A role for working memory in shaping the action policy for reinforcement learning

by

Huang H. Ham

A thesis submitted in satisfaction
of the Honors Program option in pursuit of the

Bachelor of Arts

in

Psychology

in the

COLLEGE OF LETTERS AND SCIENCE

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Faculty sponsor in charge:

Professor Anne G.E. Collins

Post-Doc mentor:

Dr. Samuel D. McDougle

May 2020

Abstract

During learning, humans recruit multiple cognitive mechanisms, including value-based reinforcement learning and executive functions, like working memory. Recent research has begun to unmask connections between these two systems, proposing roles for attention and working memory in shaping underlying learning computations. Here, using a simple instrumental learning task, we provide evidence that working memory plays a role in establishing the correct state space that reinforcement learning operates over. We show that reinforcement learning is impaired when executive functioning is taxed by a secondary task and that this effect is especially pronounced when the two tasks are performed simultaneously rather than alternated. Computational modeling suggests that when the executive function is occupied, the reinforcement learning system forms policy over a confused state-space. This study adds to a growing body of research proposing a more fundamental role for high-level executive processes in low-level reinforcement learning computations.

Keywords: Reinforcement learning; Working memory; Executive control; Computational modeling; Dual-Task manipulation

A role for working memory in shaping the action policy for reinforcement learning

Introduction

Human beings possess flexible cognitive tools to adapt to an often capricious world. Even a seemingly simple task of bringing a car to a stop at a red light requires good coordination of different cognitive systems. To accomplish this task, a driver must monitor potentially chaotic surroundings (e.g., checking a bike lane, deducing the intentions of neighboring cars, etc.) while at the same time performing the motor actions to smoothly putting the vehicle to a stop. Moreover, the driver must decide whether these actions are appropriate under the current situation. For example, if the traffic light is red, the action of pressing the brake is appropriate, but if the traffic light is green, pressing the gas pedal may instead be the appropriate action. This task is especially onerous for beginner drivers because they must learn all of these with limited feedback. This simple example illustrates how high-level executive functions (e.g. attention, working memory) and low-level instrumental actions can interact in the performance of everyday skills.

The Computational Framework of Reinforcement Learning

Research on *instrumental learning*, the process of reinforcing actions that lead to rewards, typically focuses on the *reinforcement learning* system (RL), which is best captured by a computational framework that formalizes reward as a numerical value (Rescorla, 1972). The key behavioral signature of instrumental learning is that a learning agent (a human, an animal, or an artificial agent) learns to repeat an action if it has been reinforced by rewards over time. In other words, an agent is more likely to adopt a more rewarded action over less rewarded ones. To capture this behavioral signature, the computational framework of reinforcement learning formalizes reinforcement error signals as the process of comparing an agent's expected reward with the actual reward incurred by the action. The difference between the actual and the expected reward is called the *reward prediction error* (RPE). If the reward prediction

error is positive, then the likelihood of performing the corresponding action in the future will increase; if it is negative, then this likelihood will decrease. To illustrate this framework, we can consider an example.

Suppose Pierre is trying to learn the appropriate action to take at a traffic light. In real life, this learning process involves many cognitive systems, but here let us simplify it to illustrate the process of reinforcement learning alone. Apparently, the goal for Pierre is to learn that he should press the brake pedal if the traffic light turns red, but instead press the gas pedal if the traffic light turns green. Therefore on this issue, Pierre has two potential actions, namely pressing the gas pedal and pressing the brake pedal. Critically, which action is rewarding depends on which stimulus Pierre receives from the traffic light (i.e., green or red). Without loss of generality, let us first consider the case where Pierre is learning he must press the brake when seeing the red light. Suppose initially when seeing the red light, he does not expect any action to be rewarding so he chooses which action to adopt by chance. Once he presses the brake and stops the car, however, he receives an unexpected reward, namely not getting pulled over or safely arriving at the destination. Therefore a positive reward prediction error arises. In order for this positive reward prediction error to influence Pierre's future action, the reinforcement learning framework assumes that Pierre implicitly stores the value of each action (i.e., pressing the brake and pressing the gas pedal). These values keep track of the cumulative, long-term reward received for each action. Conventionally, they are called the *Q values*. Every time Pierre receives a reward due to "pressing the brake", the Q value of that action increases. Similarly, every time he presses the gas pedal at a red light, he receives no reward but rather punishments, yielding a negative reward prediction error. The negative reward prediction error then decreases the Q value of "pressing the gas pedal". Because Q values encode the reward history of each action, they stand as the best guide for Pierre to choose the most historically rewarding action when he sees a red light. Every time he faces the choice of which pedal to press, he implicitly uses the Q values to form a *policy* (i.e. a probability distribution over the candidate actions). The policy would assign a higher probability to the action with a

higher Q value. Overtime, Pierre would have in store a higher Q value for pressing the brake pedal and would more likely choose to do so when seeing a red light. Symmetrically, Pierre is assumed to store another *separate* set of Q values encoding the reward history with respect to green light and guiding his action when he sees the green light instead.

As illustrated in the example, the reinforcement learning algorithm can successfully produce the behavioral signature of instrumental learning as rewarding the action (pressing the brake at a red light and pressing the gas pedal at a green light), gradually increasing the likelihood for Pierre to adopt it. The example also showcases that the general process of the reinforcement learning algorithm has two stages, which are the *value updating stage* and the *policy formation stage*. In the value updating stage, the Q value is iteratively updated by the reward prediction errors of past actions. In the policy formation stage, an action policy is created according to the Q values.

Reinforcement learning is not only a helpful computational framework, but prior studies have also found neurological evidence supporting its biological implementation: Studies have found that some dopaminergic cells in the sub-cortical regions of the brain can compute reward prediction error by substantiating the effect of reward using dopamine (Bayer & Glimcher, 2005; Schultz, Dayan & Montague, 1997). Other studies have also discovered cortico-striatal loops responsible for carrying out various steps of reinforcement learning, further endorsing its biological reality (Alexander, DeLong & Strick, 1986; Haber & Behrens, 2014).

Reinforcement Learning is not Alone

Although reinforcement learning is a powerful learning system, human beings also utilize higher-level executive functions during learning processes, such as *working memory* (WM) which is defined as a brain system that provides temporary storage and manipulation of the information necessary for complex cognitive tasks (Baddeley, 1992). For example, working memory could allow an agent to temporarily store a string of numbers (e.g., like a phone number). Could such higher-level functions contribute to

instrumental learning as well? A growing body of research suggests that executive functions like working memory and attention indeed shape the learning of simple instrumental skills alongside reinforcement learning (Collins & Frank, 2012; Leong, Radulescu, Daniel, DeWoskin & Niv, 2017). This study focuses on extending the line of research on the interplay between reinforcement learning and working memory during instrumental learning.

To dissociate these systems during learning, one approach is to design laboratory computerized tasks that reveal different qualitative learning processes. In one such task, participants are asked to press buttons in response to visually-displayed stimuli to learn a set of stimulus-response (SR) mappings (e.g., apple = button 1; carrot = button 3; etc.). These mappings are learned in different blocks of trials associated with a particular *set size*(i.e., the number of stimulus-response pairs to be learned). The set size manipulation takes advantage of the limited capacity of working memory shown to be around 3 to 5 items (Cowan, 2010). If working memory contributes to instrumental learning alongside reinforcement learning, the performance of the task should rely increasingly more on reinforcement learning if the set size overloads working memory by exceeding its capacity. Since working memory can acquire immediately the correct stimulus-response mapping upon encounter, less working memory involvement would lead to worse performance. Hence the performance of the task would worsen as set size increased beyond working memory capacity. Alternatively, if learning a stimulus-response mapping is mostly accomplished by the reinforcement learning system — which is traditionally defined as a robust, high-capacity learning system — set size effects should be minimal.

Across various populations, studies have shown that working memory and reinforcement learning indeed operate in parallel during simple instrumental learning tasks and compete for action prediction (Collins & Frank, 2012; Viejo, Khamassi, Brovelli & Girard, 2015), primarily exemplified by pronounced set size effects. These findings can be formalized in reinforcement learning models designed to capture human behavioral and neural data (Collins, Ciullo, Frank & Badre, 2017; Collins & Frank,

2018; Viejo et al., 2015).

The Open Question

What is poorly understood, however, is *if reinforcement learning actually relies on executive functions, or if these cognitive processes are functionally independent while contributing to instrumental learning.* Concretely, existing literature provides evidence that the working memory and reinforcement learning systems form policies in separation and the policies are combined into a final policy that determines the agent's action. Nonetheless, it has been called into question whether working memory and other executive functions merely operate separately from reinforcement learning or rather certain stages of reinforcement learning computations rely on executive functions. Some recent research supports the latter idea, suggesting that attentional processes shape reinforcement learning computations (Leong et al., 2017; Niv et al., 2015). Other research points out that working memory influences the formation of reward prediction errors in reinforcement learning (Collins, 2018). However, most previous research does not control for alternative cognitive processes that can appear to mimic reinforcement learning but are not true reinforcement learning (e.g., hypothesis testing, explicit strategies, etc.). Here, we attempted to investigate the influence of working memory on reinforcement learning by attenuating working memory.

We identified *four possible predictions* for how weakening working memory might impact the reinforcement learning system. Firstly, under the hypothesis that the interaction with working memory is weak enough, we predict no noticeable change to the reinforcement learning system occurs. Secondly, the efficacy of reinforcement learning may increase, supporting the hypothesis that working memory has an inhibitory effect on reinforcement learning. Thirdly, under the hypothesis that reinforcement learning relies on working memory to function efficaciously, we predict a decrease in the efficacy of reinforcement learning. Lastly, the interaction between these two systems may be more complicated, including multiple joint effects described previously. Working memory may impede some computational stages of reinforcement

learning, and yet also make essential contributions to other stages.

In attempting to examine these hypotheses, we used a *"dual-task" manipulation* (Baddeley, 1992; D'Esposito et al., 1995; Economides, Kurth-Nelson, Lübbert, Guitart-Masip & Dolan, 2015) to directly tax working memory during instrumental learning. Particularly, the "dual-task" manipulation directly taxes working memory by presenting extra information for the participant to remember while simultaneously performing the learning task. We tested various computational models of working memory and reinforcement learning interactions within this dual-task setting, asking how forcing the executive to multi-task may have affected computations within the reinforcement learning system.

Methods

Participants

Participants (N = 33) were recruited through the University of California Berkeley's SONA platform and earned undergraduate psychology class credit for their participation (26 females and 7 males; mean age = 21.43). Data from a separate single-task only experiment (N = 31) were used for performance bench-marking and are collected through the same platform (see below). No participants were excluded. The experimental protocol was approved by the university's local ethics committee. Written, informed consent was obtained from all participants before their participation.

Experimental Procedure

Participants were seated in front of a computer monitor and had their hands comfortably positioned on a computer keyboard. They then proceeded to the main experiment which was a computerized task written using Psychtoolbox (version 3.0.10 or newer) on Matlab (version R2016a). The main goal for the participants was to learn which key (out of 3 candidate keys) on the keyboard was associated with each stimulus presented on the screen. We used images from (Collins et al., 2017) as stimuli in our task. A more detailed description is in the appendix A

The task had *three phases*: practice, learning, and testing. The practice phase aimed to familiarize the participant with the task, the learning phase was the main chunk of the experiment where the participants attempted to learn the stimulus-response pairs, and the testing phase displayed all the stimuli in the learning phase again in sequence to test learning outcome.

The learning phase consisted of 10 blocks of trials. A trial was the smallest unit of the task where participants saw a stimulus (in this case, an image) presented on the screen and pressed a key in the response. A block was a collection of trials run one after another. Each block was a learning problem independent of the other. At the beginning of each block, the screen presented all the images that the participants had to learn in this block.

The learning phase consisted of two conditions: *Dual-Task* and *Task-Switch*. Within each condition, the *set size* of the instrumental learning task (the main task of interest) was varied among 2, 3, and 6 (Collins & Frank, 2012). That is, in each block participants had to either learn 2, 3, or 6 stimulus-response associations. Stimuli were never repeated across blocks. 12 iterations of each stimulus were interleaved throughout each block. In the Dual-Task condition, two blocks were performed at each set size, and in the Task-Switch condition, one block was performed each at set sizes 3 and 6, and two blocks at set size 2. The last (10th) block always had a set size 2 and trials in the Task-Switch condition, i.e., the easiest type of block. It served as a buffer between the learning and the testing phase and thus it was excluded from all analyses.

In both the Dual-Task and Task-Switch conditions, a secondary task — the *number judgment task* — was performed in addition to the instrumental learning task (Economides et al., 2015). For this task, two numbers were simultaneously displayed side-by-side with varying font size and integer value (e.g., a large font “2” on the left and a smaller font “6” on the right). Participants were asked to make either a “size” or “value” judgment of the number stimuli by pressing a key that corresponded to the position of either the visually larger number (e.g., “2”, or left button) or the higher-value number (e.g., “6”, right button). The particular judgment required (value

versus size) was randomly selected on each trial. Approximately 80% of trials consisted of conflict trials, where the visually larger integer was smaller in value and vice-versa. The specific integers presented were drawn randomly from $[0, 9]$ without repetition.

Critically, within each block, all trials of the two tasks were either simultaneously performed (Dual-Task condition) or alternated (Task-Switch condition). The two conditions were identical in all other aspects. In Dual-Task blocks, the trial structure was as follows: Participants viewed one of the learning stimuli on the screen and two numbers positioned above the stimulus (Figure 1). The numbers were displayed for 0.3 seconds. The learning stimulus was continually displayed either until the participant responded with one of the three possible actions ("j", "k", or "l" with their right index, middle, or ring finger), or if 1.5 seconds had elapsed. If the response designated as correct for that stimulus was made, +1 "points" were displayed on the screen. If an incorrect response was given, 0 points were displayed. If the reaction time exceeded 1.5 seconds, the message "please respond faster" was displayed, and if the response was faster than 0.15 seconds the message "too fast" was displayed. Critically, after receiving feedback for the instrumental learning task, the participant was then asked to make either a "size" or "value" judgment of the previously-displayed numbers ("a" or "d" with their left ring and index fingers, corresponding to the number displayed on the left or right, respectively). Feedback was then given for the number judgment task ("correct", "incorrect", "please respond faster", or "too fast"). An inter-trial interval of 1.5 seconds minus the reaction time then occurred, which consisted of a white fixation cross displayed in the center of the screen. The interval was computed as such to control for total trial duration.

Thus, in Dual-Task blocks, the number task and instrumental task were performed simultaneously — the number sizes and values had to be encoded and maintained while the correct stimulus-response association was being learned and/or retrieved. In contrast, in Task-Switch blocks, the same two tasks were performed, but in succession — a complete trial of the instrumental learning task was performed (learning stimulus, response, feedback), followed by a complete trial of the number judgment task

(number stimuli, response, feedback). The Task-Switch condition was included to benchmark the global effects of taxing executive function without requiring secondary task representations to occupy working memory during the choice and feedback phases of the instrumental task.

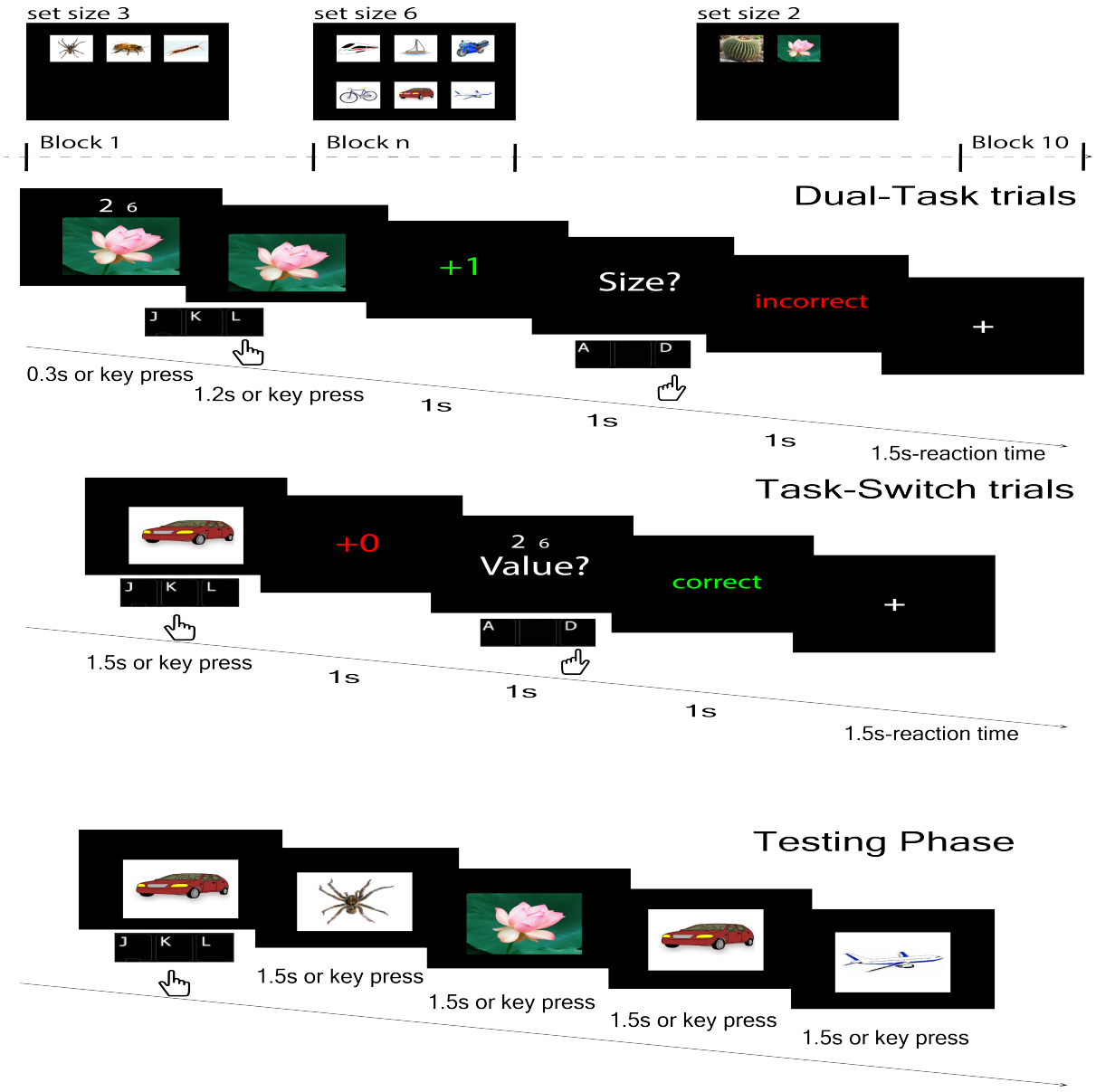


Figure 1. Task Design: Example trial sequence for the Dual-Task and Task-Switch conditions. Each block only corresponds to one of the set size conditions and one of the trial conditions. Testing phase: Each image repeats four times at randomized places in the sequence. No feedback was given.

To become familiarized with the tasks, participants performed the practice phase with four unique practice rounds before the learning blocks began: They first practiced

the instrumental learning task on its own (10 trials), followed by the "number judgment" secondary task on its own (10 trials), then both the Dual-task and Task-Switch tasks (10 trials each). Experimenter instructions emphasized that participants should focus on performing equally well on both tasks in all blocks.

After the learning phase, participants, unknown to them in advance, proceeded to perform a testing phase (Figure 1). In the testing phase, the screen first displayed the instruction telling them that they would see images that they had encountered before, and they had to respond by pressing the key that they originally learned is correct for that image (j, k, or l). Unlike in the learning phase, however, no feedback followed their actions. The testing phase was not divided into blocks, and all the images in the learning block were shuffled and presented in sequence at the center of the screen. Each image would appear four times in total in this shuffled sequence. Similar to the learning phase, participants' response to a trial was valid if made between 0.15 and 1.5 seconds since the onset of the image.

Lastly, for bench-marking purposes, we also present data from a separate group of participants ("single-task"; N = 31) who performed the same instrumental stimulus-response task without any accompanying number dual-task, neither during nor between learning trials. The trial structure for this task thus echoed the Task-Switch condition, but with the secondary task removed. However, the inter-trial interval remained the same length, and thus each trial had a shortened total duration due to the absence of the secondary number task. Participants performed one block of single-task learning at each set size — 2, 3, and 6.

The Basic RLWM Computational Model

Here we present the details of the "RLWM" model architecture, which functions as the foundation of our modeling analyses (Collins & Frank, 2012). The model was designed to fit participants' choices in this instrumental learning task, and capture contributions from working memory and reinforcement learning. The learning of stimulus-action values is modeled using a variant of a typical reinforcement learning

model (Sutton & Barto, 1998). The model relies on two main variables representing the task environment. The first one is the state $s \in S$ where S represents the full stimulus/state space within a block (i.e., all the possible images that could appear). In our experiment, $|S| \in \{2, 3, 6\}$. The second variable is the action $a \in A$ where A is the full action space (i.e., j, k, l). In our experiment, $|A| = 3$ because there were three possible buttons to press as a response to the instrumental learning task. The algorithm proceeds in two stages, as introduced in the introduction: the value updating stage and the policy formation stage. In the value updating stage, for stimulus s and action a on trial t, the model estimates an expected value (i.e., the Q value) $\boldsymbol{Q}(s_t, a_t)$ by performing an update using the delta rule (equation 2; Rescorla, 1972):

$$\boldsymbol{Q}_{t+1}(s_t, a_t) = \boldsymbol{Q}_t(s_t, a_t) + \alpha\delta_t \tag{1}$$

$$\delta_t = r_t - \boldsymbol{Q}_t(s_t, a_t) \tag{2}$$

where α represents the **learning rate** and $\boldsymbol{Q}_t$ represents a $|S| \times |A|$ matrix encoding all Q values given a trial t. $\boldsymbol{Q}_0$ is initialized as a uniform matrix of $\frac{1}{|A|}$. $\delta \in [0, 1]$ is the reward prediction error, and $r \in \{0, 1\}$ is the (binary) reward received. Critically, the model captures the parallel recruitment of working memory (WM) and reinforcement learning (RL) by training two simultaneous learning modules: The reinforcement learning module is described by equation 1. The working memory module is formally similar, but has a learning rate of $\alpha = 1$ (algebraically equivalent to equation 3). For convenience, we analogously name the values stored in the working memory module *W value*. Thus, the working memory delta rule has perfect retention of the outcome of the previous trial with stimulus s_t, reflecting rapid learning of stimulus-response pairs that is qualitatively distinct from classic reinforcement learning. Working memory is also vulnerable to forgetting (Posner & Keele, 1967): The model

captures trial-by-trial decay of W values (equation 4),

$$\boldsymbol{W}_t(s_t, a_t) = r_t \tag{3}$$

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \phi(\boldsymbol{W}_0 - \boldsymbol{W}_t) \tag{4}$$

where $\phi \in [0, 1]$ is the **forgetting parameter** that draws all W values toward their initial values $\boldsymbol{W_0} = \boldsymbol{Q_0}$. The model also captures a positive learning bias (i.e., the neglect of negative feedback) upon negative prediction errors (i.e., $\delta < 0$), the learning rate α is reduced multiplicatively:

$$\alpha^- = \gamma\alpha \tag{5}$$

where $\gamma \in [0, 1]$ controls the **learning bias** (higher values cause less bias toward positive feedback, and lower values cause more). Learning bias occurs for *both the reinforcement learning and working memory modules*; in the latter case, the perfect learning rate of 1 is also scaled by γ.

In the policy formation stage, Q and W values are transformed by the *Softmax function* into a policy, i.e., a vector of probabilities of taking each action. Separate working memory and reinforcement learning policies (represented by *row vectors* $\vec{\pi}_t^{WM}$ and $\vec{\pi}_t^{RL}$) are then combined in the calculation of the final policy via a weighted sum (equation 8),

$$\vec{\pi}_t^{RL} = p(\vec{A}|s_t) = Softmax(\boldsymbol{Q}(s_t), \beta) = \frac{e^{\beta \boldsymbol{Q}(s_t)}}{\sum_{a \in A} e^{\beta \boldsymbol{Q}(s_t, a)}} \tag{6}$$

$$\vec{\pi}_t^{WL} = p(\vec{A}|s_t) = Softmax(\boldsymbol{W}(s_t), \beta) = \frac{e^{\beta \boldsymbol{W}(s_t)}}{\sum_{a \in A} e^{\beta \boldsymbol{W}(s_t, a)}} \tag{7}$$

$$\vec{\pi}_t = w\vec{\pi}_t^{WM} + (1 - w)\vec{\pi}_t^{RL} \tag{8}$$

where $\beta \in [0, \infty)$ represents the softmax temperature and $w \in [0, 1]$ approximates how much working memory contributes to the eventual decision. This value is determined by two free parameters, the **working memory capacity** (i.e., resource limit) $C \in \mathbb{N}$, and the **initial working memory weight** $\rho \in [0, 1]$,

$$w = \rho * min\left(1, \frac{C}{|A|}\right) \tag{9}$$

This equation says that the weight given to the working memory module is reduced if the set size exceeds working memory capacity C, in proportion to the ratio of items that can be held in working memory.

Finally, **un-directed decision noise** ($\epsilon \in [0, 1]$) is added to the final weighted policy (π) to capture the potential noise during the action formation,

$$\vec{\pi}_t \leftarrow \epsilon\left(\frac{1}{|A|}\right) + (1 - \epsilon)\vec{\pi}_t \tag{10}$$

Candidate Interaction Models

The RLWM model as outlined above has been shown to outperform model architectures that only capture a single learning process, such as the ones with reinforcement learning alone (Collins & Frank, 2012). Nevertheless, this model does not posit an explicit interaction between the two systems — while the two systems vie for control of the final decision (equation 8), they do not interact before each system reaches to their own policy. Here, we tested the hypothesis that, by using a dual-task approach, we could identify models that capture potential interactions between the two systems. In this study, we tested three hypotheses using three candidate models of how working memory could affect reinforcement learning:

In the **modulation** model, the taxing of the working memory system is hypothesized to modulate the learning rate (α) of the reinforcement learning system. This effect should be pronounced in the Dual-Task condition versus the Task-Switch condition because in the former the number stimuli are held in working memory as the stimulus-response association is retrieved, a response is made, and the resulting

feedback is observed and integrated. The modulation model is identical to the RLWM model, but the RL learning rate α is allowed to freely vary in each condition, yielding two separate learning rate parameters, α^{DT} and α^{TS}.

In the **noise** model, the taxing of the working memory system by the secondary task is hypothesized to simply add noise to the decision process — for example, through attentional lapses — but not change the underlying learning process. This is accomplished by allowing the ϵ parameter to vary freely between the Dual-Task and Task-Switch conditions, i.e. ϵ^{DT} and ϵ^{TS}.

In the **action confusion** model, the taxing of the working memory system is hypothesized to partially confuse the action formation stage of the reinforcement learning system as to at what state (i.e., the stimulus dimension of the stimulus-response association) the agent is acting on a given trial. That is, the action confusion model says that the policy taken on the current trial ($\vec{\pi}_t^{RL}$) will not only be formed using the appropriate current state/stimulus s_t, but *also* using all additional unseen states/stimuli,

$$\vec{\pi}_t^{RL} = \lambda \underbrace{Softmax\left(\frac{\sum_{s \neq s_t, s \in S} \boldsymbol{Q}(s)}{|S| - 1}, \beta\right)}_{\text{confusion term}} + (1 - \lambda) \underbrace{Softmax(\boldsymbol{Q}(s_t), \beta)}_{\text{main term}} \tag{11}$$

where s_t is the stimulus of the current trial. An additional free parameter, $\lambda \in [0, 1]$, weights what we call the *confusion term*, and $1 - \lambda$ weights the *main term*. We call λ the **confusion parameter**. In words, the equation is saying that to obtain the reinforcement learning policy, two terms are computed. The main term is the same as equation 6 and the confusion term substitutes the $\boldsymbol{Q}(s_t)$ in the main term with the *average* Q value of all stimuli in the block that are not the stimulus of the current trial. Then, the reinforcement learning policy is the weighted sum of these two terms. The λ parameter also freely varies in each condition, yielding λ^{DT} and λ^{TS}. Thus, in the action confusion hypothesis, the taxing of working memory should lead the Q values from other stimuli to interfere with the policy formation of the current correct stimulus. This captures the idea that executive functions may help guide reinforcement learning

to choose appropriate actions in particular contexts. The model schematic of the action confusion model is illustrated in Figure 2.

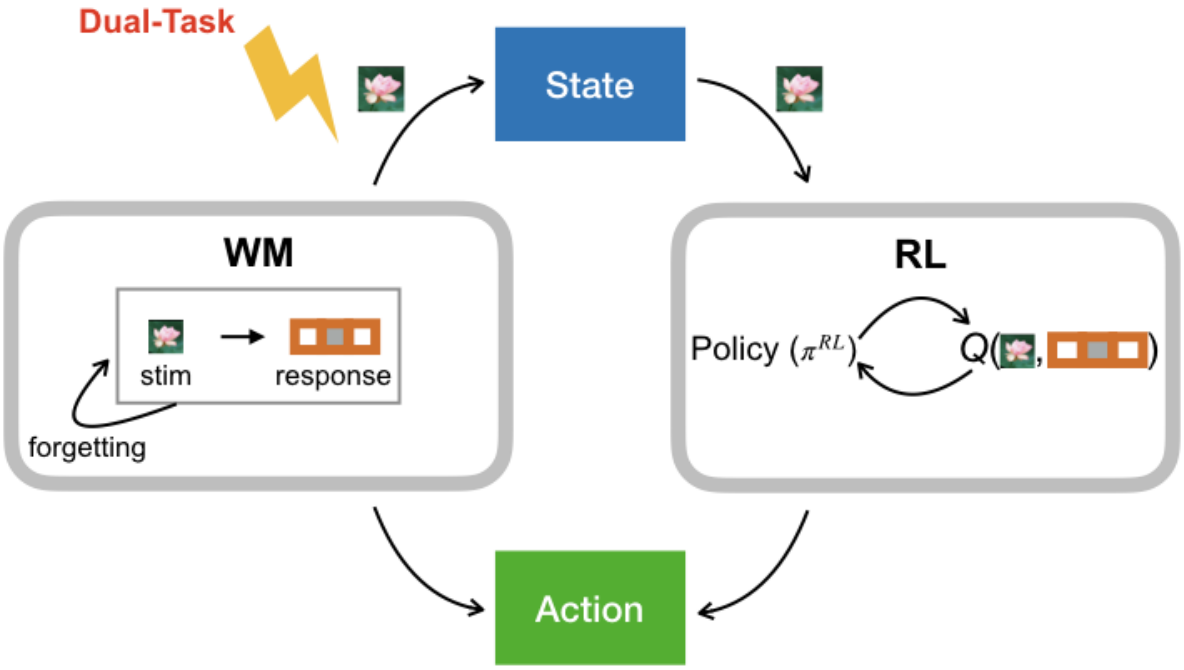


Figure 2. Model Schematic: In the action confusion model, one role for working memory (WM) is to enhance state information in the reinforcement learning (RL) system. In our model, taxing working memory with a dual-task interferes with this process during action formation, leading to confusion over irrelevant states.

We tested two additional benchmark models: The **none free** model is simply the RLWM model described in equations 1 - 10, where no parameters vary between the Dual-Task and Task-Switch conditions. Therefore it has 6 parameters. This model predicts that the taxing of working memory either within (Dual-Task) or between (Task-Switch) trials are functionally equivalent in terms of the effect on the instrumental learning task. In the **all free** model, all parameters except the capacity parameter in the RLWM model could vary between conditions. Therefore it has 11 parameters. This model serves as a benchmark with maximum degrees of freedom.

Finally, all 5 models characterize the taxing of working memory via the secondary number task as a "filling" of one working memory "slot" in the Dual-Task condition relative to the Task-Switch condition. That is, in the Dual-Task condition, $C^{DT} = C^{TS} - 1$, and in the Task-Switch condition, $C^{TS} = C$.

Modeling Procedure

The modeling followed five steps: model fitting, model comparison, parameter recovery, model recovery, and model simulation and validation (Wilson & Collins, 2019). Models were fit to participants' choices using maximum likelihood estimation, by minimizing the negative log posterior using the python function *minimize* in package *scipy*. Parameter constraints were defined as follows: $\alpha, \gamma, \phi, \rho, \epsilon, \lambda \in [0, 1]$ and $C \in \{2, 3, 4\}$. Initial parameter values (except C) were randomized within their constraint across fitting iterations. Because C has a discrete value space, each possible value in the constraint were fitted to choose the best one. 20 iterations were used per possible C value (hence 60 in total) to avoid local minima. Inverse temperature β was fixed at 50 for all fits and simulations. Parameter recovery was performed to measure the identifiability of our models. Model comparisons were conducted using both the Akaike Information Criterion (AIC) (Akaike, 1974) and Bayesian Information Criterion (BIC) (Schwarz, 1978). Model recovery was performed as a check for the validity of our model comparison criteria. Model simulation and validation are performed to ensure that the model's key parameter value correlates with key behavioral features of the data, and that the models' learning behavior reproduces a qualitative pattern similar to that of human participants. Model simulations were conducted by simulating the model using each participant's best-fit parameters and their actual observed sequence of stimuli and blocks. Model simulations were performed 100 times per subject and averaged.

Specifically, We were interested in validating two main behavioral signatures. One was the response accuracy of the instrumental learning task in learning phase and testing phase. Second was the average amount of *action bias*. The first was self-evident, so here let me explain how we define action bias and why it is an important behavorial signature to validate. Our hypothesis suggests that the incorrect state-space interfere with policy formation; that is, we predict that when an action is rewarded in one state, its probability of later being selected in another state, where it is actually the incorrect action, should increase, and *vice versa* when it is not rewarded. Hence we performed a *post-hoc* analysis to test this intuition: We computed an *action bias*, defined by the

proportion of error trials where the action taken had a higher running reward rate (in all previous trials at all states) versus the other alternative erroneous action. For example, if the subject made a mistake in the current trial, there are two possibilities. Suppose the correct key for the current stimulus is "j", then she may err by pressing "k" or "l". Each of the two erroneous keys is associated with a reward rate. Suppose the subject has pressed "k" 10 times in total (in response to whatever stimuli in the block) before the current trial, and is rewarded 5 times out of 10. Then the reward rate of action "k" in the current trial is $\frac{5}{10} = 0.5$. Similarly, we can compute the reward rate of action "l". Say it is 0.3. In this example, we observe that "k" has a larger reward rate, so if the subject errs by pressing "k", she is considered to commit an action bias in this trial, encoded by "1". If the subject errs by pressing "l", (i.e. the action with a smaller reward rate), an action bias is not present and thus is encoded by "0". In trials where the subject is correct or both erroneous actions have the same reward rate, the action bias is encoded as NA. Notice on average, the chance value of this bias is 0.5, where the subject randomly choose a wrong action out of two and is not systematically biased to choose the one with a higher reward history. If our hypothesis is true, however, a systematic bias should exist because if an action has a higher reward rate, it would have a higher average Q value across stimulus. Due to action confusion, all Q values influence policy formation in the form of average (equation 11), and thus the action with a higher average Q value tends to have a higher probability of being chosen.

Results

The analysis consists of two parts: *behavioral analysis* and *computational modeling*. In the behavioral analysis, we fit a mixed-effect logistic regression model to examine the contribution of task conditions in predicting the performance of the instrumental learning task. With computational modeling, we fit the data to various computational agents with different learning parameters which provide insights into the mechanism of the instrumental learning process. Here we report the modeling result from the 5 candidate models reported in the method section. We will summarize briefly

other under-performing computational models we attempted in the appendix B.

Behavior analysis

Participants showed clear signs of learning the stimulus-response mappings in both the Dual-Task and Task-Switch conditions. The probability of getting a trial correct increased with the number of stimulus appearances (Figure 3). Furthermore, there was a strong negative effect of the dual-task on learning: Learning was markedly weaker in the dual-task experiment when compared to a separate group of participants who performed the instrumental learning task alone (Figure 3, dashed lines; all $t > 2.38, p < 0.05$, in t-tests comparing performance to the single-task group for all set sizes and in both the Dual-Task and Task-Switch conditions).

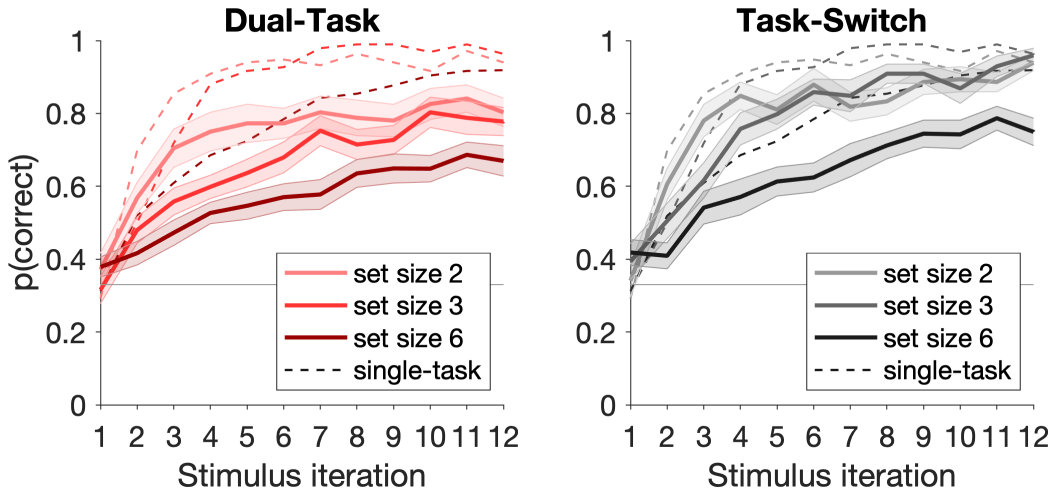


Figure 3. Learning Curves: Participants learned stimulus-response associations over time. Curves reflect the proportion of correct responses as a function of the number of times that each stimulus was presented, plotted separately for each set size. Dashed lines reflect mean learning curves from a separate group of participants performing the stimulus-response learning task without any secondary task (note, these data are identical on each panel). Error shading reflects the s.e.m.

To model the impact of different task variables on performance, we performed a *mixed-effect logistic regression* analysis using R function *glmer* in package *lme4*, with correct/incorrect responses as the outcome variable and subject identification number as the random intercept. All predicting variables were scaled before being passed into the regression model and no interaction term was included. For the learning phase data,

we passed in four task variables as predictors: *condition, set size, delay, and cumulative reward* (see below). Consistent with our predictions, we observed a significant effect of condition ($z = -6.03, p < 0.001$), with participants performing worse on the learning task in the Dual-Task condition versus the Task-Switch condition. This result supports our prediction that performing the secondary task while concurrently retrieving and/or integrating reward feedback of the stimulus-response associations (Dual-Task) would have a stronger negative effect on learning relative to a situation where the secondary task is performed in between trials (Task-Switch).

In our regression analysis we also tested the effect of set size on performance — if working memory is recruited in this task, increasing the set size should decrease performance because holding more stimulus-response associations in mind across trials should become harder. We also analyzed the effect of cumulative reward for each stimulus in the regression model, obtained by adding all the points rewarded to each stimulus up to each trial. If reinforcement learning is incrementally increasing the value of the correct action associated with each stimulus, then performance should increase with the number of previous trials in which a stimulus has been rewarded. Replicating previous results, we observed both a significant negative effect of set size on performance ($z = -5.65, p < 0.001$) and a significant positive effect of cumulative reward on performance ($z = 32.92, p < 0.001$), likely reflecting, respectively, the influences of working memory load and trial-by-trial reinforcement learning in this task.

The regression model also tested the effect of "delay" on performance, captured by the number of trials passed since the last time a particular stimulus was observed and correctly responded to. We observed a significant negative effect of this trial-based delay ($z = -7.21, p < 0.001$), suggesting that short-term forgetting occurs during the task (a result which is also consistent with the recruitment of working memory). Lastly, In terms of performance on the secondary task (number judgment), participants performed well in both conditions, showing a mean performance of 82.7% in the Dual-Task condition and 82.9% in the Task-Switch condition ($t(32) = -0.12, p = 0.90$).

For the testing phase data, we passed in three task variables as predictors:

condition, *set size*, and *asymptotic learning phase performance*. We obtained the condition and set size of the stimuli presented in the testing phase by looking up the condition and set size they had belonged to in the learning phase. The asymptotic rate of the correctness of each stimulus is obtained by computing the average correctness of the last 3 trials of that stimulus from the learning phase. We observed a significant positive effect of set size ($z = 1.97, p < 0.05$), replicating seemingly counter-intuitive previous findings (Collins, 2018): This result suggests that when set size is low and working memory is contributing the lion's share to learning, long term memory is actually hindered; conversely, when the set size is higher and reinforcement learning contributes more to learning, long-term memory is improved. Thus, the testing phase acts as a proxy for the strength of stimulus-response associations learned via the reinforcement learning system.

For the same reason, we would expect participants to perform better in the testing phase on stimuli from the Dual-Task condition where working memory is directly taxed, versus the Task-Switch condition. Contrary to this expectation, however, we found that participants performed much worse on trials with stimuli from the Dual-Task condition ($z = -5.03, p < 0.001$). This key result implies that directly blocking working memory seems to impair the performance of the reinforcement learning system as well, leading to a decreased accuracy of testing phase responses. We return to this point in the next section.

The asymptotic rate of correctness, expectedly, positively predicts the performance in the testing phase ($z = 5.70, p < 0.001$). This result gives more assurances that participants perform better on trials with stimuli that were well learned in the learning phase.

Computational Modeling

We next used computational modeling to identify the processes underlying the decrease in performance in the Dual-Task condition compared to the Task-Switch condition. In particular, can the Dual-Task condition reveal if working memory and

reinforcement learning interact, or if they operate independently? Our candidate computational models were designed to test this question.

Model Comparison. Critically, the RLWM model variant where all parameters were allowed to freely vary between the Dual-Task and Task-Switch conditions ("all free") showed the weakest fit to the data when penalized for complexity (Figure 6a). This likely reflects over-fitting and suggests that underlying learning mechanisms between the conditions remained mostly similar. However, the behavior in the two conditions was significantly different (Figure 3); consistent with that our observation of the similarly weak performance of the "none free" model, where the two conditions share all parameters.

One reasonable hypothesis is that taxing working memory in the Dual-Task condition would not affect reinforcement learning, but would lead to noisier decision-making (e.g., through attentional lapses), and thus worse performance. This intuition is reflected in the "noise" model, where un-directed decision noise (equation 10) is allowed to vary between conditions, but the underlying learning process is unchanged. This model was outperformed as well, suggesting that noisier decision-making cannot explain our results.

Another reasonable hypothesis is that taxing working memory simply changes the learning rate α of the reinforcement learning system (equation 1), as captured by the modulation model. Nonetheless, the model comparison shows that this model is outperformed as well.

The model that best fit the data was the "action confusion" model (Figure 6a; all comparisons to this model, all $p < 0.03$). This model reflects the hypothesis that working memory helps specify the proper dimensions for reinforcement learning to form an action policy over — here, interference of the wrong states during policy formation leads to diminished learning (equation 11). We captured the amount of interference using the confusion parameter λ, which is larger in the Dual-Task condition versus the Task-Switch condition ($t(32)=2.71$, $p = 0.01$). This difference is consistent with increased action confusion when irrelevant stimuli from the secondary number task have

to be maintained in working memory while the learning task is being performed. The full set of fitted parameter values of this model is reported in Table 1.

Table 1
Action Confusion Model Parameter Value

Parameters	mean	standard error	frequency (out of 33)
α	0.09	0.03	-
ϕ	0.29	0.03	-
ρ	0.66	0.04	-
ϵ	0.006	0.004	-
γ	0.10	0.02	-
λ^{TS}	0.20	0.04	-
λ^{DT}	0.30	0.05	-
$C = 2$	-	-	8
$C = 3$	-	-	10
$C = 4$	-	-	15

Parameter and Model Recovery. We performed two procedures to check the validity of the model comparison result. First, we wanted a sanity check on our model-fitting procedure to make sure it reasonably produces valid parameter values. Therefore we performed *parameter recovery* where we simulate a data set using the fitted parameters of the action confusion model and fit it with the exact same model to check whether we can recover the fitting parameter values. The result shows a good recovery outcome in most but not all parameters (Figure 4). The recovery of α, ϕ, ϵ looked suboptimal, but it may be due to noise generated by outliers where the fitting procedure hit a local minimum at the boundary (0 or 1). Critically, the recovery of the confusion parameter λ is really good. Overall, the recovery gives us more confidence that if there is a true parameter for the action confusion model, our fitting procedure can approximate it.

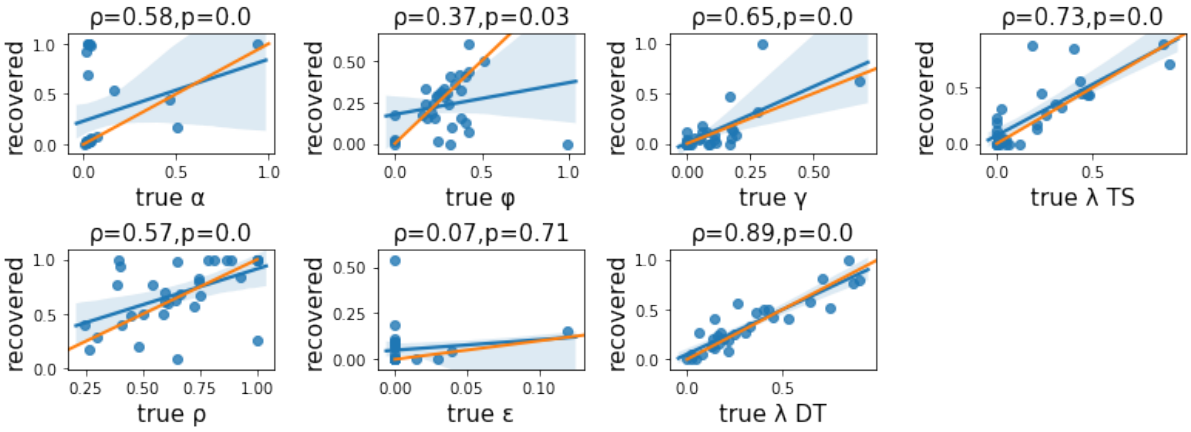
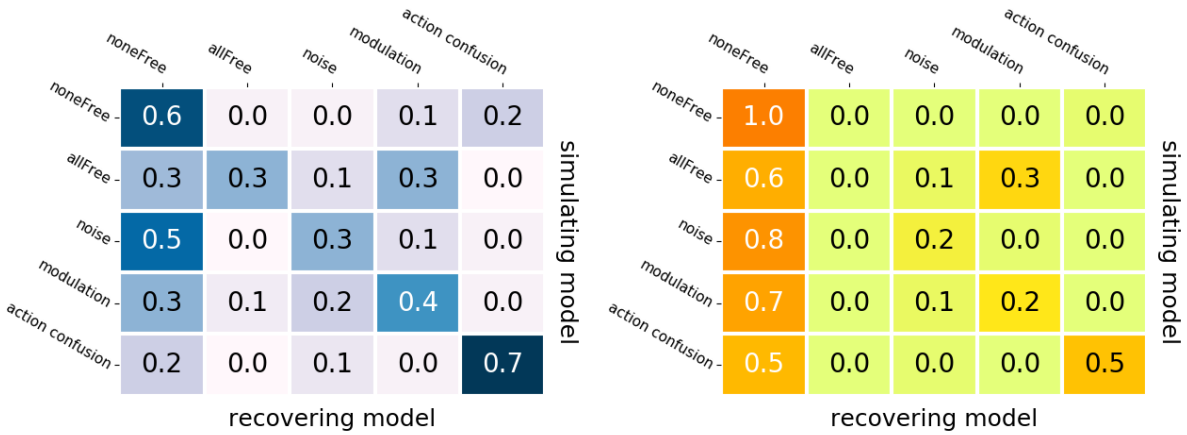


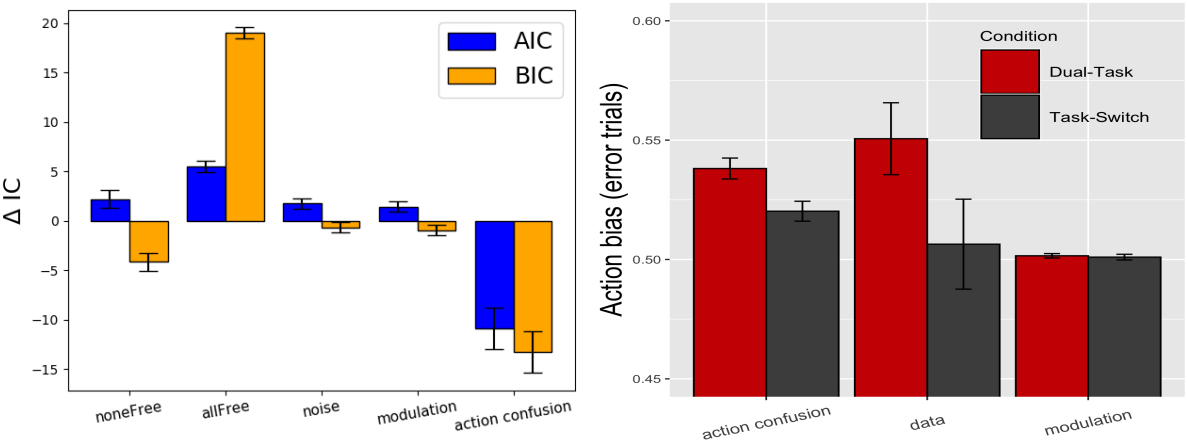
Figure 4. Parameter Recovery: 33 data are generated using the fitted parameter of the action confusion model. Then we again use the same action confusion model to try to recover these parameters. The capacity parameter is fixed to be the best-fitted one. The number of fitting iteration is 20. The recovery result of all seven continuous parameters is presented here. The orange line represents the line of $y = x$ and the Spearman correlation is printed in the title.

Second, we need to check whether Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) can effectively distinguish our models. To achieve this, we adopted the non-parametric procedure of model recovery where we simulate data from all 5 candidate models using their best-fitted parameter from each participant, and fit all 5 models to those simulated data. If AIC and BIC can effectively compare them, we would expect for each data, the model who simulated it should in most cases have the lowest AIC and BIC value because it is the true model for that data. Here we present the result using two confusion matrices — one for AIC and one for BIC. We see that AIC overall does a better job in recovering the fitting model (Figure 5a). BIC seems to over-penalize for model complexity (Figure 5b). However, BIC has 0 false-positives for our best-fitting model, i.e., it does not falsely select the action confusion model as the best model if the data were generated by any other candidate model. Therefore, we decided to include both measures. The fact that the action confusion model outperforms according to both AIC and BIC seems to strongly suggest that the action confusion model is the best model (figure 6a).



(a) Proportion of Samples with Lowest AIC (b) Proportion of Samples with Lowest BIC

Figure 5. Model Recovery: Each model attempted to recover data simulated by each model using each subject's best fitted parameters. The number of fitting iteration is the same as in fitting procedure.

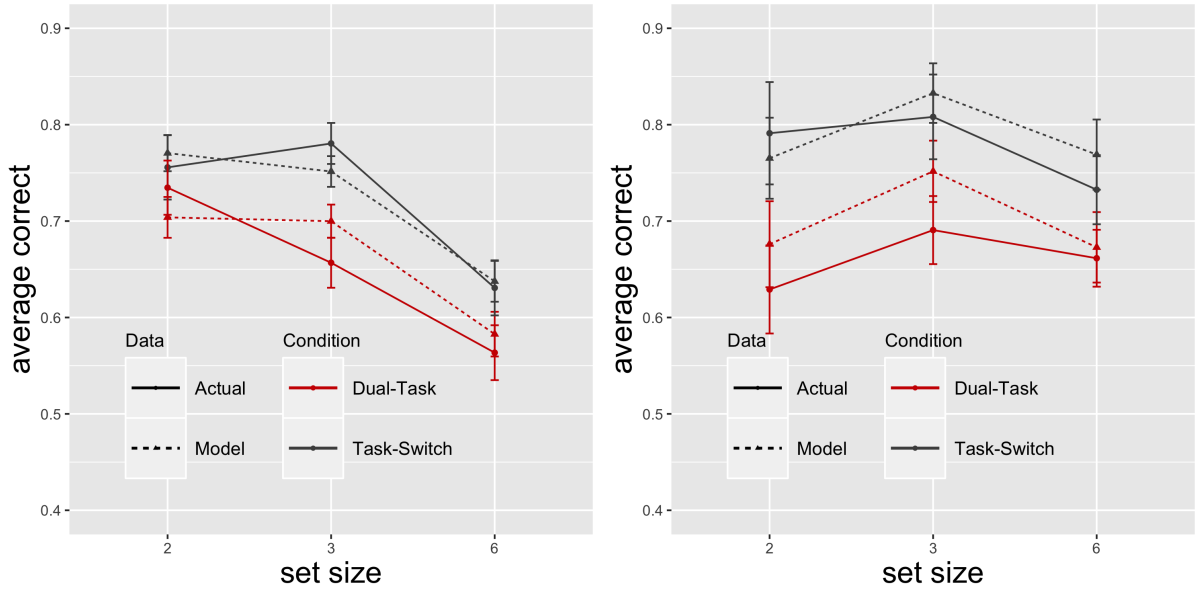


(a) Model Comparison (b) Action Bias

Figure 6. Model Performance: a): models were compared using the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). For an illustration of within-participant statistics, A(B)IC values are plotted corrected by each subject's mean A(B)IC across models. Lower values reflect a better fit. b): Action bias behavior and modeling. Action biases reflect participants' tendencies to be more likely, on error trials, to bias actions that have previously been rewarded at a higher rate. Error bars reflect the s.e.m.

Model Simulation and Validation. Lastly, We also simulated our best-fitting model to try to capture qualitative behavioral signatures in the data that our hypothesis would predict. First, we attempted to validate the learning accuracy. As shown in Figure 7a, the model was able to capture set size effects in the task, as well as the qualitative differences between the Dual-Task and Task-Switch scenarios (we note,

however, that the model did not closely capture the gap between the two conditions at set size 3). The model also does a decent job in validating the qualitative trend in the testing phase, although it tends to overestimate the performance in the Dual-Task condition in the lower set sizes (Figure 7b). We will discuss these limitations with more detail in the discussion section.



(a) Learning Validation (b) Testing Validation

Figure 7. Model Validation: validating that the action confusion model can qualitatively produce the average learning performance across set sizes in the Dual-Task and Task-Switch conditions. Error bars reflect the s.e.m.

Second, we tried to validate the action bias. On error trials, participants did show significant biases towards the erroneous action that had a higher running reward rate in the Dual-Task condition ($t(32) = 3.36, p < 0.01$; Figure 6b). Consistent with our predication, participants did not show such a significant bias in the Task-Switch condition ($t(32) = 0.34, p = 0.74$). The bias rates were marginally different between conditions ($t(32) = 1.83, p = 0.07$).

Simulations of the action confusion model predicted, qualitatively, a numerical difference of action biases between the two conditions (Figure 6b), consistent with the data. However, the model did appear to overestimate the bias rate in the Task-Switch condition, making it significantly above chance level. Critically, the second-best performing model, the "modulation" model, did not predict any bias effects. As a

caveat, the result of simulated data may be contingent on certain arbitrary factors like the number of simulations.

We did observe a significant positive correlation between the participants' action biases and the λ parameter in both the Dual-Task (Spearman's $\rho = 0.63, p < 0.001$) and the Task-Switch conditions (Spearman's $\rho = 0.37, p = 0.03$). This is consistent with that parameter indexing action confusion. However, the co-variation was weaker in the Task-Switch condition (Figure 8). Taken together, these results suggest that people display behavioral signatures consistent with some amount of action confusion occurring over an incorrect dimension when facing dual-task interference.

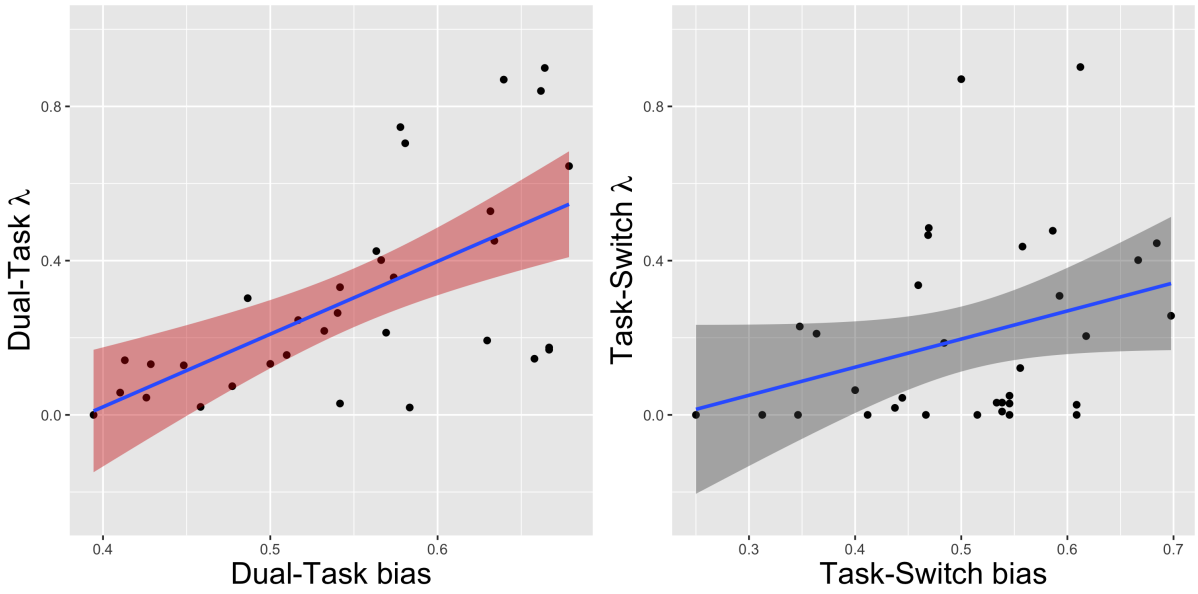


Figure 8. Parameter-Bias Correlation: The λ value is plotted against the action bias (averaged within subject) for all subjects.

Discussion

To successfully navigate the world, we must not only learn which actions are appropriate, but also in which particular situations to perform them. For example, pressing the gas pedal when driving is appropriate in situations where the traffic light shows green, but not so if it shows red, where pressing the brake is appropriate. Reinforcement learning (RL) provides a powerful framework for understanding how agents learn action-outcome contingencies. Here, we suggest that in the common

scenario where state-action-outcome contingencies must be learned, executive functions like working memory (WM) may help the reinforcement learning system form the policy appropriate to each state, or context (Figure 2). Recent work supports this idea in the domain of attention — when different features of a stimulus must be attended to in a learning task (i.e., color versus shape), attentional processes guide the reinforcement learning system while it specifies policies to different stimulus features (Leong et al., 2017; Niv et al., 2015). This result suggests that executive functions and reinforcement learning are not necessarily separate modules that are active during learning but may be closely intertwined.

Here, by separately modeling working memory and reinforcement learning, we could ask how maintaining stimuli in mind across time helps carve out the state space for reinforcement learning to form a policy over. Using a dual-task manipulation during a simple stimulus-response learning task, we revealed behavioral and computational results in support of our hypothesis. Behaviorally, we found that taxing the working memory system during instrumental learning trials (Dual-Task) is more detrimental to performance than taxing working memory between instrumental learning trials (Task-Switch). This result suggests that working memory and reinforcement learning share cognitive resources in the service of instrumental learning. Moreover, participants showed choice biases towards actions that were inappropriate for the current state but had been successful in previous trials at other states (Figure 6b). Consistent with the behavioral results, our modeling suggests that when the executive function is taxed, the reinforcement learning system suffers from state-space confusion when forming policy.

In sum, our result provides evidence against the first two predictions outlined at the end of the introduction section because clearly, blocking working memory has a negative effect on the efficacy of reinforcement learning. However, our result does not disentangle the third from the fourth prediction as our evidence only reveals an interaction at the policy formation stage, so future investigation is needed to examine whether working memory interacts with other stages of reinforcement learning computation and if so, in what fashion.

Limitations and Future Directions

A few limitations of our task design are worth mentioning. First, although the Task-Switch condition is a good control for the Dual-Task condition, participants had a shorter time between seeing the numbers and having to respond to the questions about the numbers than in the Dual-Task condition. It could be the case for some participants that the time allowed for making a valid response to the number task is too short due to the late onset of showing the numbers. Second, to fit the experiment into an hour, we designed the task so that the Dual-Task condition had twice as many blocks as the Task-Switch condition. The Task-Switch condition thus may fall short on statistical power due to the reduced amount of data. Third, because the testing phase is only one block away from the learning phase, the duration in between may not be sufficient for the information in working memory to fully fade away.

Important open questions remain given our current results. First, while we observed that participants' performance in the Task-Switch condition was superior to the Dual-Task condition, their performance in the Task-Switch condition was weaker than that of participants in the Single-Task condition (Figure 3). This suggests that switching between the instrumental learning task and the secondary task did not leave instrumental learning unharmed. Future research could ask how, for instance, task-switching may disrupt between-trial consolidation of stimulus-response associations.

Second, we note another interpretation of our action biasing results (Figure 6b). Specifically, participants could use explicit hypothesis-testing or heuristics to "trim" the action space as they learn. In set sizes 2 and 3 this would be easiest, as we designed the task such that, within a block, one action in each of those set sizes was incorrect for all stimuli. We indeed discovered that the action bias is significantly larger in lower set sizes than in set size 6 ($t(95.894) = -9.49, p < 0.001$). However, we contend that learning and applying such a heuristic would likely require additional cognitive resources, and would thus be interfered with by our dual-task. Indeed, the observed biases were actually stronger in the Dual-Task condition than the Task-Switch

condition (albeit only marginally; Figure 6b), where the load should be heavier in the former. We still spotted a significant correlation between λ^{DT} and action bias in the Dual-Task condition even when set size is 6 (Spearman's $\rho = 0.51, p = 0.002$). Given this alternative explanation, these specific results should be interpreted with caution. Future experiments could more directly test if the observed biases are driven by implicit or explicit processes.

Third, the model validation remains imperfect. For example, the model failed to reproduce the set size effect between set size 2 and set size 3 in the Dual-Task condition and overestimates the accuracy of Dual-Task trials in the testing phase. We have attempted to allow the capacity discount in the Dual-Task condition to vary between 1 and 2 and be fitted for each participant, but it did not solve the problem. Hence in the fitting procedure, we reported here, it is fixed to be 1. Future work is needed to investigate why the action confusion model does not reproduce this effect.

Fourth, neuro-imaging studies need to follow up to investigate the potential underlying neurological processes where the executive functions interact with the sub-cortical reinforcement learning process. Is there a pathway through which the executive functions provide high fidelity information of the state-space with the reinforcement learning system? What happens neurologically when the working memory is overloaded which leads to policy confusion in reinforcement learning? Neurological evidence is crucial to safeguard the reliability of our findings.

Does reinforcement learning need working memory to properly learn at all? While our behavioral and modeling results suggest that working memory can help the reinforcement learning system learn efficiently, our results do not imply that working memory is necessary for effective reinforcement learning to occur at all. Indeed, even animals with limited (if any) executive functions, like fruit flies, show reward-based learning from predictable sensory cues (Tempel, Bonini, Dawson & Quinn, 1983). Arguably, humans are unique in their capacity for rapid, sometimes single-trial learning in novel environments. We propose that interactions between low-level reinforcement learning representations and high-level cognitive representations may be one mechanism

supporting these superlative learning abilities.

Acknowledgments

We thank Helen Lu for help with data collection. We thank the CCN lab, Prof. Serena Chen, Jessica Jones, and Peter Soyster for helpful suggestions and comments. This material is based upon work supported by the National Institute of Mental Health, funded by grant NIH 1R01MH119383-01.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. doi:10.1109/TAC.1974.1100705

Alexander, G. E., DeLong, M. R. & Strick, P. L. (1986). Parallel Organization of Functionally Segregated Circuits Linking Basal Ganglia and Cortex. *Annual Review of Neuroscience*, *9*(1), 357–381. doi:10.1146/annurev.ne.09.030186.002041

Baddeley, A. (1992). Working memory. *Science*, *255*(5044), 556–559. doi:10.1126/science.1736359

Bayer, H. M. & Glimcher, P. W. (2005). Midbrain Dopamine Neurons Encode a Quantitative Reward Prediction Error Signal. *Neuron*, *47*(1), 129–141. doi:10.1016/j.neuron.2005.05.020

Collins, A. G. (2018). The Tortoise and the Hare: Interactions between Reinforcement Learning and Working Memory. *Journal of Cognitive Neuroscience*. doi:10.1162/jocn_a_01238

Collins, A. G., Ciullo, B., Frank, M. J. & Badre, D. (2017). Working Memory Load Strengthens Reward Prediction Errors. *The Journal of Neuroscience*, *37*(16), 4332–4342. doi:10.1523/JNEUROSCI.2700-16.2017

Collins, A. G. & Frank, M. J. (2012). How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, *35*(7), 1024–1035. doi:10.1111/j.1460-9568.2011.07980.x

Collins, A. G. & Frank, M. J. (2018). Within- and across-trial dynamics of human EEG reveal cooperative interplay between reinforcement learning and working memory. *Proceedings of the National Academy of Sciences*, *115*(10), 2502–2507. doi:10.1073/pnas.1720963115

Cowan, N. (2010). The Magical Mystery Four: How Is Working Memory Capacity Limited, and Why? *Current Directions in Psychological Science*, *19*(1), 51–57. doi:10.1177/0963721409359277

D'Esposito, M., Detre, J. A., Alsop, D. C., Shin, R. K., Atlas, S. & Grossman, M. (1995). The neural basis of the central executive system of working memory. *Nature*, *378*(6554), 279–281. doi:10.1038/378279a0

Economides, M., Kurth-Nelson, Z., Lübbert, A., Guitart-Masip, M. & Dolan, R. J. (2015). Model-Based Reasoning in Humans Becomes Automatic with Training. *PLOS Computational Biology*, *11*(9), e1004463. doi:10.1371/journal.pcbi.1004463

Haber, S. N. & Behrens, T. E. (2014). The Neural Network Underlying Incentive-Based Learning: Implications for Interpreting Circuit Disruptions in Psychiatric Disorders. *Neuron*, *83*(5), 1019–1039. doi:10.1016/j.neuron.2014.08.031

Leong, Y. C., Radulescu, A., Daniel, R., DeWoskin, V. & Niv, Y. (2017). Dynamic Interaction between Reinforcement Learning and Attention in Multidimensional Environments. *Neuron*, *93*(2), 451–463. doi:10.1016/j.neuron.2016.12.040

Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A. & Wilson, R. C. (2015). Reinforcement Learning in Multidimensional Environments Relies on Attention Mechanisms. *Journal of Neuroscience*, *35*(21), 8145–8157. doi:10.1523/JNEUROSCI.2978-14.2015

Posner, M. I. & Keele, S. W. (1967). Decay of Visual Information from a Single Letter. *Science*, *158*(3797), 137–139. doi:10.1126/science.158.3797.137

Rescorla, R. (1972). A theory of Pavlovian conditioning : Variations in the effectiveness of reinforcement and nonreinforcement.

Schultz, W., Dayan, P. & Montague, P. R. (1997). A Neural Substrate of Prediction and Reward. *Science*, *275*(5306), 1593–1599. doi:10.1126/science.275.5306.1593

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, *6*(2), 461–464. doi:10.1214/aos/1176344136

Sutton, R. S. & Barto, A. G. (1998). *Introduction to reinforcement learning*. MIT press Cambridge.

Tempel, B. L., Bonini, N., Dawson, D. R. & Quinn, W. G. (1983). Reward learning in normal and mutant Drosophila. *Proceedings of the National Academy of Sciences*, *80*(5), 1482–1486. doi:10.1073/pnas.80.5.1482

Viejo, G., Khamassi, M., Brovelli, A. & Girard, B. (2015). Modeling choice and reaction time during arbitrary visuomotor learning through the coordination of adaptive working memory and reinforcement learning. *Frontiers in Behavioral Neuroscience*, *9*. doi:10.3389/fnbeh.2015.00225

Wilson, R. C. & Collins, A. G. (2019). Ten simple rules for the computational modeling of behavioral data. *eLife*, *8*, e49547. doi:10.7554/eLife.49547

Appendix A

Image sets used as stimuli

The instrumental learning tasks require the participants to learn the correct key mappings of each stimulus. In our design, each stimulus is an image. But where are these images come from? We prepared 19 image sets (Collins et al., 2017). Each image set has 6 images, all belonging to a particular category. The categories covered are plants, vegetables, dining utensils, cartoon characters, geometric shapes, colors, sports activities, facial expression icons, hand-labor tools, animals, western instruments, transportation vehicles, fruits, natural sceneries, items of clothing, historical sites, interior parts of a house, arthropods, and symbols on the keyboard.

10 image sets (i.e. 10 categories) were randomly drawn and assigned to each block of the task for each participant. Hence each block would only display images from one category. For blocks with set size 6, all images from the image set were used. For blocks with a set size less than 6, images were randomly chosen from the image set. For example, if block 4 is randomly matched with the insect image set and yet has a set size of 2, then block 4 uses 2 insect images randomly drawn from the image set, such as a spider and a bee.

Appendix B

Other computational models attempted

Besides the 5 computational models reported in the paper, we also have attempted other versions of action confusion model, credit assignment models, attention filter models, and their mixture models. In total, we have attempted 24 computational models including the ones reported in the main body of the paper.

1. Other versions of action confusion model include:

   - We compute the weighted sum of Q values before applying the Softmax function:

$$\vec{\pi}_t^{RL} = Softmax\Big(\lambda \underbrace{\frac{\sum_{s\neq s_t, s\in S} \boldsymbol{Q}(s)}{|S|-1}}_{\text{confusion term}} + (1-\lambda) \underbrace{\boldsymbol{Q}(s_t)}_{\text{main term}}, \beta\Big) \tag{12}$$

   - The above version with RL-WM interaction (Collins, 2018):

$$\delta_t = r_t - (w\boldsymbol{W}_t(s_t, a_t) + (1-w)\boldsymbol{Q}_t(s_t, a_t)) \tag{13}$$

   - The first version with action confusion affecting WM as well and having WM and RL share the same λ^{DT} and λ^{ST}:

$$\vec{\pi}_t^{WM} = Softmax\Big(\lambda \underbrace{\frac{\sum_{s\neq s_t, s\in S} \boldsymbol{W}(s)}{|S|-1}}_{\text{confusion term}} + (1-\lambda) \underbrace{\boldsymbol{W}(s_t)}_{\text{main term}}, \beta\Big) \tag{14}$$

   - The version above but instead , WM has its own λ^{WM} that is the same across two conditions.

   - The first version with one more parameter $discount \in \{1, 2\}$. It captures the discount of WM capacity in the Dual-Task condition: $C^{DT} = C^{TS} - discount$. It is fixed to be 1 when fitting the 5 models reported in the main body and thus is not considered a model parameter for them.

2. Credit assignment models include:

- Instead of positing the confusion takes place during the policy formation stage, the first version of this class of models assumes that the confusion happens during the value updating stage. Similarly, λ is free to vary in two conditions, and δ_s is the state-specific prediction error that would have occurred had unseen state s been the actual current state. The following update takes place for all $s \neq s_t$ at each trial:

$$\boldsymbol{Q}_{t+1}(s \neq s_t, a_t) = \boldsymbol{Q}_t(s \neq s_t, a_t) + \lambda\alpha\delta_s \tag{15}$$

- The above version with RL-WM interaction (equation 13).
- The above version with one more interaction happening during credit assignment:

$$\delta_s = r_t - (w\boldsymbol{W}_t(s \neq s_t, a_t) + (1-w)\boldsymbol{Q}_t(s \neq s_t, a_t)) \tag{16}$$

3. Mixture of action confusion models and credit assignment models include:

- The second version of credit assignment model (the one with only one RL-WM interaction) mixed with the first version of action confusion model and thus with two more λ parameters for each condition for action confusion.
- The above version but fixing only one λ instead of two for action confusion.

4. Attention filter models include:

- Try to explain action confusion alternatively by applying an attention filter on stimuli before RL policy formation where β_{filter} is allowed to vary between two conditions. By applying a Softmax function on the stimulus indicator vector (equation 17), this class of model essentially posits that taxing WM makes the participant uncertain of which stimulus is the current one:

$$\vec{u}_t = \boldsymbol{Q}^\top Softmax(\mathbf{1}_{\{s_t\}}(\vec{S}), \beta_{filter}) \tag{17}$$

$$\vec{\pi}_t^{RL} = Softmax(\vec{u}_t^{\top}, \beta) \tag{18}$$

- The above version but instead of Q values, apply attention filter onto the W values during WM policy formation, leaving RL unchanged.
- The combination of the above two versions with the same set of β_{filter}.

5. Mixture of action confusion models and attention filter models include:

   - First version of action confusion model mixed (equation 12) with the second version of attention filter model.
   - The above model but assuming two conditions have the same β_{filter}.
   - The above model with RL-WM interaction (equation 13).
   - The second model with filtering only in the Dual-Task condition.
   - Follow the second version of attention filter model in the Task-Switch condition but follow the third version of action confusion model (equation 12 and 14) in the Dual-Task condition.
   - The above version with RL-WM interaction (equation 13).

Out of these attempts, the other action confusion models did not perform much differently from the one reported in the main body of the paper except for the fifth one which has an extra parameter and thus was penalized for the extra complexity. However, they all have worse testing phase validation, especially in set size 2 compared to the best fitting model reported in the Result section. The credit assignment models all did worse than the action confusion models and their mixture models performed the worst partially due to the increased complexity. The attention filter models and the mixtures performed worse than both credit assignment and action confusion models. Nonetheless, the second and third versions of attention filter models somehow produced a qualitatively better learning phase validation in the Dual-Task condition. Additionally, adding RL-WM interaction (equation 13 and 15) did not significantly change the model performance.