS Proceedings* (pp. 2097–2106).
- Alport, D., Styles, E., & Hsieh, S. (1994). 17 shifting intentional set: Exploring the dynamic control of tasks.

- Anderson, J. R. (2013). *The architecture of cognition*. Psychology Press.
- Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychol. Rev.*, *108*(3), 624.
- Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychol. Rev.*, *108*(4), 847.
- Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychol. Rev.*, *97*(3), 332–361.
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.
- Feng, S. F., Schwemmer, M., Gershman, S. J., & Cohen, J. D. (2014). Multitasking vs. multiplexing: toward a normative account of limitations in the simultaneous execution of control-demanding behaviors. *Cogn Affect Behav Neurosci*, *14*(1), 129–146.
- Gilbert, S. J., & Shallice, T. (2002). Task switching: A pdp model. *Cognitive Psychology*, *44*(3), 297–337.
- 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.
- 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.
- 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.
- MacLeod, C. M., & Dunbar, K. (1988). Training and stroop-like interference: Evidence for a continuum of automaticity. *J Exp Psychol Learn Mem Cogn*, *14*(1), 126.
- Marill, T. (1957). Psychological refractory phase. *British Journal of Psychology*, *48*(2), 93–97.
- Meyer, D. E., & Kieras, D. E. (1997a). A computational theory of executive cognitive processes and multiple-task performance: Part I. Basic mechanisms. *Psychol. Rev.*, *104*(1), 3–65.
- Meyer, D. E., & Kieras, D. E. (1997b). A computational theory of executive cognitive processes and multiple-task performance: Part II. accounts of psychological refractory-period phenomena. *Psychol. Rev.*, *104*(4), 749–791.
- Miyake, A., & Shah, P. (1999). *Models of working memory: Mechanisms of active maintenance and executive control*. Cambridge University Press.
- Musslick, S., Dey, B., Özcimder, K., Patwary, M. M. A., 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 Conference of the Cognitive Science Society* (pp. 1547–1552). Philadelphia, PA.
- Musslick, S., Jang Jun, 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 Conference of the Cognitive Science Society* (pp. 829–834). London, UK.
- Navon, D., & Gopher, D. (1979). On the economy of the human-processing system. *Psychol. Rev.*, *86*(3), 214.
- Pashler, H. (1984). Processing stages in overlapping tasks: evidence for a central bottleneck. *J Exp Psychol Hum Percept Perform*, *10*(3), 358.
- Pashler, H. (1994). Dual-task interference in simple tasks: data and theory. *Psychological bulletin*, *116*(2), 220.
- Posner, M., & Snyder, C. (1975). Attention and cognitive control. In *Information processing and cognition: The loyalty symposium* (pp. 55–85).
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533.
- 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).
- Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychol. Rev.*, *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 SIGCHI* (pp. 1819–1828).
- 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.
- 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*, 99–124.
- Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending and a general theory. *Psychol. Rev.*, *84*(2), 127.
- Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: the leaky, competing accumulator model. *Psychol. Rev.*, *108*(3), 550.
- Welford, A. T. (1952). The psychological refractory period and the timing of high-speed performance – a review and a theory. *Br. J. Psychol.*, *43*(1), 2–19.